Measurement 119 (2018) 117–128

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

A machine learning model for improving healthcare services on cloud computing environment

Ahmed Abdelaziz[a], Mohamed Elhoseny[b,*], Ahmed S. Salama[c,d], A.M. Riad[b]

[a] Department of Information Systems, Higher Technological Institute, Cairo, Egypt
[b] Faculty of Computers and Information, Mansoura University, Egypt
[c] Information Systems Department, Faculty of Computing and Information Technology, University of Jeddah, Saudi Arabia
[d] Department of Computer and Information Systems Department, Sadat Academy for Management Sciences, Cairo, Egypt

ARTICLE INFO

Keywords:
Cloud computing
Health services
Parallel particle swarm optimization
Linear regression
Neural network
Chronic kidney disease

ABSTRACT

Recently, cloud computing gained an important role in healthcare services (HCS) due to its ability to improve the HCS performance. However, the optimal selection of virtual machines (VMs) to process a medical request represents a big challenge. Optimal selection of VMs performs a significant enhancement of the performance through reducing the execution time of medical requests (tasks) coming from stakeholders (patients, doctors, etc.) and maximizing utilization of cloud resources. For that, this paper proposes a new model for HCS based on cloud environment using Parallel Particle Swarm Optimization (PPSO) to optimize the VMs selection. In addition, a new model for chronic kidney disease (CKD) diagnosis and prediction is proposed to measure the performance of our VMs model. The prediction model of CKD is implemented using two consecutive techniques, which are linear regression (LR) and neural network (NN). LR is used to determine critical factors that influence on CKD. NN is used to predict of CKD. The results show that, the proposed model outperforms the state-of-the art models in total execution time the rate of 50%. In addition, the system efficiency regarding real-time data retrieval is greatly improved by 5.2%. In addition, the accuracy of hybrid intelligent model in predicting of CKD is 97.8%. The proposed model is superior to most of the referred models in the related works by 64%.

1. Introduction

In recent years, cloud computing gained a great attention in HCS applications due to its ability to provide different medical services over the internet. Cloud computing allows applications to provide infrastructure services to big numbers of stakeholders with assorted and dynamically changing requirements [1]. Technically, cloud is composed of datacentres, hosts, VMs, resources, etc. Datacentres are containing a big number of resources and list of different applications. Hosts are composed of several VMs to store and regain several medical resources to stakeholders. Cloud computing uses the virtualization technique which permits to share a single physical instance of a resource or an application among various stakeholders and enterprises [2]. It does this by allocating a logical name to a physical storage and providing a pointer to that physical resource when requested.

Virtualization consists of hardware virtualization, operating system (OS) virtualization, server virtualization and storage virtualization in cloud computing environment. Hardware virtualization is mainly done for the host platforms, because controlling VMs is much easier than controlling a physical host [3]. Operating System Virtualization is mainly used for experimenting different applications on different platforms of OS [4]. Server virtualization can be divided into several physical hosts on the request basis. Storage virtualization is created to recovery purposes. Virtualization plays a very significant role in the cloud computing, stakeholders' share the medical data in the clouds like medical applications etc. [5].

Currently, many healthcare applications that are used for diseases diagnosis or prediction does not support real time use which enable the stockholders to access them anytime and anywhere [6–12]. However, the time delay represents a big challenge for the most of stakeholders in HCS applications that run the medical requests on a cloud computing environment. In this paper, a new model for diseases diagnosis and prediction is proposed for CKD. According to the recent health services applications [5,8,11], surveys showed that CKD is one of the most serious diseases facing the world where the latest statistics are recorded 2.5–11.2% across Europe, Asia, Australia and North America are suffering from CKD. The United States of America has 27 million people and 50 thousand people in Egypt suffering from CKD. In addition, most available CKD diagnosis and prediction applications are based on traditional statistical methods which may lead to less accurate results.

* Corresponding author.
E-mail address: mohamed_elhoseny@mans.edu.eg (M. Elhoseny).

https://doi.org/10.1016/j.measurement.2018.01.022
Received 1 July 2017; Received in revised form 22 December 2017; Accepted 15 January 2018
Available online 02 February 2018
0263-2241/

Accordingly, the contribution of this paper is two-fold. First, a VMs optimization model is proposed using PPSO algorithm to improve the performance of HCS applications in a cloud computing environment. Second, a CKD diagnosis and prediction model is proposed to reduce the execution time of CKD prediction requests processing and speeding up reply to CKD prediction requests coming from stakeholders, and maximizing utilization of cloud resources. The proposed CKD model is implemented using as a hybrid schema composed of LR and NN. LR is used to determine critical factors of CKD. Then, NN is used for CKD prediction.

The reset of the paper is arranged as follows: Section 2 introduces the basics and the background information related to the algorithms used to develop our proposed model. The recent related work is discussed at Section 3. Section 4 describes the proposed cloud computing optimization model for VMs. Section 5 explains in details the proposed hybrid algorithms for CKD detection. Section 6 discusses the experimental results. Finally, Section 7 presents the conclusion and the future work.

2. Basics and background

2.1. Parallel particle swarm optimization

PSO has particles which perform elect solutions of the problem, each particle seeking for most favorable solution in the search space, each particle or candidate solution has a position and velocity. A particle updates its velocity and position based on its inertia, own experience and gained knowledge from other particles in the swarm, aiming to detect the best solution of the problem. The particles update its position and velocity according to the following Eqs. (1) and (2) [13,14]:

$$V_1^{k+1} = wV_1^k + c_1 rand_2 \chi (pbest_i - S_i^k) + (gbest_i - S_i^k) \quad (1)$$

where:
V_1^{K+1} = Velocity of agent i at iteration k,
W = Weighting function,
Rand = Random number between 0 and 1,
S_i^k = Current position of agent iteration k,
pbesti = Pbest of agent i,
$gbest_i$ = gbest of the group.

- The weighting function used in Equation (2):

$$W = W_{max} - ((W_{max} - W_{min})/iter_{max})\chi iter \quad (2)$$

where: W_{max} Initial weight,
W_{min} = Final weight,
$iter_{max}$ = Maximum iteration number,
iter = Current iteration number.

PPSO algorithm divides the population into sub-population and stratifies the algorithm separately to these sub-populations to minimize execution time [13]. PPSO can be executed on distributed systems. There are other types of parallel algorithms such as Master-Slave Models, but it is based on synchronous system. This paper aims to use asynchronous system through PPSO. PPSO composed of two types [14]:

- Synchronous PPSO, the particles can wait for it to finish from analyzing optimal point in the population before moving on to another task.
- Asynchronous PPSO, the particles (solutions) or as aforementioned before implementation time generated in the next iteration are analyzed before the current design iteration is finished.

2.2. Chronic kidney disease

CKD is one of the most serious diseases of the world. There are many factors that lead to CKD, shown below in Table 1.

Table 1
Preliminary factors that influence on CKD.

No	Factor	Information factor	Description
1	Age	Numerical	Years
2	Blood pressure	Numerical	Mm/Hg
3	Specific gravity	Nominal	1.005, 1.010, 1.015, 1.020, 1.025
4	Albumin	Nominal	0.1.2.3.4.5
5	Sugar	Nominal	0.1.2.3.4.5
6	Red blood cells	Nominal	Normal, abnormal
7	Pus cell	Nominal	Normal, abnormal
8	Pus cell clumps	Nominal	Present, notpresent
9	Bacteria	Nominal	Present, notpresent
10	Blood glucose random	Numerical	Mgs/dl
11	Blood urea	Numerical	Mgs/dl
12	Serum Creatinine	Numerical	Mgs/dl
13	Sodium	Numerical	mEq/L
14	Potassium	Numerical	mEq/L
15	Haemoglobin	Numerical	Gms
16	Packed cell volume	Numerical	Gms
17	White blood cell count	Numerical	Cells/cumm
18	Red blood cell count	Numerical	Millions/cmm
19	Hypertension	Nominal	Yes, no
20	Diabetes mellitus	Nominal	Yes, no
21	Coronary artery disease	Nominal	Yes, no
22	Appetite	Nominal	Good, poor
23	Pedal edema	Nominal	Yes, no
24	Anemia	Nominal	Yes, no
25	Class	Nominal	CKD, NOCKD

2.3. Linear regression analysis

LR is used to determine critical factors of CKD. Simple regression is composed of one dependent variable and one independent variable. Multiple regression analysis is composed of one dependent variable and more than independent variable. It is formulated as follows [15]:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ...+\beta_n X_n + \varepsilon \quad (3)$$

where:

- y represents the dependent variable
- $x_1, x_2, \ldots, x_n$ are the independent variables
- β_i is the regression coefficient
- ε is the random error component.
- β_0 is the y intercept

Y represents the dependent variable (degree of influence on CKD process) and x1, x2… x_n are the independent variables (factors that influencing on CKD).

