NODE CLASSIFICATION MODEL FOR ON-THE-FLY COMPUTING BASED MOBILE GRIDS USING ROUGH SET THEORY

Amit Sadanand Savyanavar[1*], *Vijay Ram Ghorpade*[2]

[1] Shivaji University, Kolhapur; [1] School of Computer Engineering & Technology, Dr. Vishwanath Karad MIT World Peace University, Pune,
[2] Bharati Vidyapeeth's College of Engineering, Kolhapur
India

* Corresponding Author: e-mail: amitsavyanavar@gmail.com

Abstract: Rough set theory provides simple and lightweight technique to classify mobile nodes in mobile grids. Mobile devices are resource constrained and hence rough set theory can be used for classification. The purpose of our work is to devise classification approach using rough set theory to classify nodes in a mobile grid. Here decision table contains node features which are used to classify nodes into Positive, Negative and Boundary sets. Proposed approach can be deployed on Android devices with anytime-anywhere computing paradigm. The novelty of our approach is that we applied non-computationally intensive rough set theory for efficient classification of nodes.

Key words: Classification, Decision table, Disaster management, Ontology, Rough set theory.

1. INTRODUCTION

Set theory in mathematics can be applied for approximate reasoning in a variety of applications. Rough set is a tool in mathematics which uses set theory. It can be effectively utilized for classification of objects and hence widely used in decision support systems. It uses approximate reasoning to classify objects into three regions: positive, negative and boundary. So applications which involve categorization into two precise classes can be given a new dimension with the addition of an approximate boundary class.

Creating true mobile grid (MG) is now feasible. A true MG [1] is one which comprises of mobile devices only. Mobile devices in the vicinity can be networked to operate as a grid for executing computationally intensive tasks. Such tasks when executed on a single device will consume significant battery and time for completion.

So a prudent choice would be to use a MG for execution. Military applications [2, 3], remote healthcare [4–5] and disaster management are typical applications with such requirements. The major advantage of using MGs is that they work without wired infrastructure. The flipside is that it faces few challenges which are inherent to constrained resources. Mobile devices have a limited battery-life. In wireless environment, communication is unreliable and the nodes are susceptive to failures [6]. Node mobility [7] adds to their unpredictability. The nodes participating in a MG may have diverse configurations with some having higher processing power, storage space, RAM and battery. All the members of a MG might not be good enough for executing computationally intensive tasks of an application. To ensure that the performance of MG is not affected, we need to identify appropriate nodes that can execute the tasks. Hence classification as execution and non-execution nodes, aids in deploying the tasks of an application on these nodes.

A data-crunching complex application usually comprises of subtasks which can be executed parallel on a collection of computing nodes. These subtasks invariably have to communicate with each other. There is parallel and precedence execution dependency amongst these subtasks. Communication in mobile networks is unreliable and can significantly affect application performance due to failures and inefficient allocation of work to the nodes. Hence the issue of resource allocation [8] needs to be addressed in a prudent manner. MG can very easily play a role in this regard. But being resource constrained it may face few serious challenges. Our work aims at designing and evaluating classification technique that can be deployed in node resource allocation for MG. Section 2 describes related work, section 3 provides classification model details, section 4 comments on experimentation and result analysis and section 5 concludes with suggestions on future work.

2. RELATED WORK

Rough set theory (RST) was used by Wenqing Zhao et al. [10] as an email classification model. Invariably emails that get into the inbox are either classified as spam or non-spam. But there’s always a possibility of an email being neither a spam nor a non-spam. So there’s a third category of emails which can be classified as suspicious and reduces the error involved in classification. Rough set theory classifies the email into spam, non-spam and suspicious. Performance evaluation of the proposed technique corroborated that it is superior to Naive Bayes classification algorithm.

For hospice recommendation, rough set theory was used by Eleazar Gil-Herrera et al. [11] in predictive classification. Data retrieved from 9103 patients who were terminally ill was used to evaluate a model for identification of hospice candidates. Rough set approach accurately used the conditional values to determine the membership in a concept class.

