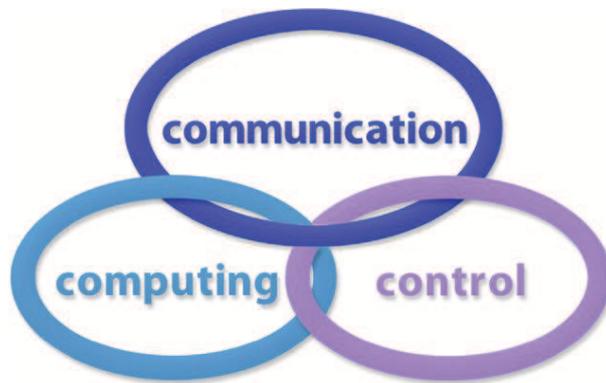


INTERNATIONAL JOURNAL
of
COMPUTERS COMMUNICATIONS & CONTROL

ISSN 1841-9836



A Bimonthly Journal
With Emphasis on the Integration of Three Technologies

Year: 2017 Volume: 12 Issue: 5 Month: October

This journal is a member of, and subscribes to the principles of, the Committee on Publication Ethics (COPE).



<http://univagora.ro/jour/index.php/ijccc/>

CCC Publications

Copyright © 2006-2017 by Agora University

BRIEF DESCRIPTION OF JOURNAL

Publication Name: International Journal of Computers Communications & Control.

Acronym: IJCCC; **Starting year of IJCCC:** 2006.

ISO: Int. J. Comput. Commun. Control; **JCR Abbrev:** INT J COMPUT COMMUN.

International Standard Serial Number: ISSN 1841-9836.

Publisher: CCC Publications - Agora University of Oradea.

Publication frequency: Bimonthly: Issue 1 (February); Issue 2 (April); Issue 3 (June); Issue 4 (August); Issue 5 (October); Issue 6 (December).

Founders of IJCCC: Ioan DZITAC, Florin Gheorghe FILIP and Misu-Jan MANOLESCU.

Indexing/Coverage:

- Since 2006, Vol. 1 (S), IJCCC is covered by Thomson Reuters/Clarivate Analytics and is indexed in ISI Web of Science/Knowledge: Science Citation Index Expanded.

2017 Journal Citation Reports® Science Edition (Thomson Reuters, 2016):

Subject Category: (1) Automation & Control Systems: Q4(2009, 2011, 2012, 2013, 2014, 2015), Q3(2010, 2016); (2) Computer Science, Information Systems: Q4(2009, 2010, 2011, 2012, 2015), Q3(2013, 2014, 2016).

Impact Factor/3 years in JCR: 0.373(2009), 0.650 (2010), 0.438(2011); 0.441(2012), 0.694(2013), 0.746(2014), 0.627(2015), **1.374(2016)**.

Impact Factor/5 years in JCR: 0.436(2012), 0.622(2013), 0.739(2014), 0.635(2015), **1.193(2016)**.

- Since 2008 IJCCC is indexed by Scopus: **CiteScore 2016 = 1.06**.

Subject Category:

(1) Computational Theory and Mathematics: Q4(2009, 2010, 2012, 2015), Q3(2011, 2013, 2014, 2016);

(2) Computer Networks and Communications: Q4(2009), Q3(2010, 2012, 2013, 2015), Q2(2011, 2014, 2016);

(3) Computer Science Applications: Q4(2009), Q3(2010, 2011, 2012, 2013, 2014, 2015, 2016).

SJR: 0.178(2009), 0.339(2010), 0.369(2011), 0.292(2012), 0.378(2013), 0.420(2014), 0.263(2015), 0.319(2016).

- Since 2007, 2(1), IJCCC is indexed in EBSCO.

Focus & Scope: International Journal of Computers Communications & Control is directed to the international communities of scientific researchers in computer and control from the universities, research units and industry.

To differentiate from other similar journals, the editorial policy of IJCCC encourages the submission of original scientific papers that focus on the integration of the 3 "C" (Computing, Communication, Control).

In particular the following topics are expected to be addressed by authors: (1) Integrated solutions in computer-based control and communications; (2) Computational intelligence methods (with particular emphasis on fuzzy logic-based methods, ANN, evolutionary computing, collective/swarm intelligence); (3) Advanced decision support systems (with particular emphasis on the usage of combined solvers and/or web technologies).

IJCCC EDITORIAL TEAM

Editor-in-Chief: Florin-Gheorghe FILIP

Member of the Romanian Academy
Romanian Academy, 125, Calea Victoriei
010071 Bucharest-1, Romania, ffilip@acad.ro

Associate Editor-in-Chief: Ioan DZITAC

Aurel Vlaicu University of Arad, Romania
St. Elena Dragoi, 2, 310330 Arad, Romania
ioan.dzitac@uav.ro

&

Agora University of Oradea, Romania
Piata Tineretului, 8, 410526 Oradea, Romania
rector@univagora.ro

Managing Editor: Mişu-Jan MANOLESCU

Agora University of Oradea, Romania
Piata Tineretului, 8, 410526 Oradea, Romania
mmj@univagora.ro

Executive Editor: Răzvan ANDONIE

Central Washington University, U.S.A.
400 East University Way, Ellensburg, WA 98926, USA
andonie@cwu.edu

Reviewing Editor: Horea OROS

University of Oradea, Romania
St. Universitatii 1, 410087, Oradea, Romania
horos@uoradea.ro

Layout Editor: Dan BENTA

Agora University of Oradea, Romania
Piata Tineretului, 8, 410526 Oradea, Romania
dan.benta@univagora.ro

Technical Secretary

Domnica Ioana DZITAC

New York University Abu Dhabi, UAE
did233@nyu.edu

Simona DZITAC

R & D Agora, Romania
simona@dzitac.ro

Editorial Address:

Agora University/ R&D Agora Ltd. / S.C. Cercetare Dezvoltare Agora S.R.L.
Piata Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel./ Fax: +40 359101032
E-mail: ijccc@univagora.ro, rd.agora@univagora.ro
Journal Website: <http://univagora.ro/jour/index.php/ijccc/>

IJCCC EDITORIAL BOARD MEMBERS

Luiz F. Autran M. GOMES

Ibmec, Rio de Janeiro, Brasil
Av. Presidente Wilson, 118
autran@ibmecrj.br

Boldur E. BĂRBAT

Sibiu, Romania
bbarbat@gmail.com

Pierre BORNE

Ecole Centrale de Lille, France
Villeneuve d'Ascq Cedex, F 59651
p.borne@ec-lille.fr

Ioan BUCIU

University of Oradea
Universitatii, 1, Oradea, Romania
ibuciu@uoradea.ro

Hariton-Nicolae COSTIN

University of Medicine and Pharmacy,
St. Universitatii No.16, 6600 Iași, Romania
hcostin@iit.tuiasi.ro

Petre DINI

Concordia University
Montreal, Canada
pdini@cisco.com

Antonio Di NOLA

Dept. of Math. and Information Sci.
Università degli Studi di Salerno
Via Ponte Don Melillo, 84084 Fisciano, Italy
dinola@cds.unina.it

Yezid DONOSO

Universidad de los Andes
Cra. 1 Este No. 19A-40
Bogota, Colombia, South America
ydonoso@uniandes.edu.co

Gintautas DZEMYDA

Vilnius University
Akademijos St. 4
Vilnius, LT-08663, Lithuania
gintautas.dzemyda@mii.vu.lt

Ömer EGECIOGLU

University of California
Santa Barbara, CA 93106-5110, U.S.A.
omer@cs.ucsb.edu

Constantin GAINDRIC

Institute of Mathematics of
Moldavian Academy of Sciences
Kishinev, 277028, Academiei 5
Moldova, Republic of
gaindric@math.md

Xiao-Shan GAO

Acad. of Math. and System Sciences
Academia Sinica
Beijing 100080, China
xgao@mmrc.iss.ac.cn

Enrique HERRERA-VIEDMA

University of Granada
Granada, Spain
viedma@decsai.ugr.es

Kaoru HIROTA

Hirota Lab. Dept. C.I. & S.S.
Tokyo Institute of Technology
G3-49,4259 Nagatsuta, Japan
hirota@hrt.dis.titech.ac.jp

Geang KOU

Southwestern University of Finance and Economics
Chengdu, 611130, China
kougang@swufe.edu.cn

George METAKIDES

University of Patras
Patras 26 504, Greece
george@metakides.net

Shimon Y. NOF

School of Industrial Engineering
Purdue University
Grissom Hall, West Lafayette, IN 47907
U.S.A.
nof@purdue.edu

Stephan OLARIU

Department of Computer Science
Old Dominion University
Norfolk, VA 23529-0162, U.S.A.
olariu@cs.odu.edu

Gheorghe PĂUN

Institute of Math. of Romanian Academy
Bucharest, PO Box 1-764, Romania
gpaun@us.es

Mario de J. PEREZ JIMENEZ
Dept. of CS and Artificial Intelligence
University of Seville, Sevilla,
Avda. Reina Mercedes s/n, 41012, Spain
marper@us.es

Dana PETCU
Computer Science Department
Western University of Timisoara
V.Parvan 4, 300223 Timisoara, Romania
petcu@info.uvt.ro

Radu POPESCU-ZELETIN
Fraunhofer Institute for Open
Communication Systems
Technical University Berlin, Germany
rpz@cs.tu-berlin.de

Radu-Emil PRECUP
Faculty of Automation and Computers
Politehnica University of Timisoara
VBd. V. Parvan 2, 300223 Timisoara, Romania
radu.precup@aut.upt.ro

Imre J. RUDAS
Óbuda University
Budapest, Hungary
rudas@bmf.hu

Yong SHI
School of Management
Chinese Academy of Sciences
Beijing 100190, China
yshi@gucas.ac.cn, yshi@unomaha.edu

Athanasios D. STYLIADIS
University of Kavala
Institute of Technology
65404 Kavala, Greece
styliadis@teikav.edu.gr

Gheorghe TECUCI
Learning Agents Center
George Mason University
U.S.A.
University Drive 4440, Fairfax VA
tecuci@gmu.edu

Horia-Nicolai TEODORESCU
Faculty of Electronics and
Telecommunications
Technical University "Gh. Asachi" Iasi
Iasi, Bd. Carol I 11, 700506, Romania
hteodor@etc.tuiasi.ro

Dan TUFIS
Research Institute for Artificial Intelligence
of the Romanian Academy
Bucharest, "13 Septembrie" 13, 050711, Romania
tufis@racai.ro

Lotfi A. ZADEH
Director,
Berkeley Initiative in Soft Computing (BISC)
Computer Science Division
University of California Berkeley,
Berkeley, CA 94720-1776
U.S.A.
zadeh@eecs.berkeley.edu

DATA FOR SUBSCRIBERS

Supplier: Cercetare Dezvoltare Agora Srl (Research & Development Agora Ltd.)

Fiscal code: 24747462

Headquarter: Oradea, Piata Tineretului Nr.8, Bihor, Romania, Zip code 410526

Bank: BANCA COMERCIALA FERROVIARA S.A. ORADEA

Bank address: P-ta Unirii Nr. 8, Oradea, Bihor, România

IBAN Account for EURO: RO50BFER248000014038EU01

SWIFT CODE (eq.BIC): BFER

Contents

Proportional-Integral-Derivative Gain-Scheduling Control of a Magnetic Levitation System C.-A. Bojan-Dragos, R.-E. Precup, M.L. Tomescu, S. Preitl, O.-M. Tanasoiu, S. Hergane	599
Benchmarking of Recommendation Trust Computation for Trust/Trustworthiness Estimation in HDNs H. El-Sayed, S. Sankar, H. Yu, G. Thandavarayan	612
An Uncertainty Measure for Interval-valued Evidences W. Jiang, S. Wang	631
Phasing of Periodic Tasks Distributed over Real-time Fieldbus S.-H. Lee, H.-W. Jin, K. Kim, S. Lee	645
Mining Users' Preference Similarities in E-commerce Systems Based on Webpage Navigation Logs P. Li, C.X. Wu, S.Z. Zhang, X.W. Yu, H.D. Zhong	661
A New Deep Learning Approach for Anomaly Base IDS using Memetic Classifier S. Mohammadi, A. Namadchian	677
Gravitation Theory Based Model for Multi-Label Classification L. Peng, Y. Liu	689
Delay Tolerant Networks over Near Field Communications: The Automatic Multi-packet Communication R.O. Schoeneich, P. Sadlo	704
Fast Motion Estimation Algorithm using Hybrid Search Patterns for Video Streaming Application R. Vani, N. Ushabhanu, M. Sangeetha	715
A Hybrid Social Network-based Collaborative Filtering Method for Personalized Manufacturing Service Recommendation S. Zhang, W.T. Yang, S. Xu, W.Y. Zhang	728
Author index	741

Proportional-Integral-Derivative Gain-Scheduling Control of a Magnetic Levitation System

C.-A. Bojan-Dragos, R.-E. Precup, M.L. Tomescu,
S. Preitl, O.-M. Tanasoiu, S. Hergane

**Claudia-Adina Bojan-Dragos, Radu-Emil Precup,
Stefan Preitl, Oana-Maria Tanasoiu, Stefania Hergane**

Politehnica University of Timisoara
Department of Automation and Applied Informatics
Bd. V. Parvan 2, 300223 Timisoara, Romania
claudia.dragos@aut.upt.ro, radu.precup@upt.ro, stefan.preitl@upt.ro,
oanatanasoiu96@gmail.com, stefania.hergane@yahoo.com

Marius L. Tomescu*

Aurel Vlaicu University of Arad
Romania, 310330 Arad, Elena Dragoi, 2
*Corresponding author: tom_uav@yahoo.com

Abstract: The paper presents a gain-scheduling control design procedure for classical Proportional-Integral-Derivative controllers (PID-GS-C) for positioning system. The method is applied to a Magnetic Levitation System with Two Electromagnets (MLS2EM) laboratory equipment, which allows several experimental verifications of the proposed solution. The nonlinear model of MLS2EM is linearized at seven operating points. A state feedback control structure is first designed to stabilize the process. PID control and PID-GS-C structures are next designed to ensure zero steady-state control error and bumpless switching between PID controllers for the linearized models. Real-time experimental results are presented for validation.

Keywords: gain-scheduling, magnetic levitation system, Proportional-Integral-Derivative control; real-time experiments.

1 Introduction

During the last years several classical and adaptive control structures have been proposed for positioning control system and applications for magnetic levitation systems. Some of these structures are presented in [49], [52], [1], [35], [47], [56], [11], [27], [23], [19], [55], [5]. For example, a high gain adaptive output feedback controller is designed in [5] by introducing two different virtual filters and using back-stepping. Another adaptive control scheme that copes with the modifications of the structural parameters of magnetic levitation systems is suggested in [1].

Gain-scheduling control solutions are popular nowadays, and they are briefly analyzed as follows: fuzzy-based gain scheduling of exact feed-forward linearization control and sliding mode controllers for magnetic ball levitation system are proposed in [28]. A high gain adaptive output feedback control to a magnetic levitation system is discussed in [29]. A Proportional-Integral-Derivative (PID) gain-scheduling controller for second order linear parameter varying, which exclude time varying delay using a Smith predictor is given in [44]. Other interesting adaptive gain scheduling control techniques for real practical applications are given in [6], [54], [7], [17], [53].

The paper is dealing with the position control of a ferromagnetic sphere in a Magnetic Levitation System with Two Electromagnets (MLS2EM) laboratory equipment. A state feedback control structure (SFCS) is first designed in order to stabilize the system by applying the control signal only to the top electromagnet [8]. The simulated external disturbance can be applied

to the bottom electromagnet. Starting with [9] the SFCS is next controlled by a Proportional-Integral-Derivative Gain-Scheduling Control (PID-GS-C) structure. In the presented scheme, the proportional, derivative and integral gains are adapted to the modifications of the operating points. The paper proposes relatively simple classical and adaptive control structures which belongs to the general case of linear and nonlinear control system structures [43], [38], [50], [20], [42], [51], [32], [18], [36].

The paper is organized as follows: the nonlinear model and the linearized mathematical model (MM) of MLS2EM are given in Section 2. The proposed control structure is next developed in Section 3. The real-time experimental results are presented in Section 4 and the conclusions are highlighted in Section 5.