2.4. Neural network overview

NN is a novel process of programming computers. It is represented of the human brain's information processing mechanism. NN is applied on more applications such as pattern recognition, diagnosis of diseases and data classification, through a learning process. NN has many types of networks such as feed-forward network and back-propagation network. NN is composed of input layers (factors that influence on CKD), hidden layers and output layers (decision, (CKD or NOCKD)), as follows:

- Input Layer – The input units represent the raw data that is fed into the network.
- Hidden Layer – The hidden unit is specified by the input units and the weights on the connections between the input and the hidden units.
- Output Layer – The attitude of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

3. Related work

Through related work, many studies were done on applying and using different optimization algorithms to determine optimal VMs on cloud environment. Previous work also introduces a set of studies based diagnosis of CKD on cloud environment, as follows:

Fang et al. [16], introduced a new framework to find the optimal VMs placement of cloud environment based on ant colony optimization (ACO). This study tries to detect VMs placement of datacentre in order to maximize quality of VMs, minimize energy cost and Energy-Saving. The contribution of this study is to use ACO to detect the optimal VMs placement on cloud computing environment.

Shabeera et al. [17], introduced model to find optimal VMs on cloud environment based on ACO algorithm. This study seeks to detect the optimal of VMs in datacenter to reduce energy consumption in servers and enhance data placement for data-intensive applications. The contribution of this study is to use ACO to find the optimal of VMs in cloud computing environment.

Almezeini et al. [18], presented a new method to improve task scheduling based on lion optimization of cloud environment. This study seeks to save power consumption of servers, maximize resources utilization and improve performance of VMs in datacenter. The contribution of this study is to enhance task scheduling on cloud computing environment.

Mishra et al. [19], presented a new model to improve the task scheduling based on bee colony optimization (BCO). This study seeks to enhance the task scheduling to give quality of services of users' tasks and minimize time request. The contribution of this study is to use BCO algorithm to enhance the task scheduling to reduce time request and maximize resources utilization.

Mohana et al. [20], presented a new way to obtain the optimal task scheduling based on PSO and ACO. This study tries to find the optimal task scheduling to enhance the overall execution time of the task and maximize utilization of resources. The contribution of this study is to use a new method to find optimal task scheduling of cloud computing environment.

Salama [21] introduced a novel approach to enhance the access time for the mobile cloud computing services and enhancing the mobile commerce transactions based on PPSO technique. This paper uses PSO technique to examine the quality of the proposed PSO and PPSO based techniques for mobile cloud computing. The contribution of this study is to use PPSO algorithm to reduce time of tasks and helping to maximize resources utilization in cloud environment.

Sung et al. [22], presented a new method to collect different data from patients in hospitals through multi-sensors and analyse it based on improved PSO on cloud computing environment. The contribution of this study is to use improved PSO to analyse patients' data efficiently.

Seddigh et al. [23], presented a new model to evaluate the scheduling of VMs in datacentre based on ACO algorithm. This study tries to predict of VMs in datacentre to save power consumption. Simulation results are implemented on CloudSim package tool. The contribution of this study is to use ACO to evaluate the scheduling of VMs in cloud environment.

Vidhya et al. [24], presented a new way to find optimal task scheduling based on PPSO algorithm. This study seeks to find optimal task scheduling to reduce execution time of requests and maximize resources utilization. The contribution of this study is to use PPSO algorithm to enhance task scheduling on cloud computing environment.

Thiruvenkadam et al. [25], presented a novel framework to find the best VM placement in datacentre based on RR and FCFS algorithms. This study seeks to enhance VM placement to maximize resources utilization; reduce response time Power Usage, Load Imbalance Rate, Migration Rate and better services to the cloud users. The contribution of this study is to use a new framework to find the best VM placement in cloud environment.

Barlaskar et al. [26], introduced a new method to get energy efficient of VM placement based on the hierarchical cluster based modified firefly algorithm (HCMFF). The goal of energy VM placement is to optimize energy consumption. The simulation results show that HCMFF outperforms honeybee algorithm and original firefly. The contribution of this study is to use HCMFF to reduce energy VM placement in cloud environment.

Teyeb et al. [27], introduced a new approach to solve the problem of VMs placement in geographically distributed datacentres on cloud computing based on a formulation. This study aims to detect the optimal VMs and to minimize the total running time and computational resources. The contribution of this study is to use new formulation based on multi-commodity flow, adopt variable aggregating methods to find the optimal VMs placement on cloud environment.

Chaurasia et al. [28], introduced a novel way to choose optimal VM placement in datacentre based on the technique for order of preference by similarity to ideal solution (TOPSIS). This study seeks to choose optimal VM placement to minimize power consumption in servers, minimize response time and maximize resources utilization. The contribution of this study is to use TOPSIS to choose optimal VM placement in cloud environment.

Fu et al. [29], introduced a new method to find optimal VMs in datacentre based on Dynamic consolidation of VMs. The proposed method concentrates on minimizing the cost spending in each plan for hosting VMs in multiple cloud providers and the response time of each cloud provider is monitored periodically, in such a way to minimize delay in providing the resources to the users. The contribution of this study is to use Dynamic consolidation to minimize energy consumption, response time and also improve physical resource utilization.

Shrivastava et al. [30], introduced a new way to detect VM allocation in datacentre based on best fit decreasing. This study seeks to solve VM allocation in datacentre to enhance resources utilization. The contribution of this study is to use a new way to solve VM allocation problem in cloud computing environment.

Jena et al. [31], introduced new method to predict kidney failure disease based classification techniques of data mining. Multilayer perception outperforms Naïve Bayes, support vector machine, conjunctive rule and decision table. Accuracy of multilayer perception is 99.75%. Classification techniques were applied on WEKA mining tool. The contribution of this study is to use multilayer perception to predict kidney failure disease.

Batra et al. [32] presented a survey to diagnosis chronic kidney failure based on predictive analytics method by exploiting the potential of hadoop and map/reduce tool. This study introduced many researches based on diagnosis or predicting chronic kidney failure and factors that influence in it. The results showed that the critical factors influence on chronic kidney failure.

Boukenze et al. [33], introduced a comparative study between support vector machine, decision tree and Bayesian network to predict chronic kidney failure. Decision tree outperforms on support vector machine and Bayesian network in predicting of chronic kidney failure. Accuracy of decision tree is 97%. The intelligent techniques are applied on hadoop and map/reduce tool. The contribution of this study is to use decision tree to predict chronic kidney failure appropriately.

Padmanaban et al. [34], introduced a comparative study between Naïve Bayes and decision tree for predicting chronic kidney failure. Decision tree outperforms on Naïve Bayes in predicting of chronic kidney failure. Accuracy of decision tree is 91%. The intelligent techniques are applied on WEKA mining tool. The contribution of this study is to use decision tree to predict chronic kidney failure appropriately.

Salekin et al. [35], presented a comparative study between K-Nearest Neighbours' and Random Forest to predict of chronic kidney failure. K-Nearest Neighbours' outperforms on Random Forest in predicting of chronic kidney failure. Accuracy of K-Nearest Neighbours' is 99.3%. The intelligent techniques are applied on WEKA mining tool. The contribution of this study is to use K-Nearest Neighbours' to predict chronic kidney failure.