Data analysis poses a challenge in the presence of vagueness and inconsistencies in datasets. This is where rough sets can help. RST analyses data in a systematic way without implicit assumptions about relationships between covariates. RST extracts information from dataset which can be represented as 'if-then-else' decision rules. In medical applications, RST aids in extracting prognostic rules from minimal available information besides extracting diagnostic rules. RST derives an information table [11] of the form S = (U: set of objects, A: set of attributes, V: set of values, f: mapping attribute to its value). Set of attributes comprises of condition attributes and decision attribute. The information table becomes a decision table when decision attribute is present. Combination of attribute and value for an object is called a descriptor.

A decision table with six prognostic factors [11] and discretized values of few patients was considered. The existence of heart disease is represented by the value Yes/ No for decision making attribute. Based on descriptor values, objects are grouped to form classes called as equivalence classes. Two objects with same descriptor values are part of indiscernibility relation. Decision table segregated objects to four groups based on the concept of equivalence classes.

$$U1=\{x1\}, U2 =\{x2\}, U3 =\{x5, x6\}, U4 =\{x3, x4, x7\} \quad (1)$$

Decision classes are formed using possible values for decision making attribute. Objects with similar value for decision attribute are grouped into decision class. Table had two decision classes based on possible values (Yes/ No) for the attribute: $C_{No} = \{x1, x2, x3\}$ and $C_{Yes} = \{x4, x5, x6, x7\}$. In a consistent decision table, objects having same attribute values should belong to the same decision class. Typically this isn't true for an inconsistent decision table. Inconsistent decision table makes use of a lower and upper approximations for a decision class. Lower-approximation comprises of objects that unquestionably are part of a decision class, whereas upper-approximation includes objects that could most likely belong to a decision class. These approximations can be determined as illustrated below:

i) Find the R-indiscernibility relation which consists of equivalence classes $[x]_R$

$$R(C) = \{(o_i, o_j) \in U : \forall a \in A, f(o_i, a) = f(o_j, a) \} \quad (2)$$

ii) Determine set of objects whose decision attribute d=Yes

$$C_{yes} = \{ o |\ \forall o \in U, d = Yes\} \quad (3)$$

iii) Derive the lower-approximation of C_{yes}

$$\underline{R}C_{yes} = \{o \in U : [o]R \subseteq C_{yes}\} \quad (4)$$

iv) Derive the upper-approximation of C_{yes}

$$\overline{R}C_{yes} = \{o \in U : [o]R \cap C_{yes} \neq \phi \} \quad (5)$$

v) Boundary region includes those objects that are possibly but not certainly part of d=Yes membership class. Boundary region of C_{yes} is

$$BND = \overline{R}C_{yes} - \underline{R}C_{yes} \quad (6)$$

Mohit Jain et al. [12] proposed a model using RST to detect malicious nodes in a mobile adhoc network. Simple rules were derived from attributes to distinguish good nodes from bad ones. Results corroborate that throughput and network capacity improved to 98.9%.

3. RST BASED NODE CLASSIFICATION MODEL

Table 1. Symbols and their description

Symbol	*Description*
M	*Total number of nodes in MG network*
m_i	*Node i in MG* $(m_i \in M)$
S_{m_i}	*Score for a node i*
C	*Set of condition attributes (Processing capability with % CPU utilization, storage space, RAM availability, residual battery and strength of connectivity)*
d	*Decision attribute(Executor_node)*
c_n	*Total count of attributes used to determine score*
c_v	*Condition attribute value*

Mobile grid resource allocation (MGRA) [13, 15] is a resource allocation technique that needs to classify its nodes for efficient allocation of tasks of an application. This is where a lightweight classification model is required. RST is used to fill this gap. Being lightweight it doesn't drain the resource constrained mobile devices. The node attributes that can be utilized for classification are processing capability with % CPU utilization, storage space, RAM availability, residual battery and strength of connectivity. RST classifies the nodes into positive, boundary and negative sets. Nodes in positive category can definitely be used for task execution, whereas nodes in negative category should not be used at all. If required the nodes in boundary category can be used.