2 Process modeling

The controlled MLS2EM laboratory equipment includes: two electromagnets (EM1 - the top electromagnet and EM2 - the bottom electromagnet), the ferromagnetic sphere, sensors to detect position of the sphere, computer interface, drivers, power supply unit, connection cables and an acquisition board. The nonlinear state-space MM of ML2SEM is [24]:

$$\begin{aligned} \dot{p}(t) &= v(t), \\ \dot{v}(t) &= -\frac{i_{EM1}^2(t) \cdot F_{emP1} \cdot \exp(-p(t)/F_{emP2})}{m \cdot F_{emP2}} + g + \frac{i_{EM2}^2(t) \cdot F_{emP1} \cdot \exp(-(x_d - p(t))/F_{emP2})}{m \cdot F_{emP2}}, \\ \dot{i}_{EM1}(t) &= \frac{k_i \cdot u_{EM1}(t) + c_i - i_{EM1}(t)}{\frac{f_{iP1}}{f_{iP2}} \cdot \exp(-p(t)/f_{iP2})}, \\ \dot{i}_{EM2}(t) &= \frac{k_i \cdot u_{EM2}(t) + c_i - i_{EM2}(t)}{\frac{f_{iP1}}{f_{iP2}} \cdot \exp(-(x_d - p(t))/f_{iP2})}, \\ y(t) &= k_m \cdot p(t), \end{aligned} \quad (1)$$

where: $p \in [0, 0.0016]$ - the sphere position (m), $v \in \mathfrak{R}$ - the sphere speed (m/s), $i_{EM1}, i_{EM2} \in [0.03884, 2.38]$ - the currents in the top electromagnet (EM1) and bottom electromagnet (EM2) (A), $u_{EM1}, u_{EM2} \in [0.005, 1]$ - the signals applied to EM1 and EM2, respectively (V), and y - the process output (m), i.e., the measured sphere position. The parameters of the process are determined analytically and experimentally [24], [16].

The model (1) is linearized around seven operating points (o.p.s) with the coordinates $P^{(j)}(p^{(j)}, v^{(j)}, i_{EM1}^{(j)}, i_{EM2}^{(j)})$ where j is the index of the current operating point, $j = 1 \dots 7$. The linearized state-space models and their matrices are:

$$\begin{aligned} \Delta \dot{\mathbf{x}}^{(j)} &= \mathbf{A}^{(j)} \Delta \mathbf{x}^{(j)} + \mathbf{b}^{(j)} \Delta \mathbf{u}^{(j)}, \\ \Delta y^{(j)} &= \mathbf{c}^{T(j)} \Delta \mathbf{x}^{(j)}, \\ \Delta \mathbf{x}^{(j)} &= [\Delta p^{(j)} \ \Delta v^{(j)} \ \Delta i_{EM1}^{(j)} \ \Delta i_{EM2}^{(j)}]^T, \quad \Delta \mathbf{u}^{(j)} = [\Delta u_{EM1}^{(j)} \ \Delta u_{EM2}^{(j)}]^T, \\ \mathbf{A}^{(j)} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{21}^{(j)} & 0 & a_{23}^{(j)} & a_{24}^{(j)} \\ a_{31}^{(j)} & 0 & a_{33} & 0 \\ a_{41}^{(j)} & 0 & 0 & a_{44}^{(j)} \end{bmatrix}, \mathbf{b}^{(j)} = [\mathbf{b}_{u_{EM1}}^{(j)} \ \mathbf{b}_{u_{EM2}}^{(j)}] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ b_{31}^{(j)} & 0 \\ 0 & b_{42}^{(j)} \end{bmatrix}, \mathbf{c}^{T(j)} = [1 \ 0 \ 0 \ 0], \end{aligned} \quad (2)$$

where T stands for matrix transposition, with the following elements of the matrices $\mathbf{A}^{(j)}$ and $\mathbf{b}^{(j)}$, which depend on $P^{(j)}$:

$$\begin{aligned}
 a_{21}^{(j)} &= \frac{i_{EM1}^{(j)2}}{m} \frac{F_{emP1}}{F_{emP2}^2} \exp(-p^{(j)}/F_{emP2}) + \frac{i_{EM2}^{(j)2}}{m} \frac{F_{emP1}}{F_{emP2}^2} \exp(-(x_d - p^{(j)})/F_{emP2}), \\
 a_{23}^{(j)} &= -\frac{2i_{EM1}^{(j)}}{m} \frac{F_{emP1}}{F_{emP2}} \exp(-p^{(j)}/F_{emP2}), \quad a_{24}^{(j)} = \frac{2i_{EM2}^{(j)}}{m} \frac{F_{emP1}}{F_{emP2}} \exp(-(x_d - p^{(j)})/F_{emP2}), \\
 a_{31}^{(j)} &= -(k_i u_{EM1}^{(j)} + c_i - i_{EM1}^{(j)}) \frac{p^{(j)}}{f_{iP1}} \exp(p^{(j)}/f_{iP2}), \quad a_{33}^{(j)} = -\frac{f_{iP2}}{f_{iP1}} \cdot e^{\frac{x_{10}}{f_{iP2}}}, \\
 a_{41}^{(j)} &= -(k_i u_{EM2}^{(j)} + c_i - i_{EM2}^{(j)}) \frac{p^{(j)}}{f_{iP1}} \exp((x_d - p^{(j)})/f_{iP2}), \quad a_{44}^{(j)} = -\frac{f_{iP2}}{f_{iP1}} \exp((x_d - p^{(j)})/f_{iP2}), \\
 b_{31}^{(j)} &= k_i \frac{f_{iP2}}{f_{iP1}} \exp(p^{(j)}/f_{iP2}), \quad b_{42}^{(j)} = k_i \frac{f_{iP2}}{f_{iP1}} \exp((x_d - p^{(j)})/f_{iP2}).
 \end{aligned} \tag{3}$$

The operating points were chosen as follows such that to belong to the steady-state zone of the sphere position sensor input-output map [33], to cover the usual operating regimes and to avoid the extremities of the input-output map due to instability that may occur:

$$\begin{aligned}
 &P^{(1)}(0.0063, 0, 1.22, 0.39), P^{(2)}(0.007, 0, 1.145, 0.39), P^{(3)}(0.0077, 0, 1.07, 0.39), \\
 &P^{(4)}(0.0084, 0, 1, 0.39), P^{(5)}(0.009, 0, 0.9345, 0.39), P^{(6)}(0.0098, 0, 0.89, 0.39), \\
 &P^{(7)}(0.0105, 0, 0.83, 0.39).
 \end{aligned} \tag{4}$$

The transfer function (t.f) of the state-space linearized MM (2) used in the control system design is:

$$H_{PC}^{(j)}(s) = \mathbf{c}^T(s\mathbf{I} - \mathbf{A}^{(j)})^{-1} \mathbf{b}_{u_{EM1}}^{(j)} = \frac{k^{(j)}}{\prod_{k=1}^3 (s - p_k^{(j)})} = \frac{k_p^{(j)}}{\prod_{k=1}^3 (1 + T_k^{(j)} s)}, \tag{5}$$

where $k_p^{(j)} = k^{(j)} / \prod_{k=1}^3 p_k^{(j)}$, \mathbf{I} is the third-order identity matrix and the time constants are $T_k^{(j)} = -1/p_k^{(j)}$, $k = 1...3$, $j = 1...7$. The expressions of the t.f.s $H_{PC}^{(j)}(s)$ are given in [22], [23].

3 Control solutions design

In order to support the development of next control solutions, the SFCS is designed for the reduced third-order linear mathematical model ($u_2 = u_{EM2} = 0$) of unstable magnetic levitation system [22]. The first three state variables are kept and they lead to the state vector $\mathbf{x} = [p \quad v \quad i_{EM1}]^T$.

The pole placement method is applied to compute the state feedback gain matrix, $\mathbf{k}_c^{T(j)} = [k_{c1}^{(j)} \quad k_{c2}^{(j)} \quad k_{c3}^{(j)}]^T$, $j = 1...7$. Therefore, for each linearized MM, with the t.f. $H_{PC}^{(j)}(s)$ (5), the closed-loop system poles $p_k^{*(j)}$, $k = 1...3$, $j = 1...7$, [22], [23], have been imposed in order to guarantee the stability of the linearized system. With the obtained state feedback gain matrix, $\mathbf{k}_{cbest}^T = \mathbf{k}_c^T_{-5} = [66.63 \quad 1.62 \quad -0.15]$, two types of closed-loop t.f.s of the new state feedback control structure (nSFCS), $H_{SFCS_5}^{(j)}(s)$ result as:

$$\begin{aligned}
 H_{SFCS_5}^{(j)}(s) &= H_x^{(j)}(s) = \mathbf{c}^T(s\mathbf{I} - \mathbf{A}_{x_5}^{(j)})^{-1} \mathbf{b}_{u_{EM1}}^{(j)} = \mathbf{c}^T(s\mathbf{I} - (\mathbf{A}^{(j)} - \mathbf{b}_{u_{EM1}}^{(j)} \mathbf{k}_c^T_{-5} k_{AS}))^{-1} \mathbf{b}_{u_{EM1}}^{(j)} \\
 &= \begin{cases} \frac{k_{SFCS_5}^{(j)}}{(1+T_{1x_5}^{(j)}s)(1+2c_5^{(j)}T_{2x_5}^{(j)}s+T_{2x_5}^{(j)2}s^2)}, & j = 1...3, j \in \{6, 7\}, \\ \frac{k_{SFCS_5}^{(j)}}{(1+T_{1x_5}^{(j)}s)(1+T_{2x_5}^{(j)}s)(1+T_{3x_5}^{(j)}s)}, & j \in \{4, 5\}. \end{cases}
 \end{aligned} \tag{6}$$

Table 1: nSFCS poles and parameters

O.p.	nSFCS poles				nSFCS parameters			
	$p_{1_5}^{*(j)}$	$p_{2_5}^{*(j)}$	$p_{3_5}^{*(j)}$	$k_{SFCS_5}^{(j)}$	$T_{1x_5}^{(j)}$	$T_{2x_5}^{(j)}$	$\zeta_5^{(j)}$	$T_{etax_5}^{(j)}$
(1)	-0.79+1.02i	-0.79-1.02i	-0.10	0.084	0.0988	-	0.6	0.0077
(2)	-0.92+0.88i	-0.92-0.88i	-0.13	0.065	0.0778	-	0.7	0.0078
(3)	-1.07+0.62i	-1.07-0.62i	-0.16	0.054	0.0618	-	0.9	0.0081
(4)	-1.65	-0.82	-0.21	0.046	0.0485	0.0123	-	-
(5)	-2.32	-0.41	-0.32	0.041	0.0314	0.0244	-	-
(6)	-3.07	-0.28+0.16i	-0.28-0.16i	0.038	0.0033	-	0.9	0.0308
(7)	-3.78	-0.23+0.20i	-0.23-0.20i	0.034	0.0026	-	0.7	0.0332

The new SFCS poles obtained with the state feedback gain matrix $\mathbf{k}_{c_5}^T$ and the nSFCS parameters used next in the PID-GS-C structure are given in Table 1.

Due to the fact that the natural SFCS does not contain an I component, so it cannot ensure the zero steady-state control error, the SFCS, as controlled plant, is included in a cascade control structure (CCS) with PID controller in the outer loop. Depending on the operating points, seven control solutions with PID controllers have been designed using pole-zero cancellation. The t.f.s of the designed PID controllers extended with a first order lag filter can be expressed as:

$$H_{PID_5}^{(j)}(s) = \begin{cases} \frac{k_{c_5}^{(j)}(1+2\zeta_{c_5}^{(j)}T_{c_5}^{(j)}s+T_{c_5}^{(j)2}s^2)}{s(1+T_{fd_5}^{(j)}s)}, j \in \{1, 2, 3, 6, 7\}, \\ \frac{k_{c_5}^{(j)}(1+T_{c1_5}^{(j)}s)(1+T_{c2_5}^{(j)}s)}{s(1+T_{fd_5}^{(j)}s)}, j \in \{4, 5\}, \end{cases} \quad (7)$$

with the tuning parameters $k_{c_5}^{(j)}$, $T_{c_5}^{(j)}$, $T_{c1_5}^{(j)}$, $T_{c2_5}^{(j)}$ and $T_{fd_5}^{(j)}$:

$$k_{c_5}^{(j)} = \begin{cases} 1/(2 \cdot k_{SFCS_5}^{(j)} \cdot T_{etax_5}^{(j)}), j = 1...3 \\ 0.05/(2 \cdot k_{SFCS_5}^{(j)} \cdot T_{\Sigma x_5}^{(j)}), j \in \{4, 5\} \\ 0.01/(2 \cdot k_{SFCS_5}^{(j)} \cdot T_{1x_5}^{(j)}), j \in \{6, 7\}, \end{cases} \quad (8)$$

$$\begin{cases} T_{c_5}^{(j)} = T_{etax_5}^{(j)}, j \in \{1, 2, 3, 6, 7\}, \\ T_{c1_5}^{(j)} = T_{1x_5}^{(j)}, j \in \{4, 5\}, \\ T_{c2_5}^{(j)} = T_{2x_5}^{(j)}, j \in \{4, 5\}, \\ T_{fd_5}^{(j)} = 0.1 \cdot T_{1x_5}^{(j)}, j = 1...7, \end{cases} \quad \zeta_{c_5}^{(j)} = \zeta_5^{(j)}.$$

Due to the oscillatory regime, the t.f.s coefficients $k_{c_5}^{(j)}$ must be adjusted. The numerical values of PID-C parameters and the system performance indices – overshoot and settling time – are synthesized in Table 2.

Due to the process nonlinearities, which also depend on the operating point, the switching from one PID controller to another one is a useful solution to ensure improved performance according to [48]. Therefore, a PID-GS-C structure illustrated in Figure 1 is designed. The PID-GS-C structure is developed on the basis of the PID controller (with the notation PID-C in Figure 1), with the t.f.:

$$\Delta u_{1x}(t) = k_p(t)e(t) + k_i(t) \int e(t)dt + k_d(t)\dot{e}(t), \quad (9)$$

where k_p is the proportional gain, k_d is the derivative gain and k_i is the integral gain.

Table 2: PID controller parameters

O.p.	PID controller tuning parameters		PID control system performance indices	
	$k_c^{(j)}$	$T_c^{(j)}$	$t_r^{(j)}$	$\sigma_1^{(j)}$
(1)	60.65	0.009	4,25	0,24
(2)	98.23	0.008	4,25	0,24
(3)	150.83	0.006	4,5	0,24
(4)	89.88	0.005	5,0	0,23
(5)	28.67	0.003	4,0	0,23
(6)	40.98	0.003	3,5	0,24
(7)	111.24	0.003	4,0	0,24

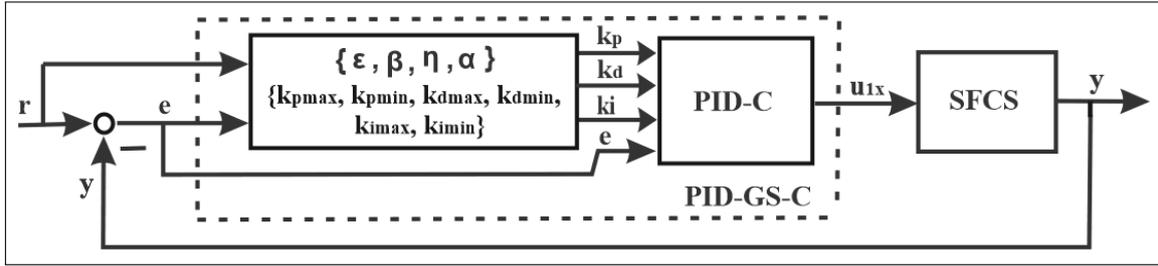


Figure 1: Block diagram of PID-GS-C system structure

As shown in Figure 1, the gain-scheduling (GS) block uses the reference input $r(t)$ and the control error $e(t)$ as input variables, and the tuning parameters of the PID-C, namely k_p , k_d and k_i , as output variables. The PID tuning parameters are obtained as follows:

$$\begin{aligned}
 k_p(t) &= k_{p \max} - (k_{p \max} - k_{p \min}) \exp[-(\alpha(t)|e(t)|)], \\
 k_d(t) &= k_{d \max} - (k_{d \max} - k_{d \min}) \exp[-(\alpha(t)|e(t)|)], \\
 k_i(t) &= (1 - \alpha(t))k_{i \max}.
 \end{aligned} \tag{10}$$

The parameters $k_{p \max}$, $k_{p \min}$, $k_{d \max}$, $k_{d \min}$, $k_{i \max}$ and $k_{i \min}$ are determined from the PID-GS-C tuning parameters:

$$\begin{aligned}
 k_{p \max} &= \begin{cases} \max[k_c^{(j)} \cdot (T_{c1}^{(j)} + T_{c2}^{(j)})], j \in \{4, 5\}, \\ \max[2 \cdot k_c^{(j)} \cdot \zeta_c^{(j)} \cdot T_c^{(j)}], j \in \{1, 2, 3, 6, 7\}, \end{cases} \\
 k_{p \min} &= \begin{cases} \min[k_c^{(j)} \cdot (T_{c1}^{(j)} + T_{c2}^{(j)})], j \in \{4, 5\}, \\ \min[2 \cdot k_c^{(j)} \cdot \zeta_c^{(j)} \cdot T_c^{(j)}], j \in \{1, 2, 3, 6, 7\}, \end{cases} \\
 k_{d \max} &= \begin{cases} \max(k_c^{(j)} \cdot T_{c1}^{(j)} \cdot T_{c2}^{(j)}), j \in \{4, 5\}, \\ \max(k_c^{(j)} \cdot T_c^{(j)^2}), j \in \{1, 2, 3, 6, 7\}, \end{cases} \\
 k_{d \min} &= \begin{cases} \min(k_c^{(j)} \cdot T_{c1}^{(j)} \cdot T_{c2}^{(j)}), j \in \{4, 5\}, \\ \min(k_c^{(j)} \cdot T_c^{(j)^2}), j \in \{1, 2, 3, 6, 7\}, \end{cases} \\
 k_{i \max} &= \max(k_c^{(j)}), k_{i \min} = \min(k_c^{(j)}), j = 1 \dots 7.
 \end{aligned} \tag{11}$$

The parameter $0 \leq \alpha(t) \leq 1$ is included in order to have a smooth and continuous variation of the switching from one PID controller to another one. The following equation is used to get the value of this parameter:

$$\alpha(t) = \tanh(\eta\beta(t)) = [\exp(2\eta\beta(t)) - 1]/[\exp(2\eta\beta(t)) + 1], \tag{12}$$

where the parameter η determines the rate at which $\alpha(t)$ changes between 0 and 1 and chosen in order to ensure a certain dynamics of the variation of $\alpha(t)$. The parameter $\beta(t)$ is set in terms of [35]:

$$\beta(t) = \begin{cases} 1, & |e(t)| > \xi, \\ 0, & |e(t)| < \xi, \end{cases}, \quad \xi = 0.9 \cdot r(t). \quad (13)$$

As shown in [35], to design the PID-GS-C structure, the following conditions can be taken into account: when the system is in steady-state error (i.e., $|e(t)|$ is large), $k_{p\max}$ and $k_{i\min}$ are activated in order to produce a large control signal and to overcome the undesirable oscillation and overshoot; during the steady-state regime, $k_{i\max}$ and $k_{p\min}$ are activated to obtain a small value of $|e(t)|$ and to overcome the undesirable problem of overshoot.