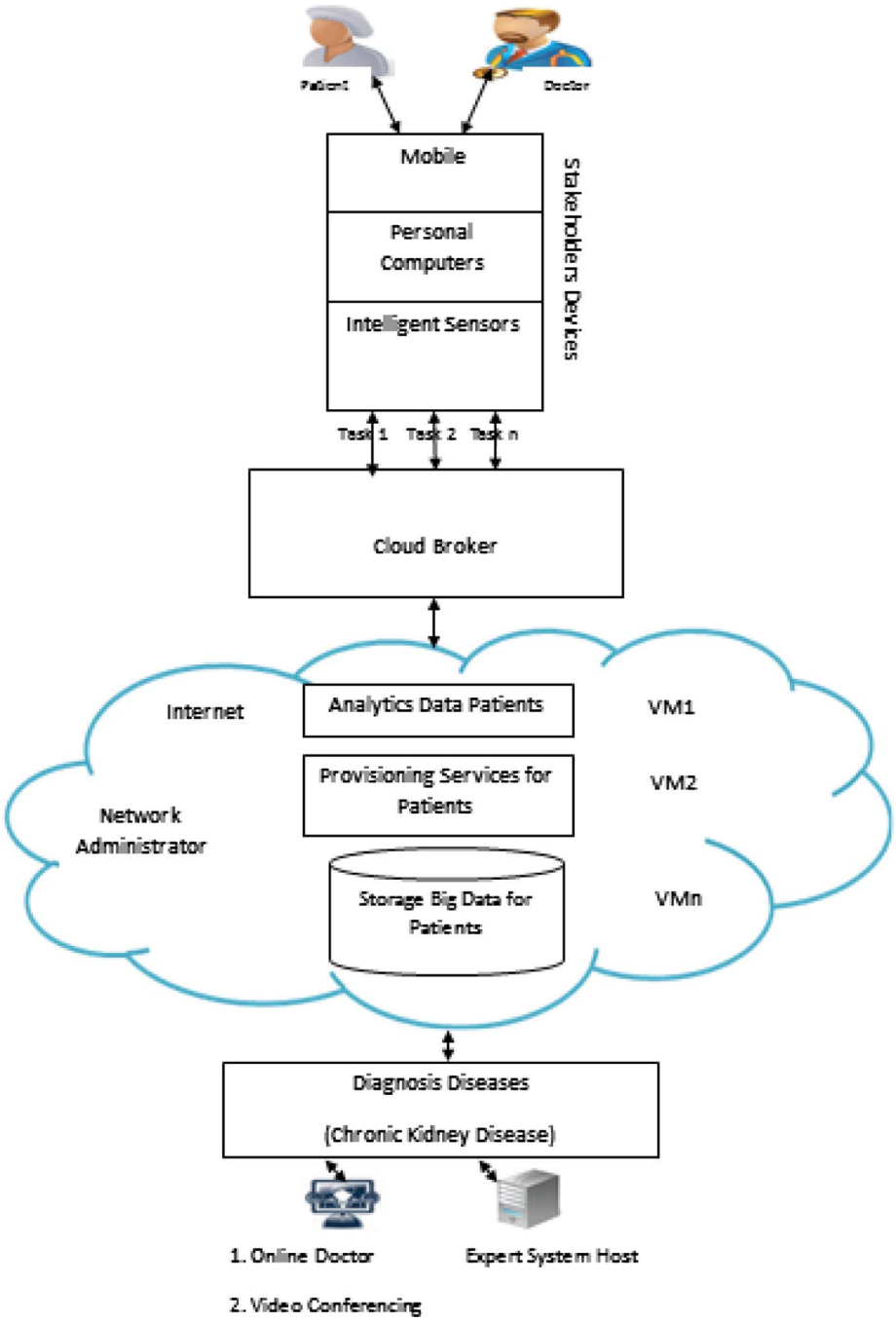

Fig. 1. VMs optimization model for cloud computing based HCS.

4. The proposed could computing based optimization model for VMs

This section describes the architecture of the proposed cloud computing model for HCS. It consists of four components are stakeholders' devices, stakeholders' requests (tasks), cloud broker and network administrator as shown below in Fig. 1. The communication devices services are responsible for implementing different network communication management between stakeholders and the cloud.

Stakeholders use a variety of devices (PC, Laptop, Smartphone, Tablet, Digital sensors, etc.) to send a variety of medical readings and requests (tasks) easily through cloud computing to obtain different medical services such as diagnosis of diseases (CKD) as an example of healthcare services. Cloud broker is responsible to send and receive requests (tasks) from the cloud service. Each network may have several application hosts = {Host1, Host2… and $Host_N$} providing the SaaS and can be allocated to execute the cloud stakeholder's requests. Each application host has a set of resources = {R1, R2 … and R_N} that can be allocated for the coming stakeholders requests. Each network has a network administrator that is responsible for the coordination of the communication between the hosts inside the network and between this network and other networks in the clouds. Network administrator is responsible for running the PPSO algorithm which leads to the maximum resources utilization. In addition, this architecture provides a hybrid machine learning model for predicting CKD as an example towards improving the healthcare services. The proposed VMs optimization model is implemented using PPSO while LR and NN are used for CKD diagnosis and prediction respectively as described at Section 5.

Table 2
Criteria of optimal selection of VMs.

SN	Criterion	Description	Formula
1	CPU utilization (U)	The percentage of CPU capacity used during a specific period of time.	U = 100% − (%time spent in the idle task)
2	Turnaround time (TT)	Time difference between completion time and arrival time.	TT = CT − AT
3	Waiting time (WT)	Time difference between turnaround time and burst time.	WT = TT − BT

4.1. VMs optimizing model using PPSO algorithm

The proposed architecture is implemented using PPSO algorithm. To calculate the execution time of stakeholders' requests, the proposed fitness function is a composition of three significant criteria which are CPU utilization, turn-around time and waiting time. The calculations of these criteria depend on the following three parameters:

- **Arrival Time (AT):** time at which the task arrives in the ready queue.
- **Burst Time (BT):** time required by a task for CPU execution. BT calculated as follow:

$$BT = \text{Clock Time to Burst} \div \text{Burst Ratio} \quad (4)$$

where:

$$\text{Burst ratio} = \text{Burst Threshold} \div \text{Burst Limit} \quad (5)$$

- **Completion Time (CT):** time at which task completes its execution.

Table 2 presents description and formula for calculating the three significant criteria using the previously mentioned parameters to reach the optimal selection of VM.

The proposed PPSO algorithm tries to find optimal selection of VMs to reduce execution time of requests (tasks) from stakeholders and maximization of resources utilization.

As shown in Fig. 2, assume that there are M particles (VMs) = 100, C1 = 1.3, C2 = 1.3, C3 = 1.3 and the number of iterations = 100. Each VM in the cloud(s) is considered a particle which represents a potential solution (VM) that can be allocated for executing the stakeholder's subtasks. In PPSO, the parallel processing aims to produce the same results of the classical PSO using multiple processors simultaneously with the goal of reducing the run time. Compute fitness function (optimal selection of VMs) by using U, TT, and WT. The barrier synchronization stops the algorithm from move to the next step until the fitness function (optimal selection of VMs) has been reported, which is required to maintain algorithm coherence. Ensure that all of the particle (VM) fitness evaluations have been completed and results reported before the velocity and position calculations can be executed. Compare the calculated fitness function of each particle (VM) with its pbesti. If current value is better than pbest, then put the current location as pbest location. Moreover, if current value is better than gbest, then reconstruct gbest to the current index in particle array. Assign the best particle (VM) as gbest. Update each Particle Velocity and position according to Eqs. (6), (7) as follows:

(a) Update particle velocity according to Eq. (6) [36].

$$\begin{aligned} VI_{,j}(t) = {} & W\,VI_{,j}(t-1) + \\ & C1\,R1\,(Pi_{,j}(t-1) - Xi_{,j}(t-1)) + \\ & C2\,R2\,(Ps_{,j}(t-1) - Xi_{,j}(t-1)) + \\ & C3\,R3\,(Pg_{,j}(t-1) - Xi_{,j}(t-1)) \end{aligned} \quad (6)$$

where:

- Xi, j = the position of ith particle in jth swarm,
- VI, j = the velocity of ith particle in jth swarm,
- Pi, j = the pbest of ith particle in jth swarm,
- Ps, j = the swarm best of jth swarm,
- Pg, j = the global best among all the sub swarms,
- C1, C2, C3 = acceleration parameters
- R1, R2, R3 = the random variables.