Table 2. Decision rules

Rule	*Execution_node*
if $S_{n_i} \geq 15$	*Yes*
if $(S_{n_i} \geq 9$ *&&* $S_{n_i} \leq 14)$ *AND* *if((%_Residual_battery≤30 && Connectivity_strength ≥ 10)*	*No*
if $(S_{n_i} \geq 9$ *&&* $S_{n_i} \leq 14)$ *AND* *rule 2 is false*	*Yes*
if all above rules are false	*No*

Decision_table_creation()

{

Step 1: Prepare information table which includes the input attribute values c_v for each node in MG

Step 2: Use table 3 to discretize each attribute value

Step 3: Fill these discretized values in the table

Step 4: Convert it into decision table by adding decision attribute (*Executor_node*)

Step 5: Find and fill score for each node i=1…. M using

$$s_{m_i} = \sum_{j=1}^{c_n} c_{v_{ij}}$$

Step 6: Derive the value of decision node using the score value and decision rules from Table 2

Step 7: Return decision table

}

Fig. 1. Decision table creation procedure

RST based Classification Algorithm

{

Step 1: Determine values of condition attributes

Step 2: Invoke Decision_table_creation() routine to which condition attribute values are passed

Step 3: Using decision table and indiscernibility relation, find out equivalence classes $[m]_E$

$$E = \{(m_i, m_j) \in M : \forall c \in C, f(m_i, c) = f(m_j, c)\}$$

Equivalence class comprises of nodes which have similar condition attribute values

Step 4: Determine the set of nodes where d=Yes

$$M_{yes} = \{ m / \forall m \in M, d = Yes\}$$

Step 5: Derive the lower approximation of M_{yes}

$$\underline{E}M_{yes} = \{m \in M : [m]_E \subseteq M_{yes}\}$$

Step 6: Derive the upper approximation of M_{yes}

$$\bar{E}M_{yes} = \{m \in M : [m]_E \cap M_{yes} \neq \phi\}$$

Step 7: Positive region for M is

$$POS = \underline{E}\, M_{yes}$$

Step 8: Boundary region for M is

$$BND = \bar{E}M_{yes} - \underline{E}M_{yes}$$

Step 9: Negative region for M is

$$NEG = M - \bar{E}M_{yes}$$

Step 10: Return all the derived *POS, BND, NEG* sets

}

Fig. 2. RST based node classification algorithm

Table 3. Discretized attributes

Attribute	***Categorization***	***Description***	***Attribute value***
Processing_capability with <30% CPU utilization	*Octa core and above*	*High*	*3*
	Dual & quad core	*Medium*	*2*
	Single core	*Low*	*1*
Storage_space	*> 16GB*	*High*	*3*
	8GB to 16GB	*Medium*	*2*
	<8GB	*Low*	*1*
RAM_availability	*> 2GB*	*High*	*3*
	1.5GB to 2GB	*Medium*	*2*
	< 1.5GB	*Low*	*1*
%_Residual_battery	*> 65*	*High*	*6*
	25 to 65	*Medium*	*3*
	< 25	*Low*	*1*
Connectivity_strength	*< 8seconds*	*Strong*	*6*
	8 to 12 seconds	*Moderate*	*3*
	> 12 seconds	*Weak*	*1*

Table 4. Decision Table illustration

Node	**Processing_ capability**	**Storage_space**	**RAM_ availability**	**%_Residual_battery**	**Connectivity_ strength**	**Score**	**Executor _node**
m_1	*2*	*2*	*1*	*3*	*3*	*11*	*No*
m_2	*2*	*2*	*1*	*3*	*3*	*11*	*Yes*
m_3	*1*	*1*	*1*	*1*	*1*	*5*	*No*
m_4	*1*	*1*	*2*	*6*	*6*	*16*	*Yes*
m_5	*3*	*3*	*3*	*6*	*6*	*21*	*Yes*
m_6	*3*	*3*	*2*	*3*	*6*	*17*	*Yes*
m_7	*2*	*2*	*2*	*1*	*1*	*8*	*No*
m_8	*2*	*3*	*2*	*1*	*6*	*14*	*Yes*

In Table 4, m_1 & m_2 have same discretized attribute values scores but %_Residual_battery = 30 for m_1 and so as per decision rules it will not qualify as an executor node by our algorithm.