4 Experimental results

All control structures, namely, SFCS, PID-C and PID-GS-C, were tested on the nonlinear laboratory system and validated by real-time experiments. In all cases, the reference input was set to 0.007 m from the top electromagnet and the control structures responses were tested on the time frame of 20 s. The responses of the sphere position, the current and the control signal in the top electromagnet were plotted.

The values of the parameters of the PID-GS-C block in Figure 1 are $k_{p\max} = 5516$, $k_{p\min} = 0.6$, $k_{d\max} = 0.1225$, $k_{d\min} = 0.0036$, $k_{i\max} = 151$, $k_{i\min} = 0$ and $\eta \in \{0.001, 0.1\}$. They have been obtained using several experiments such that to get the best values in the context of a compromise to tradeoff to overshoot. But other empirical performance indices can be considered in this tuning.

The following experimental scenarios were considered and performed:

(a) The state feedback control system structure with the best state feedback gain matrix was tested on the laboratory equipment and the results are presented in Figure 2.

(b) The PID controller designed at seven operating points was tested on laboratory equipment. The experimental results of the control system with PID controller designed only for three o.p.s defined in Tables 1 and 2, i.e., (1), (5) and (7) are presented as responses of several variables measured in the laboratory setup in Figures 3 to 5 as follows: the control signal versus time in Figure 3, the current through EM1 versus time in Figure 4 and the controlled output versus time in Figure 5. These results are better in comparison with the results presented in Figure 2 because of the PID-C, which ensures the zero steady-state control error and the reference input is tracked. The pair of complex conjugated poles from the cases that correspond to the o.p.s (1) to (3) (similarly to the o.p.s (6) and (7)) and the nonlinearities of the process lead to oscillations at the beginning of transient responses and during the real-time experiments.

(c) The experimental results of the control system with PID-GS controller are presented as responses of several variables measured in the laboratory setup in Figures 6 to 8 as follows: the control signal versus time in Figure 6, the current through EM1 versus time in Figure 7 and the controlled output versus time in Figure 8. These results are better in comparison with the results presented in Figures 3 to 5 because of the PID-GS controller that ensures improved dynamic behavior characterized by smaller overshoot and settling time.

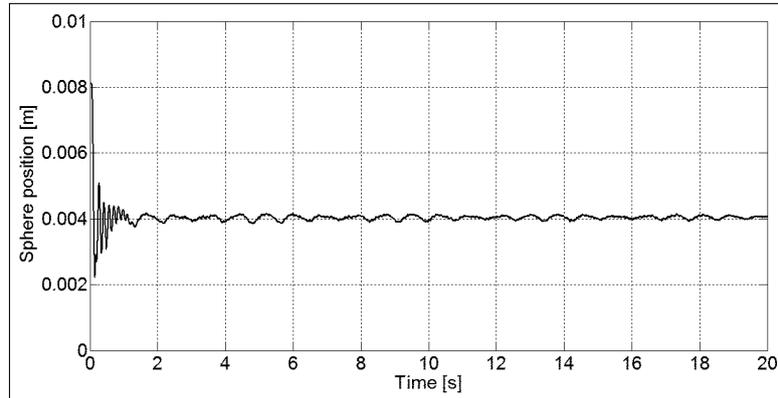


Figure 2: Real-time experimental results of state feedback control system structure with the best state feedback gain matrix

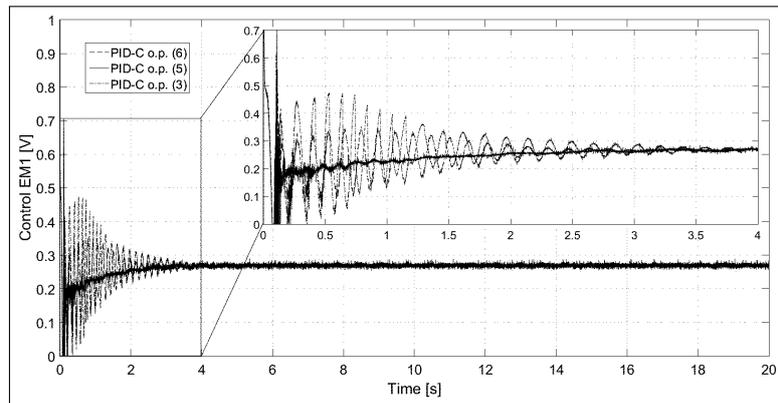


Figure 3: Control signal $u_1 = u_{EM1}^{(j)}$, $j \in \{3, 5, 6\}$, versus time for control systems with PID controller for o.p.s. (3), (5) and (6)

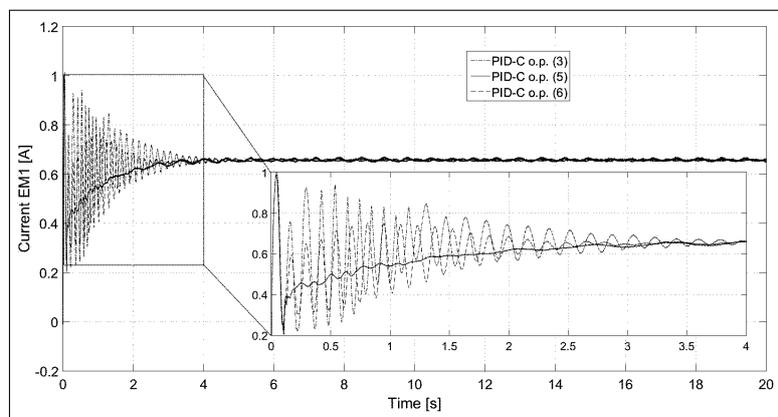


Figure 4: Current i_{EM1} versus time for control systems with PID controller designed for o.p.s. (3), (5) and (6)

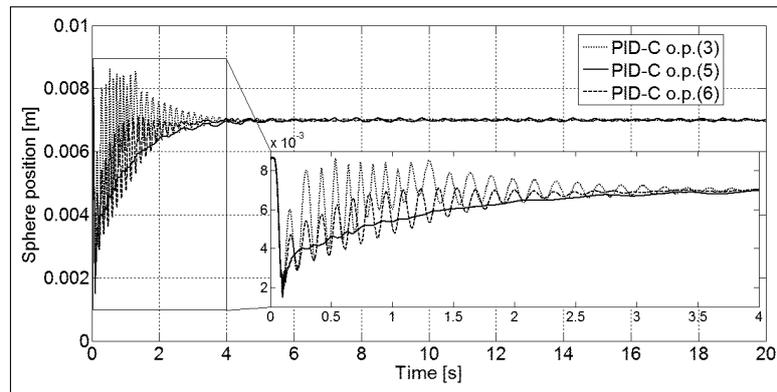


Figure 5: Sphere position p versus time for control systems with PID controller designed for o.p.s. (3), (5) and (6)

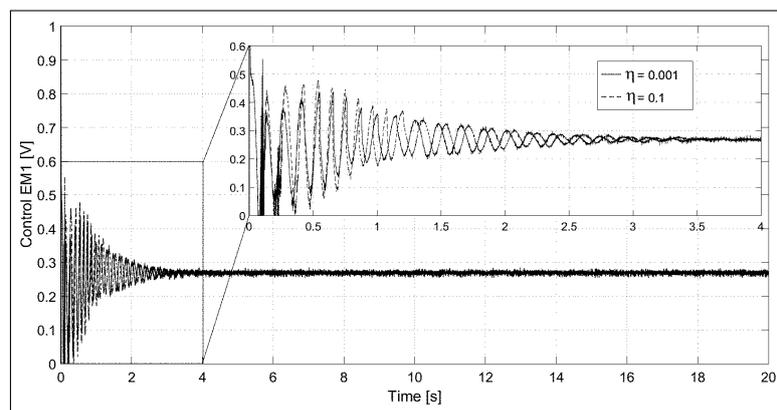


Figure 6: Control signal $u_1 = u_{EM1}^{(j)}$ versus time for control systems with PID-GS controller

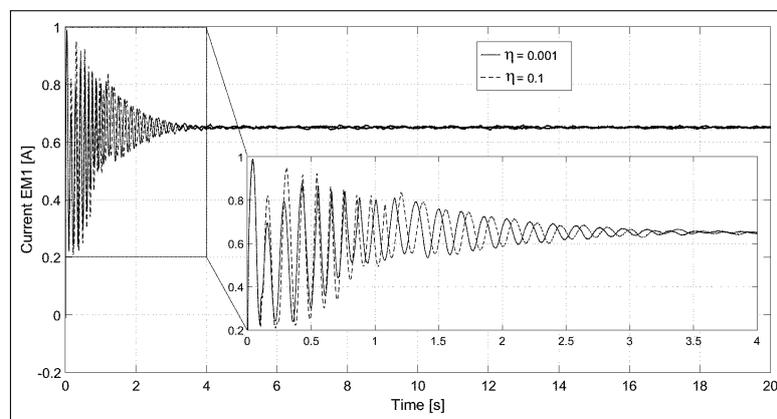


Figure 7: Current i_{EM1} versus time for control systems with PID-GS controller

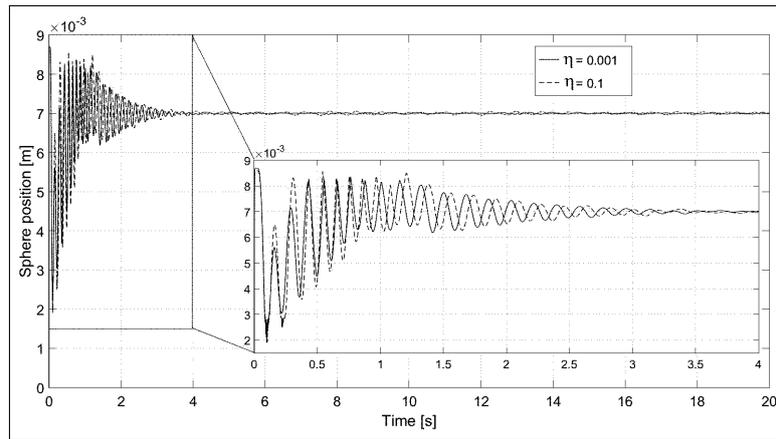


Figure 8: Sphere position p versus time for control systems with PID-GS controller

5 Conclusions

The paper has presented three control structures developed to control the position of the sphere in an MLS2EM laboratory setup. In order to use relatively simple control structures, the presented nonlinear model of the MLS2EM was linearized around seven operating points. To stabilize the process, a state feedback control structures was designed and the best state feedback gain matrix was found.

Since the SFCS does not ensure the zero steady-state control error, seven PID controllers were designed. To ensure the switching between different PID controllers, a PID-GS controller was developed and implemented. All control system structures were tested on the nonlinear model, accepting the main values of the parameters given in [24]. This paper has considered a low-cost implementation of the controllers but this must be viewed in connection with the complexity of the control algorithms, and several approaches can be used [10], [12], [37], [13], [15], [31], [46].

The real-time experimental results prove that the PID-GS-C structure discussed in this paper guarantees the improvement of control system performance regarding to step modifications of reference input. They ensure zero steady-state control error, small overshoot and settling time. As shown in the previous section, the choice of the parameters of the PID-GS-C block in Figure 1 has been carried out by conducting several experiments such that to aim the best values from the point of view of the compromise to the achievement of best overshoot and settling time. This is a limitation of the control system structure presented in this paper.

The systematic choice of the parameters of PID-GS-C will represent a direction of future research by ensuring the optimal tuning using classical [4], [22], [39], [2], [25], [14] and modern optimization algorithms [40], [41], [45], [26], [33], [30], [34], [21], [3] or their combinations. Future research will also be focused on the design of control systems with PI(D) fuzzy gain-scheduling controllers, Takagi-Sugeno fuzzy controllers and hybrid structures including sliding mode control and gain-scheduling control for improved performance indices.

Acknowledgment

This work was supported by grants from the Partnerships in priority areas – PN II program of the Romanian Ministry of National Education and Scientific Research (MENCS) – the Executive Agency for Higher Education, Research, Development and Innovation Funding

(UEFISCDI), project numbers PN-II-PT-PCCA-2013-4-0544, PN-II-PT-PCCA-2013-4-0070 and PN-II-RU-TE-2014-4-0207.

Bibliography

- [1] An S., Ma Y., Cao Z. (2009); Applying simple adaptive control to magnetic levitation system, *Proceedings of 2nd International Conference on Intelligent Computation Technology and Automation*, Changsha, Hunan, China, 1, 746-749, 2009.
- [2] Angelov P., Yager R. (2012); A new type of simplified fuzzy rule-based systems, *International Journal of General Systems*, 41(2), 163-185, 2012.
- [3] Azar D., Fayad K., Daoud C. (2016); A combined ant colony optimization and simulated annealing algorithm to assess stability and fault-proneness of classes based on internal software quality attributes, *International Journal of Artificial Intelligence*, 14(2), 137-156, 2016.
- [4] Baranyi P., Tikk D., Yam Y., Patton R.J. (2003); From differential equations to PDC controller design via numerical transformation, *Computers in Industry*, 51(3), 281-297, 2003.
- [5] Bianchi F.D., Sánchez Peña R.S. (2011); Interpolation for gain-scheduled control with guarantees, *Automatica*, 47(1), 239-243, 2011.
- [6] Bianchi F.D.; Sánchez-Pena R.S., Guadayol M. (2012); Gain scheduled control based on high fidelity local wind turbine models, *Renewable Energy*, 37(1), 233-240, 2012.
- [7] Bedoud K., Ali-rachedi M., Bahid T., Lakel R. (2015); Adaptive fuzzy gain scheduling of PI controller for control of the wind energy conversion systems, *Energy Procedia*, 74, 211-225, 2015.
- [8] Bojan-Dragos C.-A., Preitl S., Precup R.-E., Hergane S., Hughiet E.G., Szedlak-Stinean A.-I. (2016); State feedback and proportional-integral-derivative control of a magnetic levitation system, *Proceedings of IEEE 14th International Symposium on Intelligent Systems and Informatics*, Subotica, Serbia, 111-116, 2016.
- [9] Bojan-Dragos C.-A., Preitl S., Precup R.-E., Hergane S., Hughiet E.G., Szedlak-Stinean A.-I. (2016); Proportional-integral gain-scheduling control of a magnetic levitation system, *Proceedings of 20th International Conference on System Theory, Control and Computing*, Sinaia, Romania, 1-6, 2016.
- [10] Chadli M., Akhenak A., Ragot J., Maquin D. (2009); State and unknown input estimation for discrete time multiple model, *Journal of the Franklin Institute*, 346(6), 593-610, 2009.
- [11] Chauhan S., Nigam M.J. (2014); Model predictive controller design and perturbation study for magnetic levitation system, *Proceedings of 2014 IEEE Recent Advances in Engineering and Computational Sciences*, Chandigarh, India, 1-6, 2014.
- [12] Colhon M., Danciulescu, D. (2010); Semantic schemas for natural language generation in multilingual systems, *Journal of Knowledge, Communications and Computing Technologies*, 2(1), 10-17, 2010.
- [13] Danciulescu D. (2015); Formal languages generation in systems of knowledge representation based on stratified graphs, *Informatica*, 26(3), 407-417, 2015.