(b) Update particle position according to Eq. (7) [36].

$$Xi_{,j}(t) = Xi_{,j}(t-1) + VI_{,j}(t) \quad (7)$$

Xi, j (t) = the current position of ith particle in jth swarm, Xi, j

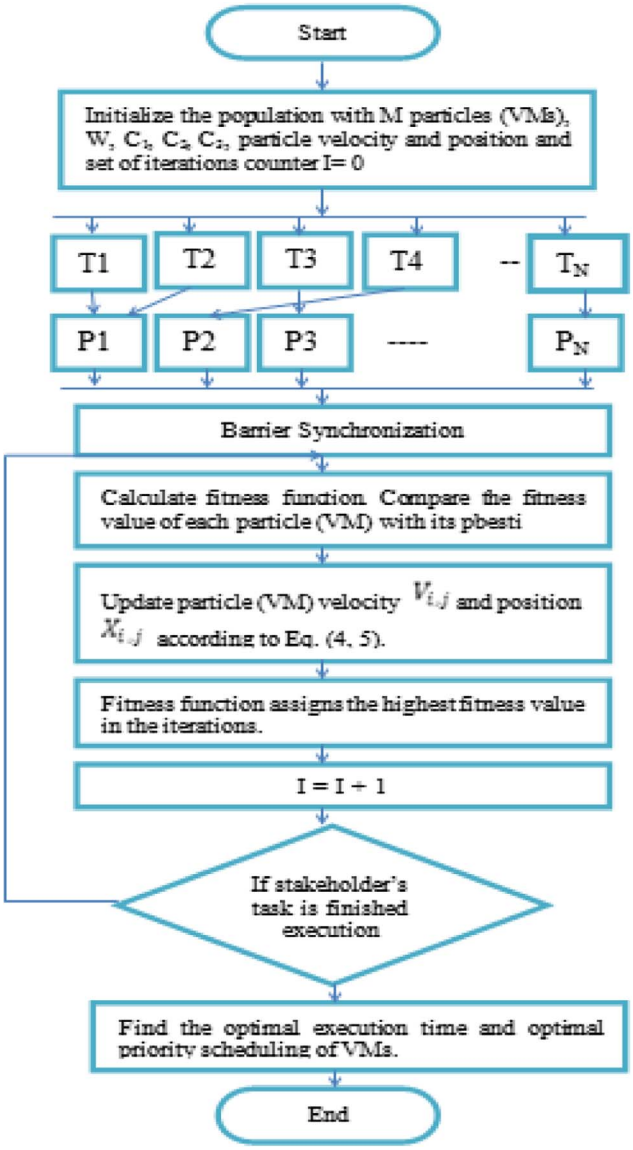

Fig. 2. The proposed flow chart of PPSO for cloud computing.

(t − 1) = the new position of ith particle in jth swarm, VI, j (t) = the current velocity of ith particle in jth swarm.

Algorithm 1. The Proposed PPSO for Selecting VMs on Cloud Computing

```
Input: Size α of Population,
  Rate β of P_g-best,
  Rate σ of P_p-best,
  Number Y of Iterations
Output: P_g-best ← (Optimal Particles (VMs), Optimal Exec Time)
Stakeholders Tasks = X;
Set number of processors
Count I = 0;
For I = 0 to α do
  P_velocity ← Random velocity();
  P_position ← Random position(α);
  σ ← P_position;
If (cost(σ) ≤ cost(β))
  β ← σ;
  End if
End for
Set a barrier synchronization;
While (X ≠ 0)
For I = 0 to Y do
  Compute fitness function;
  Update velocity (P_velocity, β, σ);
  Update position (P_velocity, P_position);
    P_velocity ← Update velocity;
    P_position ← Update position;
If (cost(P_position) ≤ cost(σ))
    σ ← P_position;
If (cost(σ) ≤ cost(β))
    β ← σ;
If (X is finished = = true)
    X = X − 1;
    Save β;
Else
    Compute fitness function for each
    particle(VM);
    End if
  End if
End if
I = I + 1;
  End for
End while
Return β;
```

5. The hybrid Machine learning model for predicting CKD

5.1. CKD diagnosis using LR analysis

This section introduces a LR analysis model for CKD diagnosis. LR analysis is used to specify the critical factors that affect CKD behavior. Fig. 3 shows the flowchart of the proposed algorithm. LR provides the mechanism for regression statistics such as the mean absolute error (MAE), the root mean squared error (RMSE), the relative absolute error (RAE), the relative squared error (RSE) and the coefficient of determination (CD). MAE is a quantity used to measure how close predictions are to the eventual outcomes. RMSE can be compared between models whose errors are measured in the same units. RAE can be compared between models whose errors are measured in the different units. RSE can be compared between models whose errors are measured in the different units. CD summarizes the explanatory power of the regression model. The steps of the proposed algorithm are as follows:

Algorithm 2. The Proposed Algorithm of the Multiple Linear Regressions to Determine Critical Factors that Influence on CKD.

```
Input: α ← (dependent variable degree of influence on CKD process),
  σ ← (independent variables factors that influencing on CKD)
Output: β ← (critical factors list of CKD)
LR = Linear Regression
MAE = Mean Absolute Error
RMSE = Root Mean Squared Error
RAE = Relative Absolute Error
RSE = Relative Squared Error
CD = Coefficient of Determination
FW = Feature Weights
Create the LR model based on the number of α and σ
Evaluate the LR model
Check the value of MAE. MAE is calculated as follows:
    MAE = (1/n) Σ_{i=1}^{n} |fi−wi|
Check the value of RMSE. EMSE is calculated as follows:
    RMSE = sqrt( Σ_{i=1}^{n} (fi − wi)2 / n )
Check the value of RAE. RAE is calculated as follows:
    RAE = Σ_{i=1}^{n} | fi − wi| / Σ_{i=1}^{n} | ϖ − wi|
Check the value of RSE. RSE is calculated as follows:
    RSE = Σ_{i=1}^{n} (fi − wi)2 / Σ_{i=1}^{n} (ϖ − wi)2
Check the value of CD. CD is calculated as follows:
    R^2 = SSR/SST = 1 − SSE/SST
R^2 = CD
If (CD < 0.5)
  Change the explanatory factors;
  Go to step 11;
Else
  Approve the model
End If
Check FW for each variable to determine β
If (FW < 0.05)
  Approve β
Else
  Refuse the other factors
End If
Return β
```

where:

- f_i, is the prediction and ϖ the true value.
- ϖ, is the mean of y_i.
- Sum of Squares Regression, SSR = $\sum_i (fi-\varpi)2$
- Sum of Squares Total, SST = $\sum_i (wi-\varpi)2$
- Sum of Squares Error, SSE = $\sum_i (wi-fi)$

5.2. The proposed NN model for CKD prediction

LR model showed that thirteen factors out of twenty-four of them

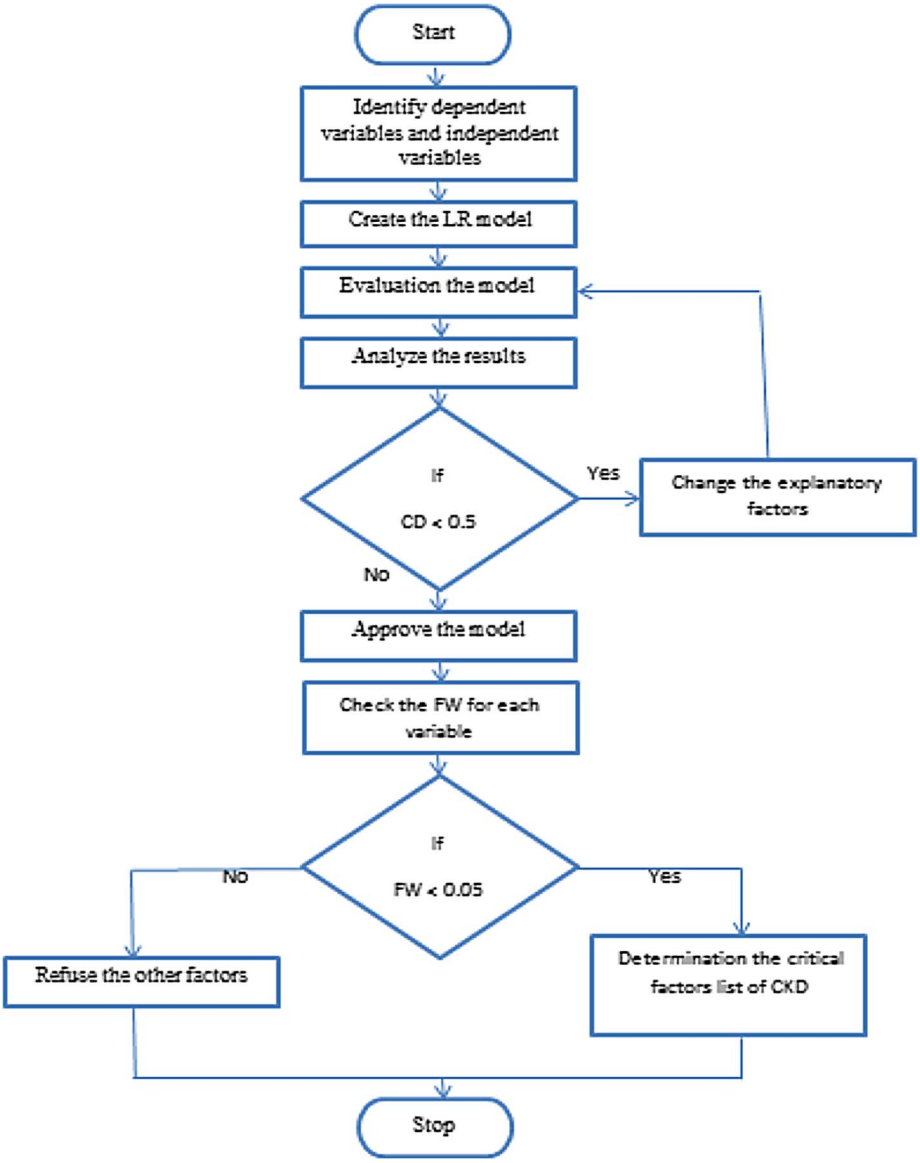

Fig. 3. Flow chart of the proposed algorithm for LR Model.