$$E = \{(m_1, m_2), (m_3), (m_4), (m_5), (m_6), (m_7), (m_8)\} \quad (7)$$

$$M_{yes} = \{m_2, m_4, m_5, m_6, m_8\} \quad (8)$$

$$\underline{E}M_{yes} = \{m_4, m_5, m_6, m_8\} \quad (9)$$

$$\overline{E}M_{yes} = \{m_1, m_2, m_4, m_5, m_6, m_8\} \quad (10)$$

$$POS = \{m_4, m_5, m_6, m_8\} \quad (11)$$

$$BND = \{m_1, m_2\} \quad (12)$$

$$NEG = \{m_3, m_7\} \quad (13)$$

4. EXPERIMENTATION AND ANALYSIS

Sets of experiments were conducted for establishing the effectiveness of our RST based classification approach. MG was created using Android smartphones connected with Wi-Fi Direct. Disaster management being a perfect scenario requiring a MG, we tested our algorithm on an application related to it. When disaster strikes, network infrastructure for communication may be destroyed and internet connectivity is lost. During such situations, a MG using Wi-Fi direct or any other protocol without using the fixed network infrastructure can be formed which can aid in expediting the recovery process. The mobile devices in MG may hoard information which could be mined to determine what specific recovery steps need to be taken at different locations to save human life. So as a case in point we tested an approach for summarizing multiple documents based on ontology [9] using our MG. Multidocument summarization technique involves a number of subtasks like: distributing files amongst the available nodes, identifying sentences from files, distributing sentence identified files amongst the available nodes, mapping sentences to corresponding concepts and storing them in corresponding concept file, taking query from user and displaying summary related to queried concept.

We compared our proposed approach with Distance based resource allocation scheme (DRA) [14] and Next location based resource allocation scheme (NLRA) [14]. DRA looks for an executor node to allocate task based on its proximity with the current node. Unfortunately DRA doesn't search for a collection of nodes to assign dependent tasks. NLRA ignores distance between the nodes and performs allocation randomly. Both these approaches do not consider the characteristics of nodes before allocating tasks. Experiments were conducted considering different scenarios based on the arrangement of nodes in MG and the amount of data in the collection of files. 10 smartphones: Micromax canvas A1, Samsung Duos, Redmi Mi4(2 nos.), Redmi Note 4, Redmi Mi A1, Gionee P5W, Samsung J6, Asus Zenfone 5 and Lenovo k8 were used with % residual battery: 70, 39, 59, 44, 73, 28, 11, 34, 10 and 69 respectively. With MGRA, RST based classification algorithm is used to identify nodes which qualify as execution nodes.

4.1. Scenario 1

In scenario 1, aggregate file size is 2MB and nodes were spread at a distance up to 3 meters. MGRA requires almost 26% lesser time than DRA and NLRA for application execution as depicted in Figure 3. The amount of battery consumed for MGRA is almost 75% less as compared to other methods as seen in Figure 4.