-
- [14] Deliparaschos K., Michail K., Zolotas A., Tzafestas S. (2016); FPGA-based efficient hardware/software co-design for industrial systems with systematic sensor selection, *Journal of Electrical Engineering*, 67(3), 150-159, 2016.
 - [15] Derr K.W., Manic M. (2015); Wireless sensor networks - node localization for various industry problems, *IEEE Transactions on Industrial Informatics*, 11(3), 752-762, 2015.
 - [16] Dragos C.-A., Precup R.-E., David R.-C., Preitl S., Stinean A.-I., Petriu E.M. (2013); Simulated annealing-based optimization of fuzzy models for magnetic levitation systems, *Proceedings of 2013 Joint IFSA World Congress and NAFIPS Annual Meeting*, Edmonton, AB, Canada, 286-29, 2013.
 - [17] Dounis A.I., Kofinas P., Alafodimos C., Tseles D. (2013); Adaptive fuzzy gain scheduling PID controller for maximum power point tracking of photovoltaic system, *Renewable Energy*, 60, 202-214, 2013.
 - [18] Dzitac I. (2015); The fuzzification of classical structures: A general view, *International Journal of Computers Communications & Control*, 10(6), 772-788, 2015.
 - [19] Elsodany N.M., Rezeka S.F., Maharem N.A. (2011); Adaptive PID control of a stepper motor driving a flexible rotor, *Alexandria Engineering Journal*, 50(2), 127-136, 2011.
 - [20] Filip F.G. (2008); Decision support and control for large-scale complex systems, *Annual Reviews in Control*, 32(1), 61-70, 2008.
 - [21] Gaxiola F., Melin P., Valdez F., Castro J.R., Castillo O. (2016); Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO, *Applied Soft Computing*, 38, 860-871, 2016.
 - [22] Haber R.E., Alique J.R. (2007); Fuzzy logic-based torque control system for milling process optimization, *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 37(5), 941-950, 2007.
 - [23] Huang Y.-W., Tung P.-C. (2009); Design of a fuzzy gain scheduling controller having input saturation: a comparative study, *Journal of Marine Science and Technology*, 17(4), 249-256, 2009.
 - [24] Inteco. (2008); Magnetic Levitation System 2EM (MLS2EM), User's Manual (Laboratory Set), Inteco Ltd., Krakow, Poland, 2008.
 - [25] Johanyák Z.C. (2013); Fuzzy modeling of thermoplastic composites' melt volume rate, *Computing and Informatics*, 32(4), 845-857, 2013.
 - [26] Kazakov A.L., Lempert A.A. (2015); On mathematical models for optimization problem of logistics infrastructure, *International Journal of Artificial Intelligence*, 13(1), 200-210, 2010.
 - [27] King R., Stathaki A. (2000); Fuzzy gain scheduling control of nonlinear processes, *Research Report, Department of Electrical and Computer Engineering*, University of Patras, Patras, Greece, 2000.
 - [28] Lashin M., Elgammal A.T., Ramadan A., Abouelsoud A.A., Assal S.F.M., Abo-Ismael A. (2014); Fuzzy-based gain scheduling of exact feedforward linearization control and SMC for magnetic ball levitation system: A comparative study, *Proceedings of 2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, Cluj-Napoca, Romania, 1-6, 2014.

-
- [29] Michino R., Tanaka H., Mizumoto I. (2009); Application of high gain adaptive output feedback control to a magnetic levitation system, *Proceedings of ICROS-SICE International Joint Conference*, Fukuoka, Japan, 970-975, 2009.
- [30] Moharam A., El-Hosseini M.A., Ali H.A. (2015); Design of optimal PID controller using NSGA-II algorithm and level diagram, *Studies in Informatics and Control*, 24(3), 301-308, 2015.
- [31] Negru V., Grigoras G., Danciulescu D. (2015); Natural language agreement in the generation mechanism based on stratified graphs, *Proceedings of 7th Balkan Conference in Informatics*, Craiova, Romania, 36:1-36:8, 2015.
- [32] Obe O., Dumitrache I. (2012); Adaptive neuro-fuzzy controller with genetic training for mobile robot control, *International Journal of Computers Communications & Control*, 7(1), 135-146, 2012.
- [33] Osaba E., Onieva E., Dia F., Carballado R., Lopez P., Perallos A. (2015); A migration strategy for distributed evolutionary algorithms based on stopping non-promising subpopulations: A case study on routing problems, *International Journal of Artificial Intelligence*, 13(2), 46-56, 2015.
- [34] Osaba E., Yang X.-S., Diaz F., Lopez P., Carballado R. (2016); An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems, *Engineering Applications of Artificial Intelligence*, 48, 59-71, 2016.
- [35] Pallav S., Pandey K., Laxmi V. (2014); PID control of magnetic levitation system based on derivative filter, *Proceedings of 2014 Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives*, Kottayam, India, 1-5, 2014.
- [36] Precup R.-E., Angelov P., Costa B.S.J., Sayed-Mouchaweh M. (2015); An overview on fault diagnosis and nature-inspired optimal control of industrial process applications, *Computers in Industry*, 74, 75-94, 2015.
- [37] Precup R.-E., David R.-C., Petriu E.M., Preitl S., Radac M.-B. (2014); Novel adaptive charged system search algorithm for optimal tuning of fuzzy controllers, *Expert Systems with Applications*, 41(4), 1168-1175, 2014.
- [38] Precup R.-E., Preitl S. (1997); Popov-type stability analysis method for fuzzy control systems, *Proceedings of Fifth European Congress on Intelligent Technologies and Soft Computing*, Aachen, Germany, 2, 1306-1310, 1997.
- [39] Precup R.-E., Preitl S. (2007); PI-fuzzy controllers for integral plants to ensure robust stability, *Information Sciences*, 177(20), 4410-4429, 2007.
- [40] Precup R.-E., Sabau, M.-C., Petriu E.M. (2015); Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for anti-lock braking systems, *Applied Soft Computing*, 27, 575-589, 2015.
- [41] Precup R.-E., Radac M.-B., Tomescu M.L., Petriu E.M., Preitl S. (2013); Stable and convergent iterative feedback tuning of fuzzy controllers for discrete-time SISO systems, *Expert Systems with Applications*, 40(1), 188-199, 2013.
- [42] Precup R.-E., Tomescu M.L., Preitl S. (2009); Fuzzy logic control system stability analysis based on Lyapunov's direct method, *International Journal of Computers Communications & Control*, 4(4), 415-426, 2009.

-
- [43] Preitl S., Precup R.-E. (1996); On the algorithmic design of a class of control systems based on providing the symmetry of open-loop Bode plots, *Scientific Bulletin of UPT, Transactions on Automatic Control and Computer Science*, 41(55)(2), 47-55, 1996.
 - [44] Puig V., Bolea Y., Blesa J. (2012); Robust gain-scheduled Smith PID controllers for second order LPV systems with time varying delay, *IFAC Proceedings Volumes*, 45(3), 199-204, 2012.
 - [45] Rebai A., Guesmi K., Hemicci B. (2015); Design of an optimized fractional order fuzzy PID controller for a piezoelectric actuator, *Control Engineering and Applied Informatics*, 17(3), 41-49, 2015.
 - [46] Qin Q., Cheng S., Zhang, Q., Qin Q., Cheng S., Zhang Q., Li L., Shi Y. (2016); Particle swarm optimization with interswarm interactive learning strategy, *IEEE Transactions on Cybernetics*, 46(10), 2238-2251, 2016.
 - [47] Sakalli A., Kumbasar T., Yesil E., Hagraş H. (2014); Analysis of the performances of type-1, self-tuning type-1 and interval type-2 fuzzy PID controllers on the magnetic levitation system, *Proceedings of 2014 IEEE International Conference on Fuzzy Systems*, Beijing, China, 1859-1866, 2014.
 - [48] Sedaghati A. (2006); A PI controller based on gain-scheduling for synchronous generator, *Turkish Journal of Electrical Engineering and Computer Sciences*, 14(2), 241-251, 2006.
 - [49] Shameli E., Khamesee M.B., Huissoon J.P. (2007); Nonlinear controller design for a magnetic levitation device, *Microsystem Technologies*, 13(8), 831-835, 2007.
 - [50] Škrjanc I., Blažič S., Agamennoni O.E. (2005); Interval fuzzy model identification using l_∞ -norm, *IEEE Transactions on Fuzzy Systems*, 13(5), 561-568, 2005.
 - [51] Vaščák J. (2010); Approaches in adaptation of fuzzy cognitive maps for navigation purposes, *Proceedings of 8th International Symposium on Applied Machine Intelligence and Informatics*, Heržany, Slovakia, 31-36, 2010.
 - [52] Wang B.; Liu G.-P.; Rees D. (2009); Networked predictive control of magnetic levitation system, *Proceedings of 2009 IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, 4100-4105, 2009.
 - [53] Xie Y.-M., Shi H., Alleyne A., Yang B.-G. (2016); Feedback shape control for deployable mesh reflectors using gain scheduling method, *Acta Astronautica*, 121, 241-255, 2016.
 - [54] Yang Y.-N., Yan Y. (2016); Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control, *Aerospace Science and Technology*, 54, 208-217, 2016.
 - [55] Zhao Z.Y., Tomizuka M. (1993); Fuzzy gain scheduling of PID controllers, *IEEE Transactions on Systems, Man, and Cybernetics*, 23(5), 1392-1398, 1993.
 - [56] Zietkiewicz J. (2011); Constrained predictive control of a levitation system, *Proceedings of 16th International Conference on Methods and Models in Automation and Robotics*, Miedzydroje, Poland, 278-283, 2011.

Benchmarking of Recommendation Trust Computation for Trust/Trustworthiness Estimation in HDNs

H. El-Sayed, S. Sankar, H. Yu, G. Thandavarayan

Hesham El-Sayed*, Heng Yu, Gokulnath Thandavarayan

College of Information Technology,

UAE University,

Al Ain-15551, United Arab Emirates

*Corresponding author: helsayed@uaeu.ac.ae

heng.yu@uaeu.ac.ae, gokul@uaeu.ac.ae

Sharmi Sankar

Department of Information Technology,

Ibri College of Applied Sciences (MoHE),

Sultanate of Oman

sharmisankar41@gmail.com.

Abstract: In the recent years, Heterogeneous Distributed Networks (HDNs) is a predominant technology implemented to enable various application in different fields like transportation, medicine, war zone, etc. Due to its arbitrary self-organizing nature and temporary topologies in the spatial-temporal region, distributed systems are vulnerable with a few security issues and demands high security countermeasures. Unlike other static networks, the unique characteristics of HDNs demands cutting edge security policies. Numerous cryptographic techniques have been proposed by different researchers to address the security issues in HDNs. These techniques utilize too many resources, resulting in higher network overheads. This being classified under light weight security scheme, the Trust Management System (TMS) tends to be one of the most promising technology, featured with more efficiency in terms of availability, scalability and simplicity. It advocates both the node level validation and data level verification enhancing trust between the attributes. Further, it thwarts a wide range of security attacks by incorporating various statistical techniques and integrated security services. In this paper, we present a literature survey on different TMS that highlights reliable techniques adapted across the entire HDNs. We then comprehensively study the existing distributed trust computations and benchmark them in accordance to their effectiveness. Further, performance analysis is applied on the existing computation techniques and the benchmarked outcome delivered by Recommendation Trust Computations (RTC) is discussed. A Receiver Operating Characteristics (ROC) curve illustrates better accuracy for Recommendation Trust Computations (RTC), in comparison with Direct Trust Computations (DTC) and Hybrid Trust Computations (HTC). Finally, we propose the future directions for research and highlight reliable techniques to build an efficient TMS in HDNs.

Keywords: direct trust computations, hybrid trust computations (HTC), heterogeneous distributed networks (HDNs), receiver operating characteristics, recommendation trust computations (RTC).

1 Introduction

Heterogeneous Distributed Networks have become increasingly popular in the recent years, expanding its contribution across different computing fields with promising results. In wireless networks, the nodes are equipped with On Board Units (OBU) that creates a dynamic communication between different agents without the need for any network infra-structure. Based on

the user demands and network behavior, the HDNs can be classified into Mobile Ad-Hoc Network (MANET), Vehicular Ad-Hoc Network (VANET) and Wireless Sensor Network (WSN), featured with unique characteristics to serve a particular environment. Many researchers have rendered incredible contribution in all these domains and have developed standards and protocols to serve the domain [3, 13]. Certain vital operations like sensing [16], routing [14, 17, 23] and event monitoring [22] are some of the major research topics carried out by the researchers, rendering prodigious contributions and problem-solving techniques in every domain.

Nowadays, the networks are loaded with built-in reliable standards and protocols, despite this the security and privacy concerns are still some of the major issues [15], degrading the efficiency and reliability of the application. Owing to the openness of the wireless network and uncontrollability of the nodes without a Central Authority (CA), the availability of compromising nodes and unauthorized access becomes inevitable in the network. Many researchers have worked on this issue and proposed various schemes that can be broadly classified as hard and soft security schemes [29, 30]. In the hard security scheme, cryptographic algorithms are implemented to ensure confidentiality on information exchange and authentication of participation nodes. It demands more resource utilization to perform these actions and consumes more power and processing capabilities that are scarcely available in these networks. Moreover, the reputation of the participating nodes and the trustworthiness of the received message cannot be verified using these schemes/techniques. Therefore, establishing a behavior analysis on these nodes and the verification and validation of the message exchange is implemented using the soft security scheme called Trust Management Systems (TMS). Using trust systems, the behavior analysis is performed for the participating nodes and the selfish acts are reported by computing the trust metrics.

In general, attributes like trustworthiness, reputation, belief and confidence impose uncertain behavior in HDNs. Imprecise information floating around HDNs is unavoidable due to their limitations in wireless communications [8]. HDNs demands an efficient technique to calibrate the opinions and recommendations derived from direct and indirect communications. For this purpose, trust computation (TC) is an important element in TMS, which will enable to compute a trust metric between the trustor and trustee with the available information. Computing and verifying the trust metric in unsupervised environment is challenging and demands an efficient technique to handle it. Also during trust computations, the probability level of acceptance rate in trust metric will change periodically due to the spatial-temporal factors. The probability levels on trustworthiness can be estimated dynamically by monitoring the risk factors involved. Risk analysis (RA) [11] is an important framework to measure trust metric in HDN and will be used to adjust the acceptance rate of the trust metric based on the current requirement as shown in Figure 1. Many technologies have been evolved for RA which primarily focuses on data driven and node driven risk analysis. Decision making in HDNs is adapted to refine the obtained trust metric using aggregated information. Due to the presence of uncertainty and imprecisions in the trust computation logic, a reliable aggregation technique will help the requestor to estimate a reliable trust metric of the enquiring node.

Implementing TMS for a fixed network is simple and direct, easily computing the trustworthiness of a participating node. But, establishing a TMS in HDNs is highly challenging due to the uncertain behavior in the environment. The following key points are some of the major issues in establishing TMS in distributed networks.

- The spatial and temporal dimensions of the participating nodes are ever changing in the dynamic networks making the node observation more challenging. Decisions made from interactions may further provide futile results as the location and time cannot be referenced in the environment which may increase the selfish behavior [10].

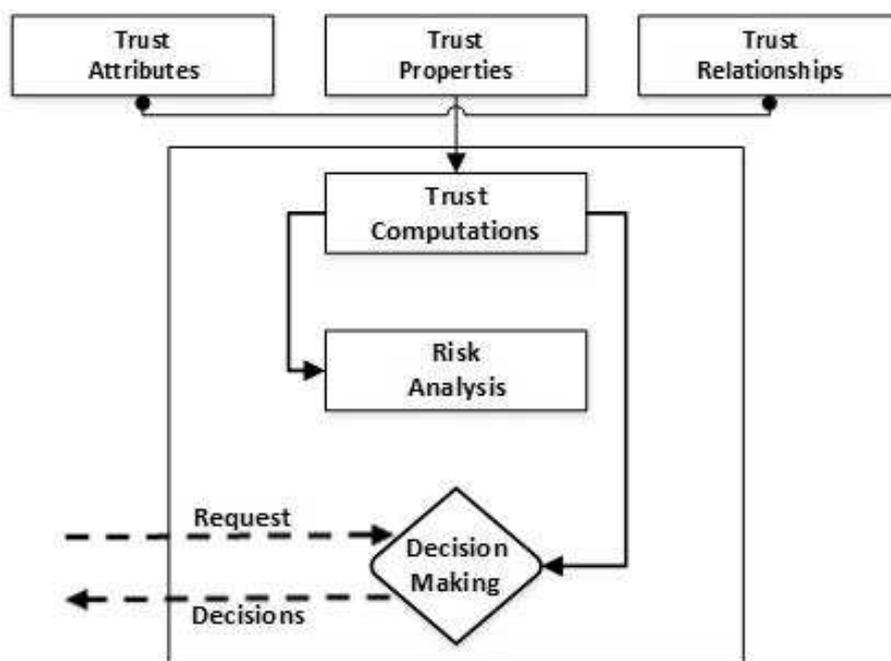


Figure 1: A general view on trust supported models

- As every network follows a particular mobility model [25], handshaking between nodes increases overhead due to the limitations in wireless network.
- Also the network does not follow a master-slave relationship, hence the decision making on a recommendation message from a neighbor node necessitates further verification to strengthen the result. It also limits the trust chain to be intact which cannot be forwarded further.