Table 3
MSE for the network.

Number of hidden layers	MSE values
2	0.3563
1	0.2675
3	0.3071
4	0.6501

had an effect on CKD. The NN uses thirteen factors as inputs in a network. NN uses three hidden layers based mean squared error (MSE). MSE is used to evaluate the performance of the model. MSE is calculated using Eq. (8) [37]. Whenever, MSE is small that indicate better Performance of the NN model. The NN model uses one hidden layer with MSE = 0.2675. The accuracy of the NN model with different hidden layers is presented in Table 3. It uses one output in a network (class, "CKD", "NOCKD"), shown below in Fig. 4.

$$\mathrm{MSE} = \sum_{j=0}^{P} \sum_{i=0}^{N} (\mathbf{t}_{ij} - \mathrm{y}_{ij})^2 \div \mathrm{NP} \tag{8}$$

where:

- P is the number of output possessing elements.
- N is the number of observations.
- t_{ij} is the target outputs.
- y_{ij} is the actual outputs.

This section introduces an algorithm for predicting CKD by NN technique. The performance of the NN model is based on accuracy. Accuracy of NN model is calculated by using true positive (TP), false positive (FP), true negative (TN), false negative (FN) from CKD data set. Fig. 5 shows the flow chart of the proposed algorithm. The steps of the proposed algorithm are as follows:

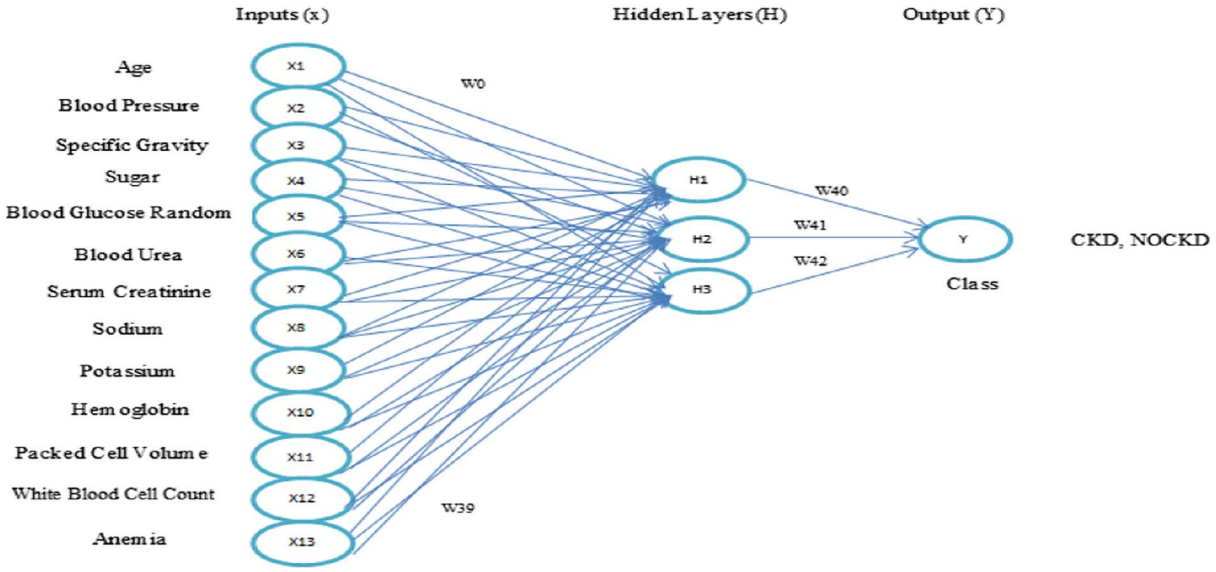

Fig. 4. The proposed NN model to predict of CKD.

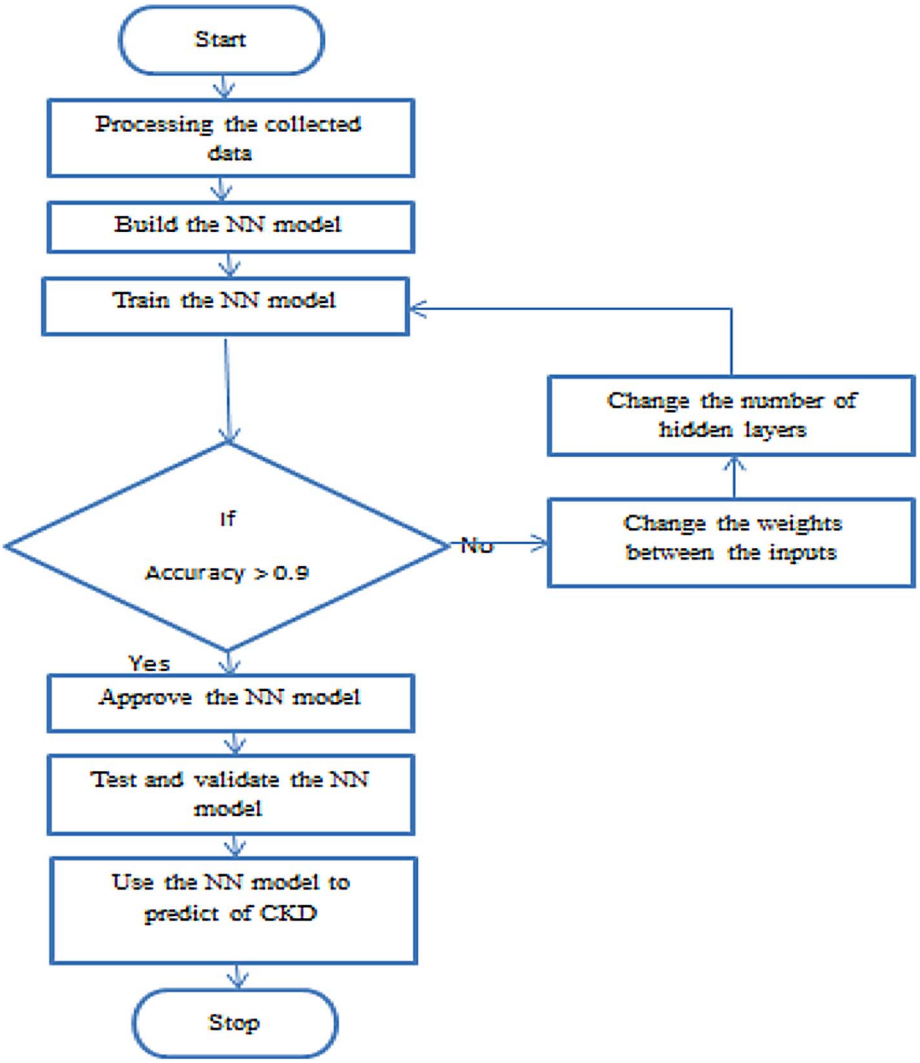

Fig. 5. Flow chart of the proposed algorithm for NN Model.

Algorithm 3. The Proposed Algorithm of the NN model to predict of CKD.

Input: α ← (Input layer thirteen critical factors that influence on CKD),
    σ ← (hidden layers three hidden layers),
    W ← (Random weights in the neural network model)
Output: β ← (Decision of the prediction (class,(CKD, NOCKD)))
NN = Neural Network
V = Accuracy of the NN model
TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative
Divide the chronic kidney data into training, test, and validate
Build the NN model initially with α, σ, W and β
Train the NN model by using two classes NN
Check the value of V. V is calculated as follows:
    $V = \frac{(TP + TN)}{(TP + TN + FP + FN)}$
If (V > 0.9)
  Approve NN model
  Go to step 19
Else
  Change W between α and σ
  Change the number of σ
  Go to step 8
End If
Determine V through test and validate the network
Return β

6. Experimental results

This section discusses the experimental results of our proposed model. The model is implemented using two different tools. The first tool is CloudSim package which is used to implement the proposed PPSO for VMs optimization [38]. While the second one is the Windows Azure which is used to implement the hybrid LR and NN model [39].