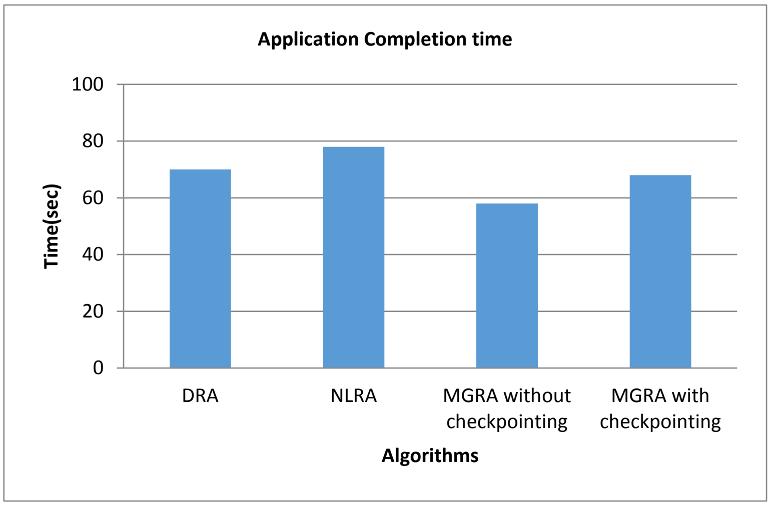

Fig. 3. Scenario 1 – Application Completion Time

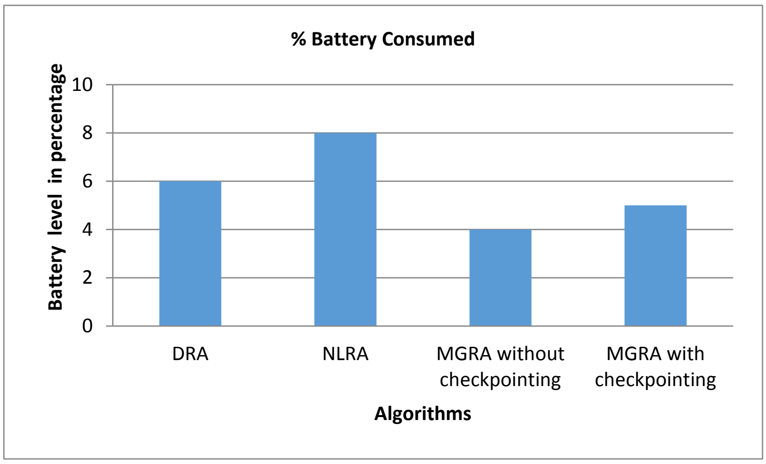

Fig. 4. Scenario 1 – % Battery Consumed

4.2. Scenario 2

In scenario 2, aggregate file size is 2MB and nodes are spread from 10-12m. MGRA requires almost 25% lesser time compared with other methods to complete application execution shown in Figure 5. Figure 6 shows that the amount of battery consumed for MGRA is almost 50% less as compared to other methods. It can be interpreted that the performance gap keeps on widening as the amount of data involved in computation increases.

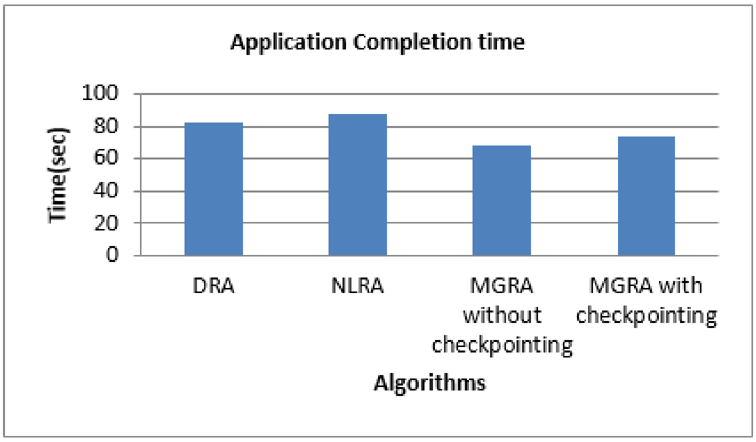

Fig. 5. Scenario 2 – Application Completion Time

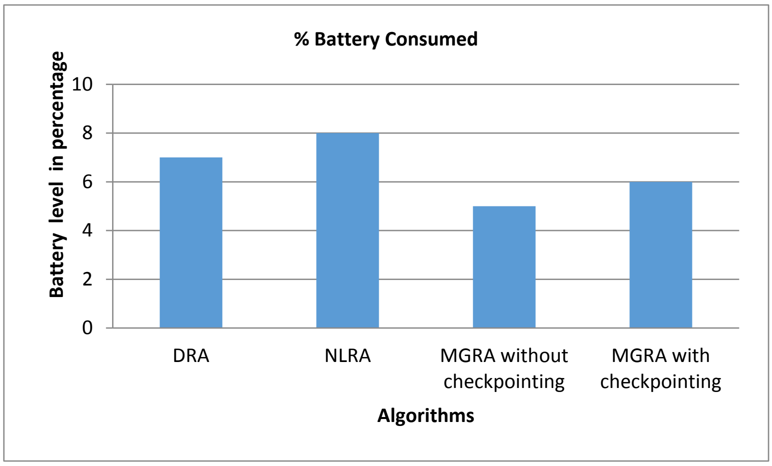

Fig. 6. Scenario 2 – % Battery Consumed

4.3. Scenario 3

In scenario 3, aggregate file size is 2MB and nodes spread at a distance of 20-22m. MGRA requires almost 23% lesser time for application execution as depicted in Figure 7. Amount of battery consumed for MGRA is almost 50% less as compared to other methods seen in Figure 8. It can be interpreted that the performance gap keeps on widening as the amount of data involved in computation increases.