In the recent years, TMS has contributed numerous models for the HDN and enormous contributions from researchers has led to the development of an efficient model. However, a detailed survey and benchmarking of these models is required for the entire HDN. Some researchers have published survey papers on TMS which highlights the trust computing techniques only for a particular network domain [9, 33]. Also, benchmarking the existing techniques and comparing the techniques for an effective study, is still a major shortfall in these survey papers.

Therefore, a cohesive survey is still in high demand for the entire distributed networks to understand the accomplishment of the existing TMS models and requires benchmarking techniques to explore the best models for the entire domain of HDN. In this paper, we have made a thorough survey on TMS on a broader spectrum, highlighting the best practicing techniques in every HDN domain. Further, the existing techniques are tested with statistical models and the effectiveness is measured. The outcome of the performance measure presented in the paper highlights the best techniques for TMS which addresses the security and privacy issues effectively. Also, the paper conclusively recommends future research trends for a holistic TMS in HDN.

2 Trust dimensions

Trust is a relationship between two or more agents demonstrating the belief with one another, with more reliability and trustworthiness. In HDN, the agent can be a vehicle node, Road-Side Unit (RSU), mobile node or sensor node. Trust relationships are formed between agents to

compute the trust relationship between trusting agents and trusted agents. In this section, we introduce the trust dimensions which help to understand the basic entities of trust.

2.1 Trust environments

A node can analyze the trust relationship and compute the trust metrics based on the network environment in which the participant nodes exist. The trust computation in the environment can be broadly classified as physical and virtual trust. In physical trust, the trust is established between nodes based on direct interactions. In this environment, the trust evaluation is restricted to single hop interaction; whereas, in the virtual trust a trust metric can be computed from nodes which are connected by intermediate nodes. In this environment, the nodes are virtually connected and share recommendations to derive a trust metric.

2.2 Trust definition in literature

The importance of trust is widely accepted and acknowledged across different arenas in a multifaceted vision and connotations. In the field of sociology, trust is referred to as a basic fact of social life that involves both emotional and cognitive dimensions, sprouted from the past as well as from the present activity. In Philosophy, trust is determined as plausible that concedes to form relationships among people. It attributes to be an inevitable trait of our societal life and determines our relationship with the environment, despite the fact that it always carries the risk of being unwarranted. Likewise, in the Economics domain, trust can be conceptualized by bounded rational and subjective probabilities. It make decisions on the business activities to reduce the cost price and thereby discerned as an economic lubricant. In the field of Computer Science and technology, trust is employed as a medium to provide a high-end security, privacy, reliability and integrity in different computing group that ranges from network application to internal system computations. In computing, trust is the firm vision and belief to authorize an entity to act dependably secure and reliable within a specific context. It is to be kept in mind that both the trustor and trustee should be validated equally to enhance the contextual factor of expectancy.

2.3 Trust attributes

The three main properties of trust on a network are asymmetry, transitivity and composability. Asymmetry trust is assured as two nodes trust each other on the same level. Transitivity trust is an inferred trust. Composability trust, receives information from all available sources and compose an opinion value.

2.4 Trust metrics

Trust evaluation metrics are classified into three categories: Trust on scale, trust by facets, and trust upon logic. Trust on scale scheme measures the level of trust using discrete or continuous series. Trust through facets measure rely on three attributes; the belief, the disbelief and uncertainty. The trust is represented as a triplet phase. Trust upon logic uses a probability approach to determine the trust. The ratio upon the number of packets forwarded and received is one trust measure.

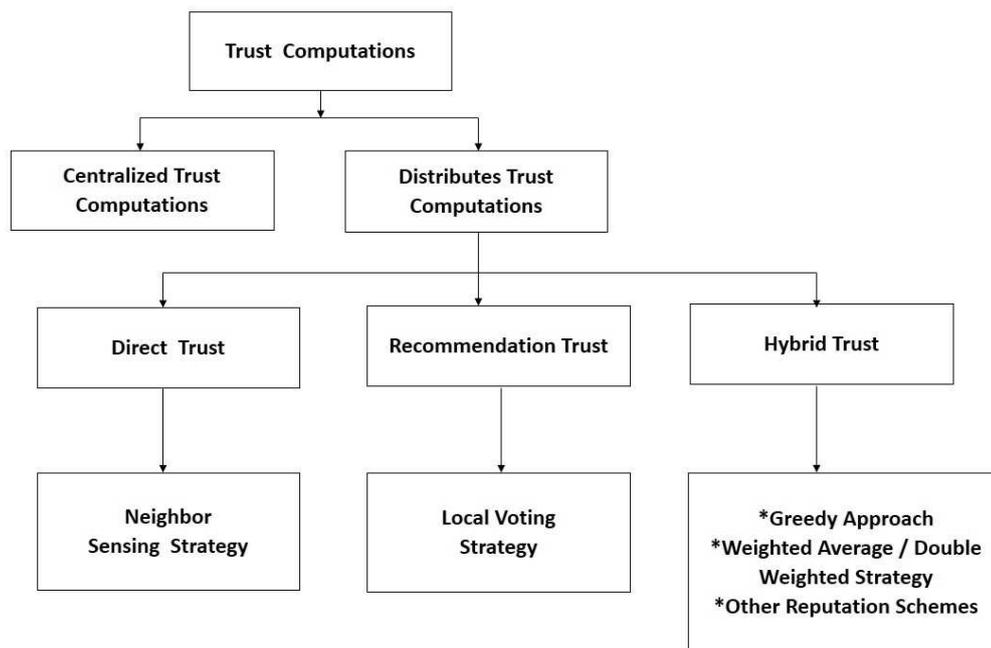


Figure 2: Taxonomy of trust computations

3 Trust computing approaches : an overview

As the HDNs is utilized by life-saving applications, the verification and validation of both data and node is necessary to ensure the authenticity of the incoming data. Therefore, trust management systems have an immense contribution in establishing the trustworthiness in the self-organized HDN. TMS ensures the network to be safe from selfish and malicious nodes. In order to enable trustworthiness between entities, measuring the trustworthiness is an essential task. With the uncertainties existing in the environment, the trustworthiness of a node or message has to be ascertained. Additionally, a security system has to be in place to thwart any attacks and to detect vulnerabilities. Adapting a reliable TMS model in HDN is the most effective way to have a secure and reliable service in the environment. Enormous efforts using different statistical and learning techniques have been built into TMS enabling seamless service.

Trust on a node can be ensured with subjective assessment on reliability and accuracy, as they traverse along the node in a given context. Trust computations are measured upon experience, recommendation and knowledge. The experience is measured at regular intervals in a trust table using direct trust strategy. Recommendation on trust sets its path as the values in the trust table are propagated to the other nodes. Knowledge of the total trust is evaluated based on the previous trust computations. The strategies used to measure the trust at various situations differ based on the environment and applicable mode of trust, the taxonomy is illustrated in Figure 2.

Based on the TMS schemes, trust computations are considered as an essential module to verify and validate both node and data. In all the trust computation techniques, a trust metric will be derived from the information shared by the peers. The reliability and credibility of the trust metric is based on the completeness of the chosen factors that are taken into account. In general, trust computations can be broadly classified into two categories based on the network compatibility, (i) centralized trust computations and (ii) distributed trust computations.

In centralized trust computations, a Central Authority (CA) will manage the behavior of the participating nodes and the trustworthiness is derived from the recommendations of CA. The

TMS based on this scheme will follow a master-slave property and the scheme will provide a reliable service to the nodes. However, there are limitations in these schemes, the entire system access can be denied when the CA is under attack or compromised. Also, having centralized servers in the distributed environment is not highly recommended as it could impact the dynamism of the nodes. Summarizing, in order to have a reliable service in the HDN environment, the node has to follow the peer-to-peer communication and has to be equally distributed in terms of receiving, maintaining and distributing the trust metric. For this purpose, a distributed trust computation offers a reliable service to the node and dynamically manages the trust metric in the HDN environment.

In distributed trust schemes, the nodes are capable of managing themselves and the trust metric is computed based on the input from the participating nodes alone. This paper discusses only the schemes related to the distributed trusted models. Further, trust computation in the distributed trust models follow three different strategies to compute the trust metric; (i) direct trust, (ii) indirect trust and (iii) hybrid trust.

In direct trust computations (DTC), the trustworthiness of a node will be computed from the direct experience. Based on the interactions with a single-hop node, the evidences are collected and stored from the direct input relay node. When a node wants to participate in the network for data exchange, a trust score for the node is computed from the direct interactions with the trustee node and the experiences are shared from the single-hop nodes.

In indirect trust computations (ITC), the node trustworthiness will be computed from the recommendations provided from the surrounding nodes. The nodes will constantly collect evidence and monitor the neighboring nodes behavior. The opinions collected are based on the past and present interaction with the node. If a trustee node wants to know about the trustworthiness of a participating node, it can request other nodes to share their opinions and the other nodes which have opinion on the participating node will share its information as recommendations based on their previous experience. Using this information, the trustee node can compute the trust metric for the participating node and can evaluate the nodes trustworthiness.

In hybrid trust computations (HTC), the results from DTC and ITC are utilized. The trust metric is computed using both the direct experience and recommendations from surrounding nodes are aggregated together. The basic idea in this scheme is to evaluate the trust metric using direct experience and verify the same using the opinions obtained from the ITC. Using this scheme, the selfish behavior nodes can be easily determined from the opinions provided by the surrounding nodes.

The existing methodologies in distributed trust computations with maximum reliability and efficiency are taken into consideration and a detailed comparison on different trust computations are discussed in Table 1-3. In this tables, the selected schemes are evaluated based on the system context, trust metrics, system merits, complexity and limitations of the system.

Based on the comparison, a brief discussion on the classifications of different networks with incorporated trust schemes are highlighted in Table 4. Heterogeneous networks are incorporated with different trust schemes to fulfill and improve the performance. The schemes are listed in table 4 with a brief explanation provided for VANETs, MANETs and wireless sensor networks (WSN). The schemes in Direct Trust Computations (DTC) category computes the trustworthiness of a node based on its own experience, as well as on the opinions obtained from the single-hop neighbor nodes.

Initially, every single node monitors the performance of other directly connected nodes and maintains a precise information table pertaining to the success ratio of the message-forwarding rate, message-strength, message-veracity and transaction rates. Subsequently, when a new target node requests for a participation with a trustee node in the network, the trustee node probes an opinion about the target node from its trustable single-hop neighbor nodes.

Table 1: Direct trust computations

Direct (Experience) Trust Computations						
Ref.	Approach	Context	Trust Metrics	Merits	Complexity	Limitations
[32]	Bayesian Approach	Packet forwarding based	Trust metric [0,1]	*windowing scheme is used*trustworthy route computation	* Route Computation * Opinion Calculation	Vulnerable to colluding and badmouth attack
[33]	Role and Experience Sensing	Cluster based aggregation	Trust value [0,1]	*identity-based aggregation *effective local action decision	*Cluster Head computation	Non-resistivity to selfish cluster-heads
[19]	A multi-faceted trust computing approach	Roles, experience, priority, and majority opinion based	Trust interval [-1,1]	*limits consulting advisors *majority consensus	*Computational complexity	Vulnerable to selfish and spoofing attacks
[20]	Bayesian probabilistic approach	Opinion analytics based	Trust value [0,1]	*beta and Gaussian reputation system *expert Opinion theory	*Computational complexity	Vulnerable to sybil attack
[6]	Markov chain approach	Trust value based	Trust value [0,1]	*secure authentication for group management	*Computational complexity	Trust decay factor not considered
[28]	Recommendation exchange approach	Neighbor trust based	Trust value [0,1]	*resistance to false *recommendation attack	*Computational complexity	Not suitable for more complex scenarios
[5]	Perron-Frobenius (PF) theorem	Message behavior analysis	Trust value [0,1]	*generate trust values full or partial data	*Computational complexity	Vulnerable to On-Off and Collusion attack

Table 2: Indirect trust computations

Indirect (recommendation) trust computations						
Ref.	Approach	Context	Trust Metrics	Merits	Complexity	Limitations
[12]	Analytical trust model	Node behavior model	Trust value [0,1]	*Entity and data approach with markov chain *Restrict selfish behavior	*Space Complexity *Computation Complexity	Demands constant monitoring and updating
[7]	Stochastic Petri net (SPN) technique	Trust chain optimization	Trust value [0,1]	*precise trust metric from social and QoS trust	*Computation Complexity	Vulnerable to selfish behavior
[24]	Geographic Hash Table	Consensus technique based	Trust opinion [0,1]	*Identify storage nodes *Decrease storage cost	*Computation Complexity	Susceptible to sophisticated attacks
[18]	Dempster-Shafer Theory(DST)	Attack resistant model	Trust value [0,1]	*Trust evaluation for both data and node	*Space Complexity *Computation Complexity	Vulnerable to sybil and collusion attacks
[21]	Dempster-Shafer theory (DST)	Trust path model	belief map [0,1]	*Detection of malicious nodes and benign nodes	*Computation Complexity	Demands constant monitoring and updating

Table 3: Hybrid trust computations

Hybrid Trust Computations						
Ref.	Approach	Context	Trust Metrics	Merits	Complexity	Limitations
[1]	Geometric mean-based	Node trust model	Trust metric [0,1]	*Estimate node trust level *Reliable routing	*Computation Complexity	Requires constant monitoring and updating
[2]	Stochastic Petri nets technique	Subjective trust Validation	Trust value [0,1]	*Resilient to black-hole, slandering and bad-mouthing attacks	*Space Complexity *Computation Complexity	System overhead and Cluster head failure
[4]	Dempster-Shafer theory (DST)	User centric privacy based	Trust value [0,1]	*Estimate trust event messages	*Computation Complexity	Vulnerable to selfish behavior
[26]	Bayesian statistical approach	Dynamic cluster based	Trust value [0,1]	*Filter dishonest recommendations	*Computation Complexity	Non-resistance to time and location dependent attacks
[27]	Fuzzy logic and graph theory	Node trust model	Trust value [0,1]	*Filtering algorithm for node *Decay method for routing	*Computation Complexity	Vulnerable to trust based attacks

Table 4: Trust in heterogeneous distributed networks

Heterogeneous distributed networks			
Trust schemes	VANET	MANET	WSN
Trust agent based scheme	*Effective decision making using agents on routing	*Manage reputation score for agents	*Estimate node trust level using agents and provides reliable routing
Hybrid trust scheme	*Estimate trust event messages *Helps to extricate malicious and selfish nodes *Detect inside attackers	*Distinguish trust nodes Decay method for routing	*Identify malicious sensor data
Recommendation trust scheme	*Enhance trustworthiness estimation *Detect fraudulent servers	*Precise trust metric from social and QoS trust	*Resistance to false attacks on sensor nodes
Direct trust scheme	*Generate trust values with full or partial data	*Secure authentication for group management	*Behavior analysis of nodes and computes trust score

The neighbor nodes which has previous experience with the target node, share its opinion about the target node. After collecting the opinions, the trustee node computes the trust metric for the target node and evaluates its trustworthiness.

In order to serve this purpose, techniques like Bayesian approach, Markov chain modelling and Perron-Frobenius (PF) models are employed in computing the trust metric in DTC. As the trustworthiness are computed only from the direct nodes, the derived trust metric is instantaneous and may not be accurate. Additionally, due to the limitations like mobility and connectivity issues that already exists in the HDN environment, the communication between two nodes are short lived and the opinions obtained from the neighbor nodes will be significantly minimal. These lead the trust metric in DTC provide a futile and biased results.

The Recommendation Trust Computations (RTC) category utilizes techniques and computes the trust value from the recommendations relayed from various indirect nodes. Every node follows a transitive path to derive the trustworthiness of other nodes. In order to find the trustworthiness of a target node, the trustee node broadcasts a message to the neighboring nodes, requesting an opinion about the target node. The nodes holding previous handshaking experience with the target node will share their opinions to the trustee node in a multi-hop fashion. Once the trustee node receives the opinions, it validates the received data based on the signal strength, distance, time lapse, similarity, energy level and closeness of the node. For this purpose, techniques like Dempster-Shafer theory (DST), Stochastic Petri Net (SPN) and Geographic Hash Tables are employed to verify the opinions and compute the trust metric for the target node. Due to fact that the recommendations are obtained from unknown and indirect nodes, the recommendations exhibit selfishness that are too hard to identify and validate. Moreover, attacks like Sybil and On-Off attacks would easily compromise the network and malicious nodes remain undetectable due to the indirect connections.