6.1. Experiments for VMs optimization

Firstly, CloudSim is used to implement the proposed PPSO algorithm to find optimal selection of VMs. The first implementation is the default CloudSim where first task takes the first VM; the second task takes the second VM, etc. The results show that the total time that is required to build successful cloudlets is 3 s as shown in Table 4.

The second experiment is conducted by allowing the task x to be assigned VM_x or VM_{x+1}. This flexibility greatly reduces the running time. In this experiment, the total required time to build successful cloudlets is 1 s as shown at Table 5. This means that the improvement ratio is more than 30%.

Fig. 6 shows the inverse relationship between the number of processors and the Makespan (time) of requests from stakeholders. It is clear that whenever the number of the used processors increased, the

Table 4
Results of default CloudSim.

Cloudlet ID	Status	Datacentre ID	VM ID	Time	Start time	Finish time
0	Success	2	0	800	0.1	800.1
1	Success	2	1	1200	0.1	800.1
3	Success	2	3	8000	0.1	800.1
2	Success	2	2	16,000	0.1	800.1
Build Successful (total time: 3 s)						

Table 5
Sample of results of PPSO algorithm on CloudSim.

Cloudlet ID	Status	Datacentre ID	VM ID	Time	Start Time	Finish Time
0	Success	2	1	1600	0.1	1600.1
1	Success	2	1	2000	0.1	2000.1
3	Success	2	3	8000	0.1	8000.1
2	Success	2	2	16,000	0.1	16000.1
Build Successful (total time: 1 s)						

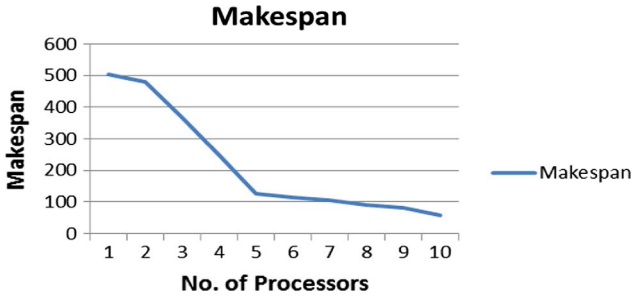

Fig. 6. Proposed PPSO leads to minimized makespan with increasing in No. of processors.

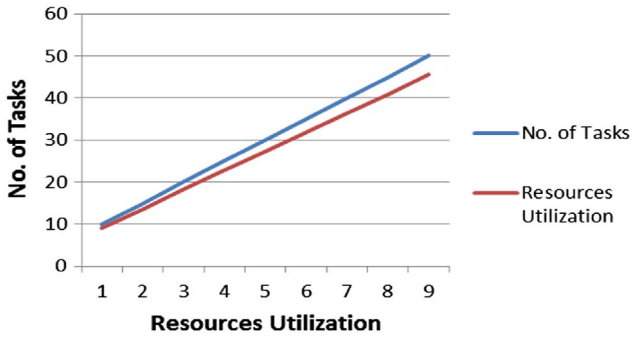

Fig. 7. Relationship between resources utilization and No. of tasks.

Table 6
Sample of total execution time of the proposed model compared to the related work algorithms.

No	Algorithm	Total execution time (ET) (sec)
1	Dashti et al. [12]	6.95
2	Parmar et al. [15]	7.82
3	Moorthy et al. [22]	1.5
4	Hanen et al. [25]	3
5	The proposed PPSO model	1

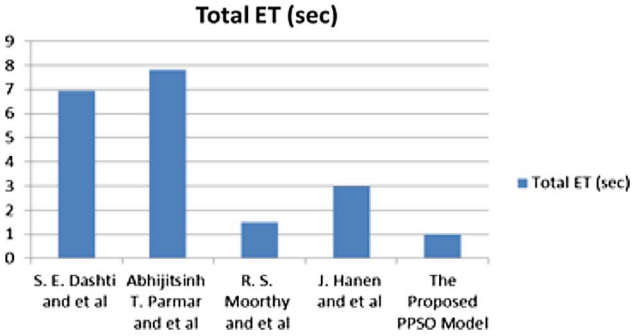

Fig. 8. The total ET of the proposed model compared to the state-of-the art models.

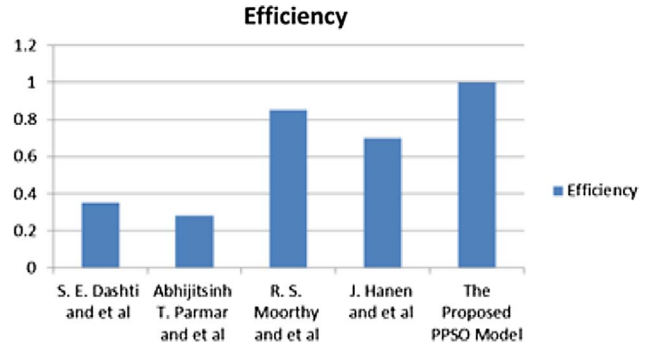

Fig. 9. The system efficiency of the proposed model compared to the state-of-the art models.

makespan value is highly decreased.

In addition, Fig. 7 shows the positive relationship between the number of tasks requested by different stakeholders and the resource utilization.

Compared to the state-of-the art methods, Table 6 lists the total execution time of the proposed model. The results show an improvement ratio in range of 50% to 88% regarding to the total ET. This big improvement can be noticed as shown at Fig. 8.

Continuously, Fig. 9 shows a competition between the proposed model and the state-of-the art methods regarding to the system efficiency. The results show that the proposed model greatly improves the system efficiency by 5.2%.

Table 7
Sample of factors that influencing on CKD.

No	Factor name	FW
1	Age	−0.001
2	Blood pressure	0.0006
3	Specific gravity	−0.001
4	Sugar	0.01
5	Blood glucose random	−0.0005
6	Diabetes mellitus	0.9008
7	Coronary artery disease	0.6012
8	Appetite	0.0801
9	Pedal edema	0.0616
10	Blood urea	0.001
11	Serum creatinine	−0.04
12	Sodium	0.002
13	Potassium	0.004
14	Haemoglobin	0.03
15	Packed cell volume	−0.001
16	White blood cell count	−0.00004
17	Anemia	0.01

The proposed model has a high flexibility through changing the number of processors, tasks, VMs and etc easily. The robust of the proposed model is clear concerning the efficiency of PPSO that outperformed the efficiency of default CloudSim package and the state-of-the art techniques and it succeeded to select the optimal VMs in cloud environment. The proposed PPSO algorithm can reduce execution time