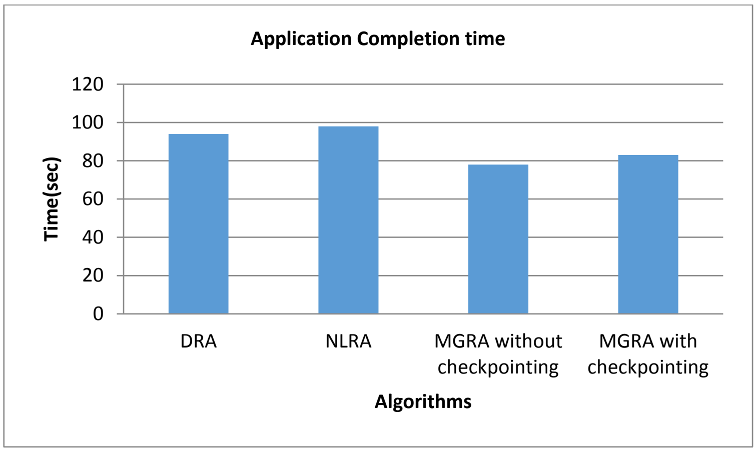

Fig. 7. Scenario 3 – Application Completion Time

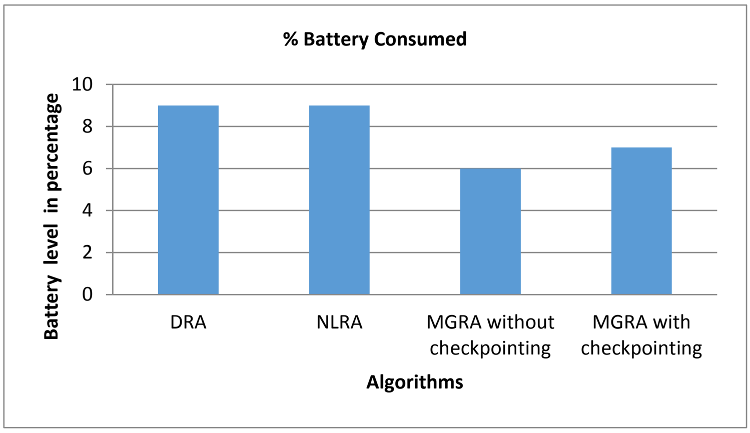

Fig. 8. Scenario 3 – % Battery Consumed

4. CONCLUSION

RST is truly a lightweight approximation tool in mathematics for classification of objects on resource constrained devices. Using simple set theory, it classifies objects into three sets. Decision support systems can use such theory to make appropriate prudent decisions for tremendous gains. RST is a rule-based approach which outperforms non-rule based methods. Boundary set will aid in reducing the error of classifying a possible object from positive set as being in negative set. As can be seen from our work, RST based classification approach can result in approximately 28% lesser application completion time and 62% savings on battery consumption as compared to existing approaches. Also, classification can help identify appropriate devices rather than simply using all the available devices for computation. As a part of future work, hybrid methods developed over the years for RST along with appropriate optimization approaches like set cover [16] can be explored to improve performance of resource constrained devices.

REFERENCES

[1] Amit Sadanand Savyanavar, Dr. Vijay R. Ghorpade, A survey on resource allocation in mobile grid computing, *International Journal of Wireless Communication*, ISSN: 0974-9640, Vol. 5, No. 5, 2013, pp. 212-216, http://www.ciitresearch.org/dl/index.php/wc/article/view/WC052013004

[2] Samad T., Bay J. S., Godbole D., Network-centric systems for military operations in urban terrain: the role of UAVs, *Proceedings of the IEEE*, Vol. 95, No. 1, 2007, pp. 92–107, https://doi.org/10.1109/JPROC.2006.887327.

[3] S. Se, D.G. Lowe, J. J. Little, Vision-based global localization and mapping for mobile robots, *IEEE Transactions on Robotics*, ISSN: 1941-0468, Vol. 21, No. 3, 2005, pp. 364–375, DOI: 10.1109/TRO.2004.839228.

[4] H. Viswanathan, B. Chen, and D. Pompili, Research Challenges in Computation, Communication, and Context awareness for Ubiquitous Healthcare, *IEEE Communications*, ISSN:1558-1896, Vol. 50, No. 5, 2012, pp. 92 –99, DOI: 10.1109/MCOM.2012.6194388.