Consequently, in such above mentioned cases, the Hybrid Trust Computations (HTC) can be employed as a reliable technique that involves both the functionalities of direct and recommenda-

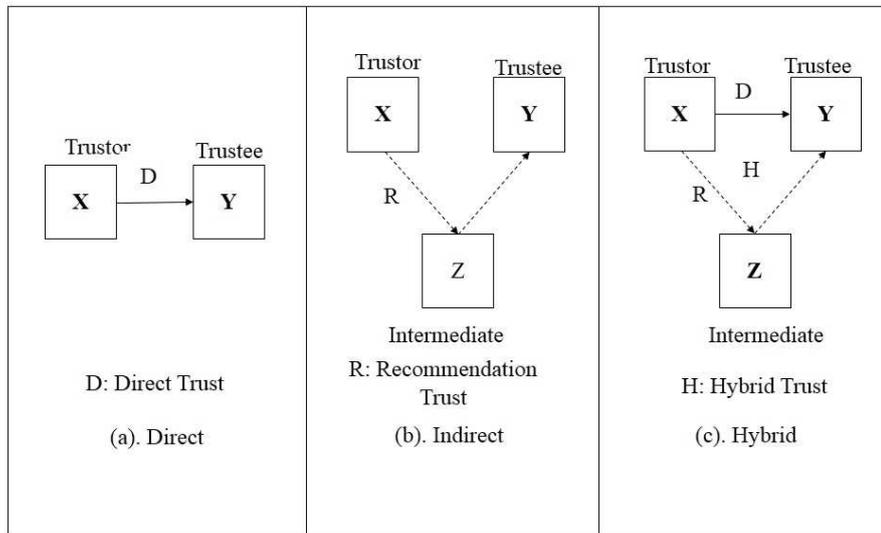


Figure 3: Taxonomy of trust computations

tion trust computation schemes. Along the lines of HTC technique, the trustee node computes the trustworthiness of a target node, procures opinions from the neighbor node without any limitations. Both direct referrals and indirect recommendations are utilized to derive the trust metric for the target node. Using this technique, the trustworthiness of both data and node could be verified.

Generally, the trustee node combines the recommendations from indirect vehicles and aggregates it to a single opinion. This aggregated metric will be further validated by the direct referrals that are in close proximity to the trustee node. In this technique, the derived trust metric for the target node is verified and validated by both the direct and the indirect nodes. Thereby, this multifaceted computation strategy efficiently computes and validates the trust metric, providing more resistance to the selfish behavior and thwarts the trust related attacks in the HDN environment.

4 Estimation of trust computations

Trust is computed based on two varieties of trust between the trustor and trustee nodes. They are direct trust and recommendation trust. In direct trust, the trust value is authenticated by a directed path between the trustor and the trustee. Whereas, in recommendation trust, the trust value is authenticated by third parties. In hybrid computation, both directed path trust value and third party recommended trust values are considered and the trust computation models are shown in Figure 3.

4.1 Direct trust computation (DTC)

Direct-trust computations are estimated with trust values ranging between +1 and -1. A tangent-hyperbolic function is used to estimate the trust value (a) built on the experience of trustor node (b), as shown in Equation (1). The equation is generalized, as a trustor node en-

Table 5: Trust elements and their properties

Element	Property
P_i	Path directed trust value
W_i	Prioritizes the level of importance on trust
τ_i	Node directed trust value; +1 for positive experience; -1 for negative experience
N	Unlimited, varies with environment
R_i	Recommended trust value range between +1 and -1

counters many trustee nodes and affect the evaluation of trust. The trust value will be computed using Equation (2), where P_i represents the direct-path experience with i^{th} trustee node, n is the total number of trustee nodes that participate with the trustor node, W_i represents the weight for the directed path prioritizing the experience and τ_i is +1 for a positive trust with the i^{th} trustee node and -1 for distrust. The trust element properties are listed in Table 5.

$$a = \frac{\sinh(b)}{\cosh(b)} = \tanh(b) \quad (1)$$

$$T_{DTC} = \tanh\left(\sum_{i=1}^n \tau_i \times P_i W_i\right) \quad (2)$$

The reason behind the choice of tangent hyperbolic function in comparison to the other identity activation function and logistic sigmoidal function being, the identity function maps output values range between $-\infty$ to ∞ , logistic sigmoidal function maps output range lies between 0 and 1, whereas tangent hyperbolic function maps output values range from -1 to 1 . Summarizing these, the trust values calculated in DTC use hyperbolic tangent function, to return trust values between -1 and $+1$. The corresponding hyperbolic tangents values ratio over hyperbolic sine represented as $\sinh()$ function and hyperbolic cosine represented as $\cosh()$ function.

4.2 Recommendation trust computation (RTC)

The trustor node has no directed path of experience with the trustee, therefore, it enquires a third node otherwise called the recommender node. When nodes do not encounter directed path experiences, they rely on third party recommendations. In this situation, to justify reasonable recommendation, the trustor always initiates multiple recommendation path experiences. The path experiences are not limited. The recommended trust values depend on the trustworthiness of the third party nodes with the trustor itself. Multiple trust recommendations from multiple third party nodes to the trustor widens the vision of the trustor in fixing the trust value for the trustee.

$$T_R = \sum_{i=1}^n (R_i \times T_{DTC_i}) \quad (3)$$

The recommender node when enquired delivers a trust value R_i on the i^{th} trustee node. The trustor holds T_{DTC} the direct trust value for the recommender node, as shown in Equation (3). The recommender nodes trust values reasonability is crosschecked by the trustor by inquiring/investigating several third party recommendation nodes.

Table 6: Trust computation (TC) summary

Trustee-id	Trustor-id	W_i	P_i	τ_i	TC (Expected) DTC	RTC	HTC	Actual Trust
166497924	725979012	4	1	1	0.999329	0.999329	0.999329	+1
166497924	258110	5	1	1	0.999909	0.999909	0.999909	+1
166497924	334400	5	1	1	0.999909	0.999909	0.999909	+1
166497924	492685	4	1	1	0.999329	0.999329	0.999329	+1
166497924	396581	4	1	1	0.999329	0.999329	0.999329	+1
166497924	328049	4	0.5	-1	0.995055	-0.99505	-0.59703	-1
166497924	534929	4	1	-1	0.995055	-0.99505	-0.59703	-1
166497924	768446340	4	1	-1	0.999329	-0.99933	-0.59960	-1
166497924	223418	3	0.5	-1	0.964028	-0.96403	-0.57842	-1
166563460	379568	5	1	1	0.999909	0.999909	0.999909	+1

4.3 Hybrid trust computation (HTC)

The trustor acquires direct trust values and recommendation trust values for the trustee node. The equations (3) and (2) are combined together in Equation (4) to arrive at hybrid computation trust value T_H . If T_{DTC} value is high and T_R value is low, then considerations on T_{DTC} is more when compared with T_R , or vice versa.

$$T_H = \{T_{DTC} = +\}, (1 \times |T_{DTC}| \times T_R) (-1 \leq T_{DTC} \leq 1) (-1 \leq T_R \leq 1) \quad (4)$$

In particular instances, when $T_{DTC} = 1$ and the trustor is assured complete certainty on the trust value, then the trustor will no longer consider the third party nodes recommendations. Otherwise, as $T_{DTC} = 0$ and the trustor is not assured complete certainty on the trust value, then the trustor will rely on the recommendations that are relayed from the third party nodes. In addition, the consideration of recommendations trust depends on the percentage of certainty values of direct trust.

5 Empirical results

The dataset from www.trustlet.org is explored and the trust values are computed using all the three computation strategies DTC, RTC, and HTC which are listed in Table 6. The expected/actual trust values are computed and plotted for visual illustration as shown in Figure 4.

6 Receiver operating characteristics (ROC) benchmarking trust computations

The operating characteristics are classified for the actual and estimated trust values with true positive (TP), true negative (TN) along with false positive (FP) and false negative (FN) accordingly in Table 7. The dataset from www.trustlet.org is being used to plot the ROC for 1000 instances, with equally distributed trust and distrust values (500 instances each). The characteristic of true positive denotes the proportion of trusted trust values correctly classified as +1, true negative is the proportion of distrust trust values correctly classified as -1, false positive

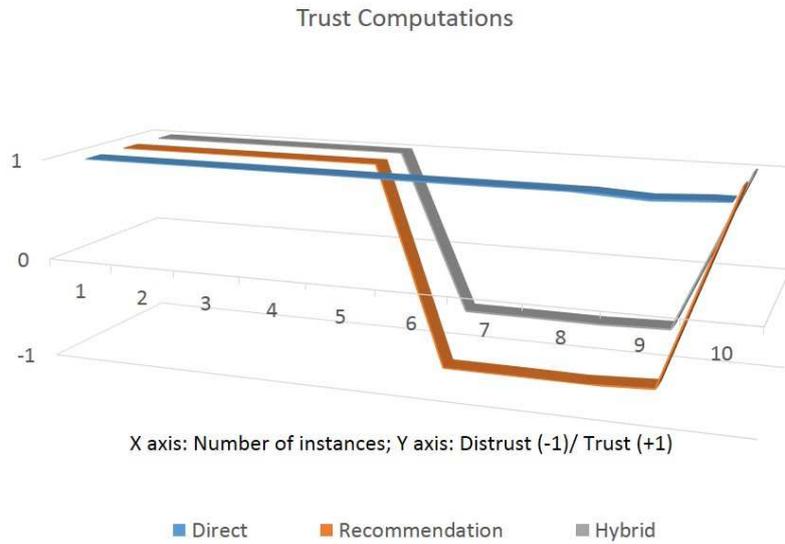


Figure 4: Trust computation summary (Direct, Recommendation, and Hybrid)

Table 7: Actual vs Estimated trust values

Actual Trust value		Estimated Trust value		
1 / -1	Trust	Distrust	Trust	Distrust
Trust	TP	FP	TP	FN
Distrust	FN	TN	FP	TN

shows the proportion of distrust trust values incorrectly classified as +1 and false negative as the proportion of trusted trust values incorrectly as -1.

The sensitivity or true positive rate (TPR) on trust values are calculated by dividing the true positive values by the summed up value of true positive and false negative. Later, specificity or true negative rate (TNR) on trust values are calculated by dividing the true negative values by the summed up value of true negative and false positive. The estimations on true positive/negative and false positive/negative for expected and actual trust/distrust values as shown in Figure 5(a) moving ahead further graph is plotted on sensitivity and specificity as shown in Figure 5(b). The values estimated for sensitivity and specificity are presented in table 8.

The true positive rates will be plotted along the Y axis and false positive rates will be plotted along the X axis. Hence, an ROC is defined by TRP and FPR. The ROC curve shown in Figure 6, illustrates no false negatives and no false positives on the left upper coordinate (0, 1).

A probability threshold has been used to classify the trust/distrust values. The values are classified based on probability. When the probability shoots the threshold it gets into class 1 (trust) and as it becomes low, class 0 (distrust). The group discrimination can also point some errors in certain cases. A ROC graph is constructed for all the three trust computation techniques described above. The computations are done for many varying thresholds. The ROC curve in Figure 6 showcases the area under curve (AUC) for all DTC, RTC and HTC. The graph with the highest AUC portion is the best recommended trust computation model. Hence, RTC

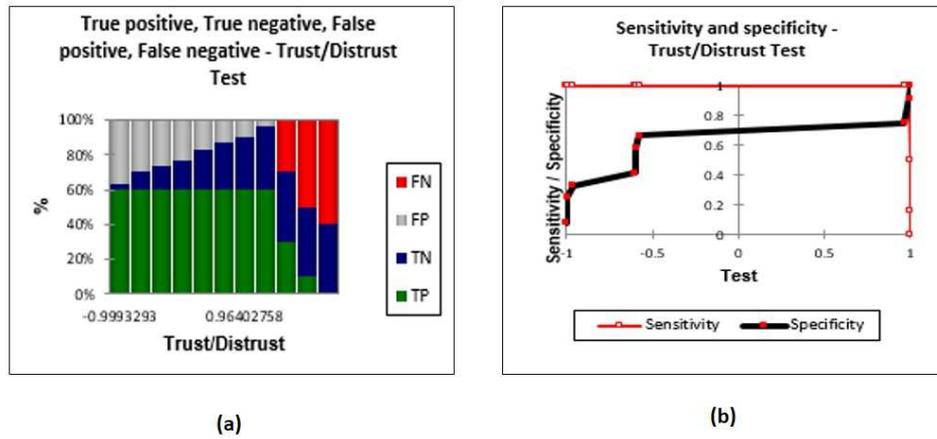


Figure 5: Benchmarking results

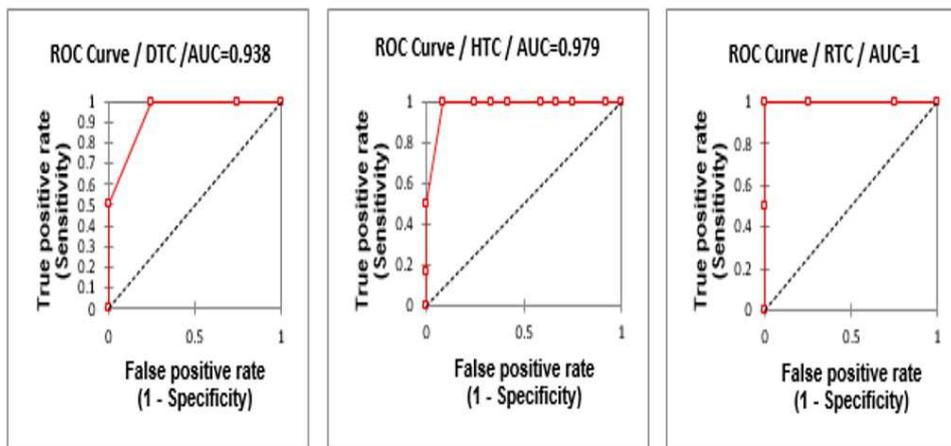


Figure 6: ROC Curve benchmarked on trust computations

Table 8: ROC analysis (Sensitivity vs Specificity)

Trust/Distrust	Sensitivity	Low (95%)	Up (95%)	Specificity	Low (95%)	Up (95%)
-0.999	1.000	0.789	1.000	0.083	0.000	0.379
-0.995	1.000	0.789	1.000	0.250	0.085	0.540
-0.964	1.000	0.789	1.000	0.333	0.138	0.612
-0.600	1.000	0.789	1.000	0.417	0.194	0.681
-0.597	1.000	0.789	1.000	0.583	0.319	0.806
-0.578	1.000	0.789	1.000	0.667	0.388	0.862
0.964	1.000	0.789	1.000	0.750	0.460	0.915
0.995	1.000	0.789	1.000	0.917	0.621	1.000
0.999	0.500	0.291	0.709	1.000	0.713	1.000
1.000	0.167	0.052	0.402	1.000	0.713	1.000

is pronounced as the best recommended trust computation to be adopted for trust management and is benchmarked with experimental results.

The complete ROC analysis is highlighted in Table 8, where the upper bound and lower bound of the sensitivity, specificity are specified. The negative/positive trust values are mapped towards both FPR and TPR. The upper/lower bounds ensure the trustworthiness for the received trust values between the nodes.

7 Conclusion and future enhancements

In this research, we have quantified the trust metrics between nodes in heterogeneous distributed network. We have defined three possible distributed trust computation strategies, we make an effort to zero-in on the best one and describe its performance by substantiating with highly accurate trustworthiness. The nodes in HDNs demonstrate trustworthiness among the participating nodes using DTC, RTC or HTC. The trust evaluation is performed and trust metrics are shared and recommended among the participating nodes. The trust values are measured using discrete/continuous series. The dataset from www.trustlet.org when computed with DTC, HTC and RTC delivers appropriate values for TP, TN, FP and FN. When the computed values of TPR and FPR are plotted on a ROC, the AUC presents itself. The AUC (area under curve) values are 0.938, 0.979 and 1.000 for DTC, HTC and RTC respectively. Interpreting the ROC graph with respective AUC regions, we conclude that the Recommendation Trust Computation (RTC) delivers a highly accurate trustworthiness score when compared with the other two trust computations. Hence, the RTC techniques trust computation performance is benchmarked with appropriate experimental results and we declare that RTC delivers promising trustworthiness and is more reliable over DTC and HTC. In future, we plan to extend recommendation trust computation (RTC) in VANETs to eliminate malicious entries and secure the vehicular network appropriately. The next step is to extend our research further ahead to include more efficient and dynamic approaches for trust computing, ensuring a high degree of trustworthiness.

Acknowledgments

This research was supported by the Roadway, Transportation, and Traffic Safety Research Center (RTTSRC) of the United Arab Emirates University (grant number 31R058). Data for this study comes from www.trustlet.org. The dataset holds 841,372 trust observations

with 717,667 trusts and 123,705 distrusts, which is archived at http://www.trustlet.org/datasets/extended_opinions/userrating.txt.gz.

The Trustlet, open research datasets on trust metrics has been cited by some researchers. Reference: P. Massa, K. Souren, M. Salvetti, D. Tomasoni. Scalable Computing: Practice and Experience.