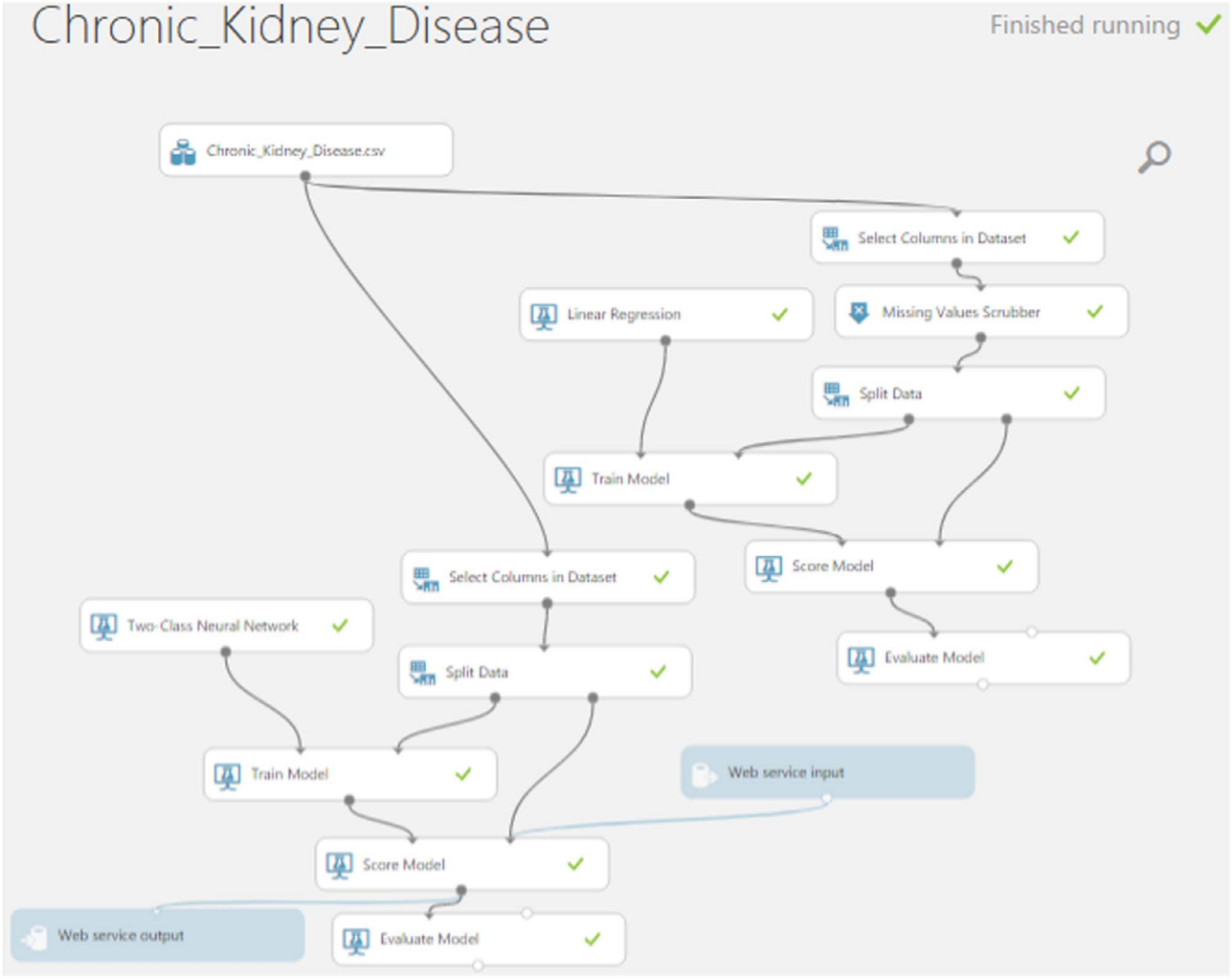

Fig. 10. Implementation of LR and NN on Windows Azure for predicting CKD.

Table 8
Summary of critical factors that influencing on CKD.

No	Factor name	FW
1	Age	−0.001
2	Blood pressure	0.0006
3	Specific gravity	−0.001
4	Sugar	0.01
5	Blood glucose random	−0.0005
6	Blood urea	0.001
7	Serum creatinine	−0.04
8	Sodium	0.002
9	Potassium	0.004
10	Haemoglobin	0.03
11	Packed cell volume	−0.001
12	White blood cell count	−0.00004
13	Anemia	0.01

Table 9
Regression statistics.

No	Factor name	FW
1	MAE	0.409284
2	RMSE	0.603883
3	RAE	0.460733
4	RSE	0.412968
5	CD	0.587032

Table 10
Sample of scored dataset.

No	Class	Scored labels	Scored Probabilities
1	CKD	CKD	0.000022
2	CKD	CKD	0
3	NOTCKD	NOCKD	0.959307
4	CKD	CKD	0.008337
5	CKD	CKD	0.417251
6	NOTCKD	NOCKD	0.937067
7	CKD	CKD	0

of requests and maximize utilization of resources. The versatile of the proposed model can adapt easily from one to another of various tasks.

6.2. Experiments of CKD prediction using Windows Azure

Fig. 10 shows the implementation details of the proposed CKD prediction model on Windows Azure.

6.2.1. Experimental results of LR model in Windows Azure

The experiment of LR model is executed by a set of sequential steps. First, the dataset variables are categories to dependent and independent variables (see Table 1). Then, the Missing Values Scrubber is used to provide some basic procedures for processing missing values. In our experiments, the dataset is divided into 70% training data and 30% testing data. The LR process starts by getting the ordinary least squares that assume strong linear relationship between the inputs and the dependent variable. After that, L2 Regularization is used to enhance the performance. The default value of L2 is 0.001. To training is conducted by using the feature weights listed at Table 7 which indicates the contribution of each independent variable to the LR model. If FW < 0.05, the factor is statistically important. For example, the FW for blood urea (FW = 0.001) is < 0.05, therefore, this factor should be accepted.

After the screening of FW-values, the most critical factors which influence in CKD are shown at Table 8.

The results show that MSE = 0.40, RMSE = 0.60, RAE = 0.46, RSE = 0.41 and finally CD = 0.58. The CD indicates that 0.58 of the variance in the dependent variable (degree of influence on CKD process) can be explained by the independent variables (selected factors), while the rest (0.42) is explained by other factors, as shown below in Table 9.

6.2.2. Experiments for the NN prediction model

In NN experiments, the dataset is divided into 70% training data and 30% testing data. The NN model contains thirteen inputs, two hidden layers and one output. The NN training parameters are the learning rate = 0.1, the number of learning iteration = 100, the initial learning weight = 0.1 and the type of normalizer is the min-max. The min-max normalizer linearly rescales every feature to the [0, 1] interval. Min-max normalizer is calculated using Eq. (9).

$$Z = \frac{x - \min(x)}{[\max(x) - \min(x)]} \tag{9}$$

In our proposed model, a score model is used to predict a value for each CKD class, as well as the probability of the predicted value as shown at Table 10. The Scored Probabilities indicates that the closer we get to zero, the greater the risk of CKD. On the contrary, the closer we get to one, the lower the probability of CKD.

The NN prediction model is evaluated by calculating TP, FP, TN, FN, accuracy, precision, recall and F1 score as shown below. TP is the ratio of the correctly identified data. FP is incorrectly identified data. TN is correctly rejected data while FN is incorrectly rejected data. As shown at Fig. 11, the results show that the accuracy is 0.978 of the total number of correct predictions. In addition, the precision is 0.962 of positive cases that were correctly identified. Moreover, the Recall is 1.000 of actual positive cases which are correctly identified. The F1 score is 0.981 the harmonic mean of precision and Recall. Accuracy, precision, recall and F1 score are calculated using Eq. (10)–(13).

$$\text{Accuracy} = \frac{(TP + TN)}{TP + TN + FP + FN} \tag{10}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \tag{12}$$

$$\text{F1score} = \frac{2TP}{(2TP + FP + FN)} \tag{13}$$

Beside the previous experiments, three patients were selected to evaluate the performance of the proposed CKD prediction model using their real data. Table 11 shows the details of this case study with thirteen critical CKD prediction factors.

As shown at Table 11, the first patient has CKD with probability (0.00004). As mentioned before, the closer the probability to zero, the greater the risk of CKD. The second and the third patients have NOCKD with probability (0.99992) and (0.99998) respectively. The closer the

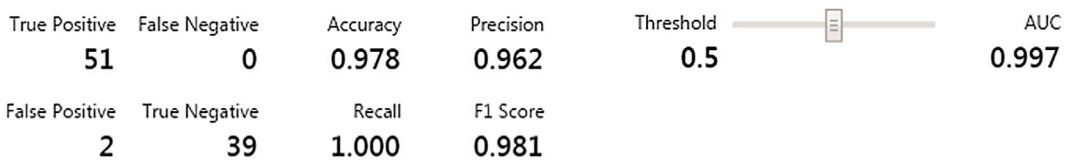

Fig. 11. Performance attributes of NN model.

Table 11
Actual data of three cases of CKD patients.

No	Factor name	Patient 1	Patient 2	Patient 3
1	Age	48	58	28
2	Blood pressure	70	80	100
3	Specific gravity	1.005	1.025	1.1
4	Sugar	0	0	1
5	Blood glucose random	117	131	120
6	Blood urea	56	18	20
7	Serum creatinine	3.8	1.1	1.2
8	Sodium	111	141	130
9	Potassium	2.5	3.5	2.1
10	Haemoglobin	11.2	15.8	10
11	Packed cell volume	32	53	50
12	White blood cell count	6700	6800	5000
13	Anemia	Yes	No	Yes
14	Class	CKD	NOCKD	NOCKD
15	Probability	0.00004	0.99992	0.99998

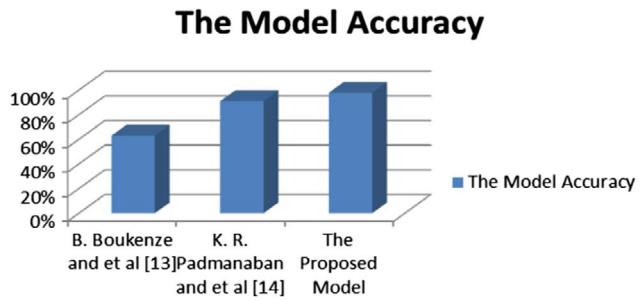

Fig. 12. The accuracy of the proposed model compared to the state-of-the art models.

probability is to one, the lower the risk of CKD.