[5] Silvia D. Olabarriaga, Tristan Glatard, Piter T. de Boer, A Virtual Laboratory for Medical Image Analysis, *IEEE Transactions on Information Technology in Biomedicine*, ISSN:1558-0032, Vol. 14, No. 4, 2010, pp. 979-985, DOI: 10.1109/TITB.2010.2046742

[6] Amit Sadanand Savyanavar, Dr. Vijay R. Ghorpade, Application Checkpointing Technique for Self-healing from Failures in Mobile Grid Computing, *IGI - International Journal of Grid and High Performance Computing*, ISSN: 1938-0259, Vol. 11, No. 2, 2019, pp. 50-62.

[7] Amit Sadanand Savyanavar, Dr. Vijay R. Ghorpade, Node Mobility Prediction in Mobile Grid Computing, *International Journal on Emerging Trends in Technology*, ISSN: 2455-0124, Vol. 2, No. 2, 2015, pp. 342-347.

[8] Amit Sadanand Savyanavar, Ashwini Patil, Dr. Vijay R. Ghorpade, Mobility prediction and resource allocation in mobile grid computing, *International Journal of Computer Engineering and Applications*, ISSN 2321-3469, Vol. 12, 2018.

[9] L. Li and T. Li, "An Empirical Study of Ontology-Based Multi-Document Summarization in Disaster Management, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 44, No. 2, 2014, pp. 162-171, DOI: 10.1109/TSMCC.2013.2258335.

[10] Wenqing Zhao and Zili Zhang, An email classification model based on rough set theory, *Proceedings of the 2005 International Conference on Active Media Technology*, ISBN:0-7803-9035-0, 2005, DOI: 10.1109/AMT.2005.1505383

[11] Eleazar Gil-Herrera, Garrick Aden-Buie, Ali Yalcin, Athanasios Tsalatsanis, Laura E. Barnes and Benjamin Djulbegovic, Rough set theory based prognostic classification models for hospice referral, *BMC Medical Informatics & Decision Making*, ISSN: 1472-6947, Vol. 98, 2015, DOI 10.1186/s12911-015-0216-9.

[12] Mohit Jain and M.P.S Bhatia, A Rough Set Based Approach to Classify Node Behavior in Mobile Ad Hoc Networks, *Journal of mathematics and computer science*, ISSN 2008-949X, Vol. 11, No. 1, pp 64-78, 2014, http://dx.doi.org/10.22436/jmcs.011.01.07.

[13] Amit Sadanand Savyanavar, Dr. Vijay R. Ghorpade, Resource allocation scheme for Dew computing paradigm using mobile grids, *International Journal of Innovative Technology and Exploring Engineering*, ISSN: 2278-3075, Vol. 8, No. 9, 2019, pp. 199-203, DOI: 10.35940/ijitee.H6545.078919

[14] Shah S. C., Chauhdary S.H., Bashir A.K., Park M.S., A centralized location-based job scheduling algorithm for interdependent jobs in mobile adhoc computational Grids, Springer, *Journal of Applied Sciences*, ISSN: 1812-5662, Vol. 10, No. 3, 2010, pp.174-181, DOI: 10.3923/jas.2010.174.181

[15] Amit Sadanand Savyanavar, Vijay R. Ghorpade, Efficient resource allocation scheme for on-the-fly computing based mobile grids, *International Journal of Information Technology*, ISSN: 2511-2112, 2018, https://doi.org/10.1007/s41870-018-0269-y

[16] Broderick C., Ricardo S., Cristian P., Marco R., Claudio T., Sanjay M., Franklin J., Fernando P., A Comparison of Three Recent Nature-Inspired Metaheuristics for the Set Covering Problem, *Proc. of the International Conference on Computational Science and Its Applications*, ISBN: 978-3-319-21410-8, Vol. 9158, 2015, https://doi.org/10.1007/978-3-319-21410-8_34

Information about the authors:

Amit Sadanand Savyanavar – Received M.E. and pursuing PhD degree in CSE from Shivaji University, Kolhapur, India. Currently working as Associate Professor at Dr. Vishwanath Karad MIT World Peace University, Pune, India. Areas of research are mobile computing, operating systems and distributed systems.

Vijay Ram Ghorpade – Received M.E. and PhD degree in Computer Engineering. Currently working as Principal at Bharati Vidyapeeth's College of Engineering and Technology, Kolhapur, India. Areas of research are mobile adhoc networks and information security

Manuscript received on 05 July 2021