Bibliography

- [1] Babu S.S., Raha A., Naskar M.K. (2011); Geometric mean based trust management system for WSNs (GMTMS), *Information and Communication Technologies (WICT)*, 444-449, 2011.
- [2] Bao F., Chen R., Chang M., Cho J.H. (2012); Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection, *IEEE Transactions on Network and Service Management*, 9(2), 69-83, 2012.
- [3] Baronti P., Pillai P., Chook V.W., Chessa S., Gotta A., Hu Y.F. (2007); Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards, *Computer communications*, 30(7), 1655-1695, 2007.
- [4] Chen Y.M., Wei Y.C. (2013); A beacon-based trust management system for enhancing user centric location privacy in VANETs, *Journal of Communications and Networks*, 15(2), 153-63, 2013.
- [5] Chaurasia B.K., Verma S., Tomar G.S. (2013); Trust computation in VANETs, *2013 International Conference on Communication Systems and Network Technologies (CSNT)*, 468-471, 2013.
- [6] Chang B.J., Kuo S.L. (2009); Markov chain trust model for trust-value analysis and key management in distributed multicast MANETs, *IEEE Transactions on Vehicular Technology*, 58(4), 1846-63, 2009.
- [7] Cho J.H., Swami A., Chen R. (2012); Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks, *Journal of Network and Computer Applications*, 35(3), 1001-12, 2012.
- [8] Conti M., Giordano S. (2014); Mobile ad hoc networking: milestones, challenges, and new research directions, *IEEE Communications Magazine*, 52(1), 85-96, 2014.
- [9] Cho J.H., Swami A., Chen R. (2011); A survey on trust management for mobile ad hoc networks, *IEEE Communications Surveys and Tutorials*, 13(4), 562-83, 2011.
- [10] Dias J.A., Rodrigues J.J., Zhou L. (2014); Cooperation advances on vehicular communications: A survey, *Vehicular Communications*, 1(1), 22-32, 2014.
- [11] Earle T.C. (2010); Trust in risk management: a model-based review of empirical research, *Risk Analysis*, 30(4), 541-74, 2010.
- [12] Gazdar T., Rachedi A., Benslimane A., Belghith A. (2012); A distributed advanced analytical trust model for VANETs, *Global Communications Conference (GLOBECOM)*, 201-206, 2012.

-
- [13] Hartenstein H., Laberteaux L.P. (2008); A tutorial survey on vehicular ad hoc networks, *IEEE Communications Magazine*, 46(6), 164-171, 2008.
 - [14] Hanzo L., Tafazolli R. (2007); A survey of QoS routing solutions for mobile ad hoc networks, *IEEE Communications Surveys and Tutorials*, 9(2), 50-70, 2007.
 - [15] Khan Pathan A.S. (2010); *Security of self-organizing networks: MANET, WSN, WMN, VANET*, Auerbach Publications, 2010.
 - [16] Lee U., Gerla M. (2010); A survey of urban vehicular sensing platforms, *Computer Networks*, 54(4), 527-44, 2010.
 - [17] Li F., Wang Y. (2007); Routing in vehicular ad hoc networks: A survey, *IEEE Vehicular Technology Magazine*, 2(2), 12-22, 2007.
 - [18] Li W., Song H. (2016); ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks, *IEEE Transactions on Intelligent Transportation Systems*, 17(4), 960-9, 2016.
 - [19] Minhas U.F., Zhang J., Tran T., Cohen R. (2011); A multifaceted approach to modeling agent trust for effective communication in the application of mobile ad hoc vehicular networks, *IEEE Transactions on Systems, Man, and Cybernetics*, 41(3), 407-20, 2011.
 - [20] Momani M., Aboura K., Challa S. (2007); RBATMWSN: recursive Bayesian approach to trust management in wireless sensor networks, *Intelligent Sensors, Sensor Networks and Information, 2007*, 347-352, 2007.
 - [21] Mali G., Misra S. (2016); TRAST: Trust-Based Distributed Topology Management for Wireless Multimedia Sensor Networks, *IEEE Transactions on Computers*, 65(6), 1978-91, 2016.
 - [22] Oliveira L.M, Rodrigues J.J.(2011); Wireless Sensor Networks: A Survey on Environmental Monitoring, *JCM*, 6(2), 143-51, 2011.
 - [23] Pantazis N.A, Nikolidakis S.A., Vergados D.D. (2013); Energy-efficient routing protocols in wireless sensor networks: A survey, *IEEE Communications Surveys and Tutorials*, 15(2), 551-91, 2013.
 - [24] Ren Y., Zadorozhny V.I, Oleshchuk VA, Li F.Y. (2014); A novel approach to trust management in unattended wireless sensor networks, *IEEE Transactions on Mobile Computing*, 13(7), 1409-23, 2014.
 - [25] Rezende C., Boukerche A., Pazzi R.W., Rocha B.P., Loureiro A.A. (2011); The impact of mobility on mobile ad hoc networks through the perspective of complex networks, *Journal of Parallel and Distributed Computing*, 71(9), 1189-200, 2011.
 - [26] Shabut A.M., Dahal K.P., Bista S.K., Awan I.U. (2015); Recommendation based trust model with an effective defense scheme for MANETs, *IEEE Transactions on Mobile Computing*, 14(10), 2101-15, 2015.
 - [27] Tan S., Li X., Dong Q. (2016); A Trust Management System for Securing Data Plane of Ad-Hoc Networks, *IEEE Transactions on Vehicular Technology*, 65(9), 7579-92, 2016.
 - [28] Velloso P.B., Laufer R.P., Cunha D.D., Duarte O.C., Pujolle G. (2010); Trust management in mobile ad hoc networks using a scalable maturity-based model, *IEEE Transactions on Network and Service Management*, 7(3), 172-85, 2010.

- [29] Yu H., Shen Z., Miao C., Leung C., Niyato D. (2010); A survey of trust and reputation management systems in wireless communications, *Proceedings of the IEEE*, 98(10), 1755-72, 2010.
- [30] Zhao S., Aggarwal A., Frost R., Bai X. (2012); A survey of applications of identity-based cryptography in mobile ad-hoc networks, *IEEE Communications Surveys and Tutorials*, 14(2), 380-400, 2012.
- [31] Zhang J. (2011); A survey on trust management for vanets, *Advanced Information Networking and Applications*, 105-112, 2011.
- [32] Zouridaki C, Mark B.L, Hejmo M, Thomas R.K. (2005); A quantitative trust establishment framework for reliable data packet delivery in MANETs, *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, DOI: 10.3233/JCS-2007-15102, 1-10, 2005.
- [33] Zhang J., Chen C., Cohen R. (2013); Trust modeling for message relay control and local action decision making in VANETs, *Security and Communication Networks*, DOI: 10.1002/sec.519, 6(1), 1-4, 2013.

An Uncertainty Measure for Interval-valued Evidences

W. Jiang, S. Wang

Wen Jiang*, **Shiyu Wang**

School of Electronics and Information,
Northwestern Polytechnical University
Xi'an, Shaanxi Province, 710072, China

*Corresponding author: jiangwen@nwpu.edu.cn

jiangwenpaper@hotmail.com, wangshiyu@mail.nwpu.edu.cn

Abstract: Interval-valued belief structure (IBS), as an extension of single-valued belief structures in Dempster-Shafer evidence theory, is gradually applied in many fields. An IBS assigns belief degrees to interval numbers rather than precise numbers, thereby it can handle more complex uncertain information. However, how to measure the uncertainty of an IBS is still an open issue. In this paper, a new method based on Deng entropy denoted as *UIV* is proposed to measure the uncertainty of the IBS. Moreover, it is proved that *UIV* meets some desirable axiomatic requirements. Numerical examples are shown in the paper to demonstrate the efficiency of *UIV* by comparing the proposed *UIV* with existing approaches.

Keywords: Dempster-Shafer theory, interval-valued belief structure, interval evidence, uncertainty measure, Deng entropy.

1 Introduction

Dempster-Shafer evidence theory, also known as D-S theory was proposed by Dempster [8] and extended by Shafer [45], it has received widespread attention and application in information processing [18,25,40,43,46,52]. As compared with classic probability theory, D-S theory allocates the belief to multi-subset proposition and does not require a priori information. Accordingly, D-S theory is used to process the uncertain information in many fields such as risk assessment [16,24,39,60], decision making [4,7,11,36,38,58], fault diagnosis [20,26,27,41,48,51], information fusion [2,9,12,19,35] and pattern classification [3,42,44,55].

Although the application of D-S theory has made considerable progress, there are still some common issues in urgent need to be solved. For instance, conflict processing should be taken into consideration when the obtained evidence is highly conflicting with each other [28,30,37,53], for we may get the count-intuitive results [29,59]. In view of this, many scholars have carried out extensive and profound research. Dencœux [15] considered the evidence expressed by fuzzy-valued which acquire lots of application [57]. Moreover, the classic D-S theory demands precise belief degrees, yet it is not always available in some cases. For instance, in the decision making, the experts sometimes cannot provide an accurate assessment because of the lack of information. At this time, an interval-valued belief structure (IBS) [56] is more suitable for dealing with the uncertainty problem. About extending the D-S theory to IBS, many scholars have carried out some research such as Dencœux [14] put forward a set of concepts about interval-valued belief structure and initially explored the combination and the uncertainty of it. Lee & Zhu [34] proposed the combination of two interval evidence. Wang [54] proposed the approach to combine and standardize the interval evidence in one step. However, it must be noted that there are still many unresolved issues about interval-valued belief structure.

One of the crucial issues is uncertainty measurement [10,50]. From the perspective of information theory, Klir elaborated the inner relationship between uncertainty and information [33].

Bronevich [5, 6] discussed some of the issues and applications of the measurement of the uncertainty for imprecise probabilities. However, even how to measure the uncertainty of the mass function in D-S theory is still a considerable issue [21, 23]. Dubois & Prade presented weighted Hartley entropy [17] to express the non-specificity of BPA. Klir & Wierman [32] explored five axiomatic requirements for the uncertainty measures including range, probabilistic consistency, set consistency, additivity and subadditivity, respectively. Abellán & Masegosa [1] have extended the axiomatic approach by appending new monotonicity requirement. Among existing uncertainty measures, aggregated uncertainty (AU) [22] and ambiguity measure (AM) [31] are two representative measures, yet they have their own shortcomings, such as low sensitivity and high computing complexity. Deng entropy [13] divided the belief for each focal element into all potential subsets. On the other hand, there is not many approaches about the uncertainty measure for interval-valued belief structure. Denoeux [14] proposed a rudiment to measure the uncertainty, yet it was immature and lacked the mathematical proof. Song [49] defined the axiomatic requirements for uncertainty measure and presented a new method IU to measure the uncertainty. But IU lost part of the information and may cause the counter-intuitive result because of the transformation from belief structures to probability distributions. Accordingly, how to effectively measure the uncertainty of interval-valued belief structure is still an open issue. In this paper, a new method based on Deng entropy to measure the uncertainty of the interval-valued belief structure and its axiomatic proof is presented as well. Several examples are shown to illustrate the rationality and effectiveness of the method.

The remainder of this paper is organized as follows. Section 2 starts with a brief presentation of D-S evidence theory and some other indispensable related concepts. In Section 3, we present a new method to measure the uncertainty of the interval-valued belief structure. Some numerical examples are given to demonstrate the validity of our new method in Section 4. Conclusions are summarized in Section 5.

2 Preliminaries

2.1 Dempster-Shafer evidence theory

Dempster-Shafer evidence theory, as introduced by Dempster [8] and expanded later by Shafer [45], has been widely used in dealing with uncertainty. Some basic concepts in D-S theory are introduced as follows.

Let Θ be a finite set of worlds, which is called a frame of discernment (FOD). Θ consists of some propositions, which are mutually exclusive and exhaustive, and indicated by

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_N\}. \quad (1)$$

Let 2^Θ be the power set of Θ , namely

$$2^\Theta = \{\emptyset, \theta_1, \theta_2, \dots, \theta_N, \{\theta_1 \cup \theta_2\}, \dots, \{\theta_1 \cup \theta_2 \cup \dots \cup \theta_i\}, \dots, \Theta\}. \quad (2)$$

For a FOD Θ , a mass function is a mapping $m : 2^\Theta \rightarrow [0, 1]$, it is also called the basic probability assignment (BPA) or the belief structure. BPA must satisfy the following condition

$$\begin{cases} \sum_{A \in 2^\Theta} m(A) = 1, \\ m(\emptyset) = 0. \end{cases} \quad (3)$$

For a BPA, its belief function $Bel : 2^\Theta \rightarrow [0, 1]$ is defined as

$$Bel(A) = \sum_{B \subseteq A} m(B), \quad (4)$$

the plausibility function $Pl : 2^\Theta \rightarrow [0, 1]$ is defined as

$$Pl(A) = 1 - Bel(\bar{A}) = \sum_{B \cap A \neq \emptyset} m(B). \tag{5}$$

Assume there are two BPAs m_1 and m_2 with the same FOD, it can be combined by Dempster’s combination rule.

$$m(A) = \frac{1}{1 - k} \sum_{B \cap C = A} m_1(B)m_2(C), \tag{6}$$

where

$$k = \sum_{B \cap C = \emptyset} m_1(B)m_2(C). \tag{7}$$

k is between $[0,1]$, which is called the coefficient of conflict. When $k = 1$, Dempster’s combination rule will be invalid.

2.2 Interval-valued belief structure

Uncertainty is sometimes no longer described by a unique belief structure, but by a convex set of belief structures verifying certain constraints. A set of concepts of interval-valued belief structure (IBS) is given as follows [14].

Let Θ be the frame of discernment, F_1, F_2, \dots, F_N be N subsets of Θ and $[a_i, b_i]$ be N intervals with $0 \leq a_i \leq b_i \leq 1$, ($i = 1, 2, \dots, N$). An interval-valued belief structure (IBS) m is a belief structure on Θ such that

$$a_i \leq m(F_i) \leq b_i, \tag{8}$$

where

$$0 \leq a_i \leq b_i \leq 1, i = 1, 2, \dots, N, \tag{9}$$

$$\sum_{i=1}^N a_i \leq 1 \text{ and } \sum_{i=1}^N b_i \geq 1, \tag{10}$$

$$m(A) = 0 \quad \forall A \notin \{F_1, F_2, \dots, F_N\}. \tag{11}$$

Obviously, m are non-empty imposes certain constraints on the a_i and b_i . If the singleton m is an IBS with $a_i = b_i = m(F_i)$ for $\forall F_i$, m degenerates to a precise belief structure (BS). An IBS means the interval associated to each subset of Θ is $[0,1]$. It may be interpreted as reflecting “second-order” ignorance, that is, ignorance of what the state of belief of an agent may be.

Let m be an interval-valued belief structure, namely $a_i \leq m(F_i) \leq b_i$ for $i = 1, 2, \dots, N$. If $\forall k \in \{1, 2, \dots, N\}$, a_i and b_i satisfy

$$\sum_{i=1}^N a_i + (b_k - a_k) \leq 1, \tag{12}$$

$$\sum_{i=1}^N b_i - (b_k - a_k) \geq 1. \tag{13}$$

Then, m is called a normalized interval-valued belief structure (NIBS) [54].

For a non-normalized interval-valued belief structure m , which violates Eq. (10), it can be normalized by following equations.

$$\hat{a}_i = \frac{a_i}{a_i + \sum_{j=1, j \neq i}^N b_j}, i = 1, 2, \dots, N, \tag{14}$$

$$\hat{b}_i = \frac{b_i}{b_i + \sum_{j=1, j \neq i}^N a_j}, \quad i = 1, 2, \dots, N. \quad (15)$$

On the other side, if m has already satisfied Eq. (10), but not Eqs. (12) and (13), it can be normalized by following two equations.

$$\hat{a}_i = \max \left\{ a_i, 1 - \sum_{j=1, j \neq i}^N b_j \right\}, \quad i = 1, 2, \dots, N, \quad (16)$$

$$\hat{b}_i = \min \left\{ b_i, 1 - \sum_{j=1, j \neq i}^N a_j \right\}, \quad i = 1, 2, \dots, N. \quad (17)$$

The concepts of belief function and plausibility function may easily be generalized to an interval-valued belief structure. Since these quantities are linear combinations of belief masses constrained in closed intervals, their ranges are both closed intervals.