Finally, the accuracy of the proposed model is compared to the state-of-the art methods as shown at Fig. 12. The results show that the proposed model greatly improves the CKD prediction accuracy by 64%.

7. Conclusion and future work

The stakeholders are facing a big challenge to during their interaction with HCS applications due to the limited resource and the time consumption. By determining the optimal VMs on cloud computing, HCS applications will be able to reduce the execution time. This paper proposes a new model for HCS in a cloud environment using PPSO to determine optimal selection of VMs. In addition, a hybrid model for predicting CKD based on cloud environment is proposed. The proposed CKD prediction model is implemented using two algorithms, which are LR and NN. The proposed model was tested against the state-of the art method to evaluate its efficiency. The results show that the proposed model outperforms the state-of-the art models in total execution time the rate of 50%. In addition, the system efficiency regarding real-time data retrieval is greatly improved by 5.2%. The accuracy of hybrid model in predicting CKD is 97.8%. In addition, the proposed hybrid model outperforms the state-of-the art models in the accuracy by 64%.

Future work will focus on applying the proposed model in different HCS applications. Different diseases will be used to evaluate the consistency of our system.

References

[1] A. Singh, M. Hemalatha, Cluster based bee algorithm for virtual machine placement in cloud data centre, JATIT 57 (3) (2013) 1–10.
[2] L. Chen, J. Zhang, L. Cai, T. Meng, MTAD: A Multitarget Heuristic Algorithm for Virtual Machine Placement, Int J Distrib Sensor Netw 2014 (2015) 1–14.
[3] R. Camati, A. Calsavara, L. Lima, Solving the virtual machine placement problem as a multiple multidimensional knapsack problem, IARIA, IEEE. (2014) 253–260.
[4] B. Suseela, V. Jeyakrishnan, A multi-objective hybrid aco-pso optimization algorithm for virtual machine placement in cloud computing, IJRET 3 (4) (2014) 474–476.
[5] J. Zhao, L. Hu, Y. Ding, G. Xu, M. Hu, A heuristic placement selection of live virtual machine migration for energy-saving in cloud computing environment, PloS ONE 9 (9) (2014) 1–13 Springer.
[6] A. Abdelaziz, M.Elhoseny, A.S. Salama, A.M. Riad and A. Hassanien, Intelligent Algorithms for Optimal Selection of Virtual Machine in Cloud Environment, Towards Enhance Healthcare Services, in: Proceedings of the International Conference on Advanced Intelligent Systems and Informatics, Springer, vol. 639, 2017, pp. 23-37.
[7] S. Sharma, Cervical cancer stage prediction using decision tree approach of machine learning, IJARCCE 5 (4) (2016) 345–348.
[8] C.B. Kumar, M.V. Kumar, T. Gayathri, S.R. Kumar, Data analysis and prediction of hepatitis using Support Vector Machine (SVM), IJCSIT 5 (2) (2014) 2235–2237.
[9] T. Prerana, N. Shivaprakash, N. Swetha, Prediction of heart disease using machine learning algorithms – Naïve Bayes, Introduction to PAC Algorithm, comparison of algorithms and HDPS, IJSE 3 (2015) 90–99.
[10] P. Tintu, R. Paulin, detect breast cancer using fuzzy c means techniques in Wisconsin Prognostic Breast Cancer (WPBC) Data Sets, IJCAT 2 (5) (2013) 614–617.
[11] A.M. Hamad, Lung cancer diagnosis by using fuzzy logic, ijcsmc 5(3) (2016) 32-41.
[12] C. Arjun, S. Anto, Diagnosis of Diabetes Using Support Vector Machine and Ensemble Learning Approach, IJEAS 2 (11) (2015) 68–72.
[13] K. Parikh, N. Hawanna, P. Haleema, R. Jayasubalakshm, Virtual machine allocation policy in cloud computing using CloudSim in Java, IJGDC 8 (1) (2015) 145–158.
[14] N.R. Darwish, A.A. Mohamed, B.S.M. Zohdy, Applying swarm optimization techniques to calculate execution time for software modules, IJARAI 5 (3) (2016) 12–17.
[15] N.R. Darwish, A.A. Mohamed, A.S. Abdelghany, A hybrid machine learning model for selecting suitable requirements elicitation techniques, IJCSIS 14 (6) (2016) 380–391.
[16] X. Fang, Z. Hui, J.D. Deng, Y. Li, T. Gu, J. Zhang, An Energy efficient ant colony system for virtual machine placement in cloud computing, Comput Intell Soc, IEEE (2016) 1–15.
[17] T.P. Shabeera, S.D. Kumar, S.M. Salam, K.M. Krishnan, Optimizing VM allocation and data placement for data-intensive applications in cloud using ACO metaheuristic algorithm, Eng Sci Technol Int J 20 (2017) 616–628.
[18] N. Almezeini, A. Hafez, Task scheduling in cloud computing using lion optimization algorithm, IJACSA 8 (2017) 77–83.
[19] R. Mishra, A. Jaiswal, Bees life algorithm for job scheduling in cloud computing, ICCIT 3 (2012) 186–191.
[20] S.J. Mohana, M. Dr, Dr Saroja, M. Venkatachalam, Comparative analysis of swarm intelligence optimization techniques for cloud scheduling, IJISET 1 (10) (2014) 15–19.
[21] A.S. Salama, A swarm intelligence based model for mobile cloud computing, IJITCS 2 (2015) 28–34.
[22] W. Sung, Y. Chiang, Improved particle swarm optimization algorithm for android medical care IOT using modified parameters, J Med Syst 36 (2012) 3755–3763 Springer.
[23] M. Seddigh, S. Sharifian, Dynamic prediction scheduling for virtual machine placement via ant colony optimization, IEEE, Amirkabir University of Technology, Tehran, IRAN, 2015, pp. 104–108.
[24] M. Vidhya, N. Mr, Sadhasivam, Parallel particle swarm optimization for reducing data redundancy in heterogeneous cloud storage, IJTET 3 (1) (2015) 73–78.
[25] T. Thiruvenkadam, P. Kamalakkannan, Virtual machine placement and load rebalancing algorithms in cloud computing systems, IJESRT 5 (8) (2016) 346–359.
[26] E. Barlaskara, Y. Jayanta, B. Issac, Energy-efficient virtual machine placement using enhanced firefly algorithm, IJMGS 12 (2016) 167–198.
[27] H. Teyeb, A. Balma, N.H. Alouane, Optimal virtual machine placement in large-scale cloud systems, Anchorage, IEEE (2014).
[28] N. Chaurasia, S. Tapaswi, J. Dhar, A pareto optimal approach for optimal selectionof virtual machine for migration in cloud, IJCSIS 14 (2016) 117–122.
[29] X. Fu, C. Zhou, Virtual machine selection and placement for dynamic consolidation in Cloud computing environment, ICAMT 9 (2015) 322–330.
[30] A. Shrivastava, V. Patel, S. Rajak, An energy efficient VM allocation using best fit decreasing minimum migration in cloud environment, IJESC 7 (1) (2017) 4076–4082.
[31] L. Jena, N. Kamila, distributed data mining classification algorithms for prediction of chronic- kidney-disease, IJERMT 4 (11) (2015) 110–118.
[32] A. Batra, V. Singh, A Review to Predictive Methodology to Diagnosis Chronic Kidney Disease, in: International Conference on Computing for Sustainable Global Development, IEEE, vol.4, 2016, pp. 2760–2763.
[33] B. Boukenze, H. Mousannif, A. Haqiq, Performance of data mining techniques to predict in healthcare case study: chronic kidney failure disease, IJDMS 8 (3) (2016) 1–9.
[34] K.R. Padmanaban, G. Parthiban, Applying machine learning techniques for predicting the risk of chronic kidney disease, ijst 9 (29) (2016) 1–5.
[35] A. Salekin, J. Stankovic, Detection of chronic kidney disease and selecting important predictive attributes, ICHI 8 (2016) 1–9.
[36] M. Vidhya, N. Sadhasivam, Parallel particle swarm optimization for reducing data redundancy in heterogeneous cloud storage, IJTET 3 (1) (2015) 73–78.
[37] A.S. Salama, A.A. Omar, A back propagation artificial neural network based model for detecting and predicting fraudulent financialreporting, IJCA 106 (2) (2014) 1–8.
[38] R. Kumar, G. Sahoo, Cloud computing simulation using CloudSim, IJETT 8 (2) (2014) 82–86.
[39] J.W. Bos, K. Lauter, M. Naehrig, Private predictive analysis on encrypted medical data, J Biomed Inform 50 (2014) 234–243.