Let m be a normalized interval-valued belief structure on Θ . For $\forall A \in \Theta$, its belief function and plausibility function are defined respectively as

$$Bel(A) = \left[\min \sum_{F_i \subseteq A} m(F_i), \max \sum_{F_i \subseteq A} m(F_i) \right], \quad (18)$$

$$Pl(A) = \left[\min \sum_{F_i \cap A \neq \emptyset} m(F_i), \max \sum_{F_i \cap A \neq \emptyset} m(F_i) \right], \quad (19)$$

where

$$\min \sum_{F_i \subseteq A} m(F_i) = \max \left[\sum_{F_i \subseteq A} a_i, \left(1 - \sum_{F_i \not\subseteq A} b_i \right) \right], \quad (20)$$

$$\max \sum_{F_i \subseteq A} m(F_i) = \min \left[\sum_{F_i \subseteq A} b_i, \left(1 - \sum_{F_i \not\subseteq A} a_i \right) \right], \quad (21)$$

$$\min \sum_{F_i \cap A \neq \emptyset} m(F_i) = \max \left[\sum_{F_i \cap A \neq \emptyset} a_i, \left(1 - \sum_{F_i \cap A = \emptyset} b_i \right) \right], \quad (22)$$

$$\max \sum_{F_i \cap A \neq \emptyset} m(F_i) = \min \left[\sum_{F_i \cap A \neq \emptyset} b_i, \left(1 - \sum_{F_i \cap A = \emptyset} a_i \right) \right]. \quad (23)$$

2.3 Deng entropy

Since Shannon entropy [47] was proposed to quantify the expected value of the information volume contained in a message, it has become a significant approach to measure the uncertainty. However, for a mass function in D-S theory, Shannon entropy cannot calculate its uncertainty because the mass function includes multiple subset elements. To measure the uncertainty of the mass function, Deng [13] proposed Deng entropy as follows

$$E_d(m) = - \sum_{A \subseteq \Theta} m(A) \log_2 \frac{m(A)}{2^{|A|} - 1}, \quad (24)$$

where m is a BPA defined on the frame of discernment Θ , A is the focal element of m , and $|A|$ is the cardinality of A .

Deng entropy is analogous with the classical Shannon entropy, but the belief for each focal element A is divided by $(2^{|A|} - 1)$ which indicates the potential supports in A .

3 Proposed uncertainty measure for interval-valued belief structures

In an interval-valued belief structure, the belief degree for each subset is not a precise value but an interval. So contrasted with single-valued belief structures, an interval-valued belief structure is more vague and more uncertain, since an IBS has the “second-order” ignorance. Thus, how to measure the uncertainty of the IBS is an essential issue. In this paper, A new method to measure the uncertainty of IBS is proposed.

Definition 1. Let m be a normalized interval-valued belief structure on the frame of discernment $\Theta = \{F_1, F_2, \dots, F_N\}$, and it satisfies $a_i \leq m(F_i) \leq b_i$, which means the accurate belief $m(F_i) \in [a_i, b_i]$. Then the uncertainty measure of the IBS m is as follows

$$UIV(m) = \sum_{i=1}^{2^N} \left[\min_{m(F_i) \in [a_i, b_i]} \widetilde{E}_d(F_i), \max_{m(F_i) \in [a_i, b_i]} \widetilde{E}_d(F_i) \right], \quad (25)$$

where

$$\widetilde{E}_d(F_i) = -m(F_i) \log_2 \frac{m(F_i)}{2^{|F_i|} - 1}, \quad (26)$$

and $|F_i|$ is the cardinality of F_i .

The new measurement method we proposed is based on Deng entropy, not Shannon entropy, so our method is more suitable to handle the proposition of multi-subsets. For Deng entropy, the belief of the focal element $m(F_i)$ is divided by the number of potential subsets $2^{|F_i|} - 1$ that demonstrates the non-specificity of the evidence. The more single elements are contained in focal elements, it is obvious that the greater the uncertainty. The term $-m(F_i) \log_2 m(F_i)$ is analogous to Shannon entropy and is the measure of discord of the evidence. Thereby, it is also appropriate to quantify the uncertainty of interval-valued belief structure. Obviously, UIV is an interval number. Its value embodies the belief distribution of different proposition in IBS, and its interval length reflects the ambiguity generated by the belief expressed in intervals.

Song [49] proposed the axiomatic requirements for a measure of uncertainty for a normalized interval-valued belief structure m .

Theorem 2. Let U be a measure of uncertainty for a normalized interval-valued belief structure m on the FOD $\Theta = \{\theta_1, \theta_2, \dots, \theta_N\}$, then U must content the following condition.

1. Whenever the NIBS defines a precise probability distribution, U degenerates to Shannon entropy.
2. When the NIBS assigned to all subsets of Θ are completely unknown, its uncertainty is maximum. Thus, U reaches its maximum value.
3. If the NIBS assigns to a certain singleton of Θ is 1, the uncertainty of it is 0. Therefore, U gets its minimum value 0.

It will be shown that our new method satisfies the above-mentioned axiomatic requirement.

Proof:

1. If the NIBS m defines a precise probability distribution on $\Theta = \{F_1, F_2, \dots, F_N\}$,

$$\begin{aligned} UIV(m) &= \sum_{i=1}^{2^N} [\widetilde{E_d}(F_i), \widetilde{E_d}(F_i)] \\ &= \sum_{i=1}^N -m(F_i) \log_2 \frac{m(F_i)}{2^{|F_i|} - 1} \\ &= - \sum_{i=1}^N m(F_i) \log_2 m(F_i). \end{aligned}$$

From the above equation, we can see that when m defines a precise belief structure on Θ , UIV degenerates to Deng entropy. Moreover, when m defines a precise probability distribution, UIV degenerates to Shannon entropy.

2. When the NIBS assigned to all subsets of Θ are completely unknown, that is for $\forall F_i \in 2^\Theta$, $[a_i, b_i] = [0, 1]$. It is apparent that

$$UIV(m) = \sum_{i=1}^{2^N} [\min_{m(F_i) \in [0,1]} \widetilde{E_d}(F_i), \max_{m(F_i) \in [0,1]} \widetilde{E_d}(F_i)]$$

where

$$\widetilde{E_d}(F_i) = -m(F_i) \log_2 \frac{m(F_i)}{2^{|F_i|} - 1},$$

and it can be seen as a function of F_i , now the independent variable $F_i \in [0, 1]$. Therefore, the minimum value of $\widetilde{E_d}(F_i)$ is 0 and the maximum value may be mutative with the change of $|F_i|$ yet it can always get its maximum value for any F_i , that is

$$\max_{m(F_i) \in [0,1]} \widetilde{E_d}(F_i) = \max \widetilde{E_d}(F_i)$$

So,

$$UIV(m) = [0, \sum_{i=1}^{2^N} \max \widetilde{E_d}(F_i)].$$

In this case, the value and the interval length of UIV are both the maximum value, which indicates that m is totally uncertain, that is, its uncertainty is maximum.

3. If the NIBS assigns to a certain singleton of Θ is 1, there is no harm in supposing that for singleton F_k , $m(F_k) = 1$, and the belief degree of all the rest subsets is 0. Then

$$\begin{aligned} UIV(m) &= \sum_{i=1}^{2^N} [\min_{m(F_i) \in [a_i, b_i]} \widetilde{E_d}(F_i), \max_{m(F_i) \in [a_i, b_i]} \widetilde{E_d}(F_i)] \\ &= [\min_{m(F_i) \in [1,1]} \widetilde{E_d}(F_k), \max_{m(F_i) \in [1,1]} \widetilde{E_d}(F_k)] + \sum_{\substack{i=1 \\ i \neq k}}^{2^N} [\min_{m(F_i) \in [0,0]} \widetilde{E_d}(F_i), \max_{m(F_i) \in [0,0]} \widetilde{E_d}(F_i)] \\ &= -1 \times \log_2 \frac{1}{2^1 - 1} - \sum_{\substack{i=1 \\ i \neq k}}^{2^N} (0 \times \log_2 \frac{0}{2^{|F_i|} - 1}) = 0 \end{aligned}$$

In fact, the UIV at this time is not 0, but $[0,0]$. This result thoroughly explains the m under this circumstance is totally definite, and it is also in line with intuition.

Table 1: NIBSs in Example 3

	$\{F_1\}$	$\{F_2\}$	$\{F_3\}$	$\{F_1, F_3\}$
m_1	[0.2,0.3]	[0.1,0.35]	[0.4,0.6]	[0,0]
m_2	[0.2,0.3]	[0.1,0.35]	[0.35,0.7]	[0,0]
m_3	[0.2,0.3]	[0.1,0.35]	[0,0]	[0.4,0.6]
m_4	[0.2,0.3]	[0.1,0.35]	[0.2,0.3]	[0.2,0.3]

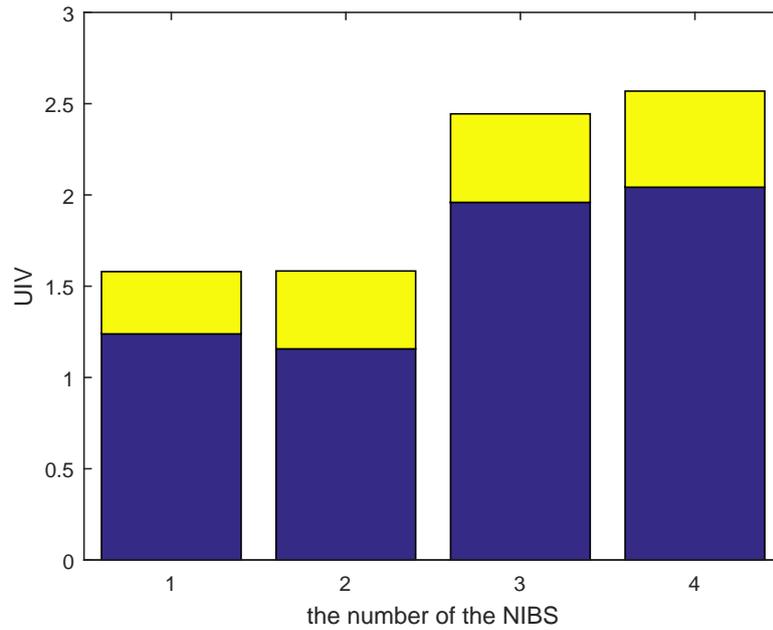


Figure 1: The UIV of each NIBS in Example 3

□

4 Numerical examples

In this section, several examples are given to demonstrate the effectiveness of UIV .

Example 3. Assume a frame of discernment $\Theta = \{F_1, F_2, F_3\}$, and consider four NIBSs defined as shown in Table 1.

We can calculate the UIV of the NIBSs as follows

$$\begin{aligned}
 UIV(m_1) &= [1.239, 1.580] & UIV(m_2) &= [1.157, 1.583] \\
 UIV(m_3) &= [1.959, 2.444] & UIV(m_4) &= [2.042, 2.569]
 \end{aligned}$$

and they are also graphically shown in Fig. 1. The yellow portion represents the endpoint of the interval of the UIV . The range of $UIV(m_2)$ is larger than $UIV(m_1)$ from the figure, since $m_2(F_3)$ is more uncertain than $m_1(F_3)$. However, the value of $UIV(m_2)$ is close to $UIV(m_1)$ because the belief distribution in m_1 and m_2 are about the same. Considering $UIV(m_3)$ and $UIV(m_1)$, it is obvious that both the length and the value of $UIV(m_3)$ are bigger since the

Table 2: UIV in Example 4

Cases	UIV
$A=\{1\}$	[2.080,3.803]
$A=\{1,2\}$	[3.216,4.886]
$A=\{1,2,3\}$	[3.949,5.864]
$A=\{1,2,\dots,4\}$	[4.609,6.743]
$A=\{1,2,\dots,5\}$	[5.238,7.581]
$A=\{1,2,\dots,6\}$	[5.851,8.400]
$A=\{1,2,\dots,7\}$	[6.458,9.209]
$A=\{1,2,\dots,8\}$	[7.062,10.013]
$A=\{1,2,\dots,9\}$	[7.663,10.816]

multi-element can take along more uncertainty than single element even though in the same interval. It is worth noting that compared with $UIV(m_3)$, $UIV(m_4)$ is close but slightly larger. Although a great deal of belief are assigned on the multi-element in m_3 and it conveys illegibility, the allocation form which distributes the belief to more subsets is more excursive and this result is we take for granted.

Example 4. Suppose that we have a frame of discernment $\Theta = \{1, 2, \dots, 10\}$. A NIBS m is shown as follows.

$$m(2, 3) = [0.1, 0.25], \quad m(A) = [0.6, 0.8], \quad m(\Theta) = [0.1, 0.2]$$

where A is a varying subset of Θ . A starts at $A = \{1\}$, increases one more element every time and ending with $A = \{1, 2, \dots, 9\}$. The UIV of m are shown in Table 2 and Fig. 2. The yellow portion represents the endpoint of the interval of the UIV .

From Fig. 2, the result shows that UIV increases monotonically with the number of elements in A . This is rational because the more elements contained in a subset, the more uncertain it is. From the example it can be seen that UIV is capable of reflecting such a feature.

In the first two examples, some superior properties are demonstrated. Then an example from Song [49] are used to illustrate our proposed UIV and contrast it with Song's uncertainty measure IU . The formula of Song's measurement are shown as follows.

Definition 5. Let m be a normalized interval-valued belief structure on the FOD $\Theta = \{F_1, F_2, \dots, F_N\}$, and it satisfies $a_i \leq m(F_i) \leq b_i$. Then IU of the IBS m is as follows

$$IU(m) = \sum_{i=1}^N \left(-\frac{a_i + b_i}{2} \log_2 \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2} \right) \quad (27)$$

Example 6. The example Song used in the paper is shown in Table 3, and to make a comparison with Song's method, the consequents of IU and our new method UIV are both demonstrated in Table 4.

For the NIBSs from m_1 to m_5 , we can see their belief intervals are completely consistent, merely the corresponding subsets are disparate. The uncertainty degree IU proposed by Song, are so similar that it is difficult to measure the uncertainty accurately. Moreover, the belief assignment of m_1 and m_5 are entirely different, yet their IU are almost identical. For UIV , m_5 with more belief assigned to multiple elements has a higher uncertainty, m_2 and m_3 take

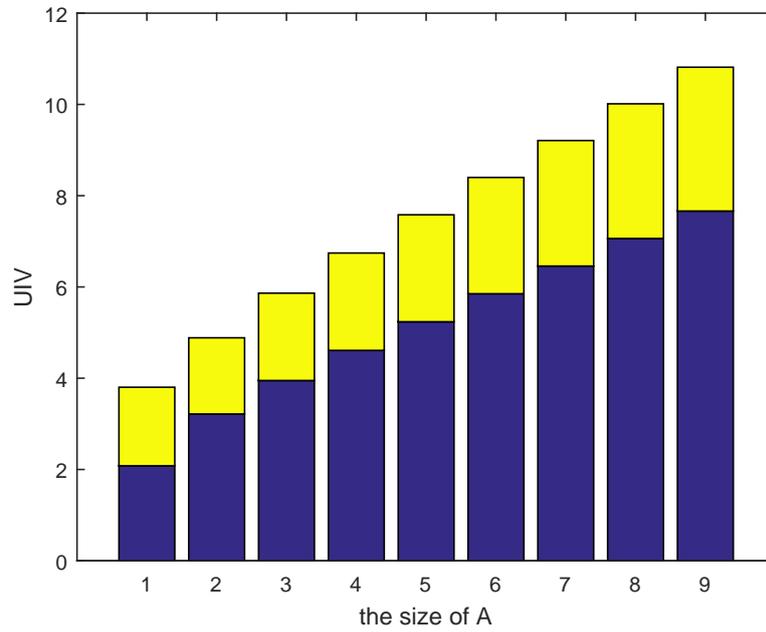


Figure 2: *UIV* in Example 4

Table 3: NIBSs in Song's example ($\Theta = \{F_1, F_2, F_3\}$)

	$\{F_1\}$	$\{F_2\}$	$\{F_3\}$	$\{F_1, F_2\}$	$\{F_1, F_3\}$	$\{F_2, F_3\}$	$\{F_1, F_2, F_3\}$
m_1	[0.2,0.4]	[0.1,0.3]	[0.3,0.6]	[0,0.1]	[0,0]	[0,0]	[0,0]
m_2	[0.2,0.4]	[0,0]	[0,0]	[0,0.1]	[0.3,0.6]	[0,0]	[0.1,0.3]
m_3	[0,0]	[0.1,0.3]	[0,0]	[0,0.1]	[0.2,0.4]	[0.3,0.6]	[0,0]
m_4	[0,0]	[0,0]	[0.3,0.6]	[0,0.1]	[0.3,0.6]	[0.1,0.3]	[0.2,0.4]
m_5	[0,0]	[0,0]	[0,0]	[0,0.1]	[0.3,0.6]	[0.1,0.3]	[0.2,0.4]
m_6	[0,1]	[0,1]	[0,1]	[0,0]	[0,0]	[0,0]	[0,0]
m_7	[0,0]	[0,0]	[0,0]	[0,0]	[0,0]	[0,0]	[0,1]

Table 4: *IU* and *UIV* of the NIBSs

	<i>IU</i>	<i>UIV</i>
m_1	1.930	[1.239,2.073]
m_2	1.609	[2.074,3.778]
m_3	1.889	[2.110,3.568]
m_4	1.575	[1.714,3.181]
m_5	1.939	[2.513,4.532]
m_6	3	[0,1.592]
m_7	1.793	2.807

Table 5: NIBSs and their IU and UIV in Example 7

	$\{F_1\}$	$\{F_2\}$	$\{F_1, F_2\}$	IU	UIV
m_1	[0.1,0.25]	[0.3,0.4]	[0.4,0.5]	1.163	[2.016,2.323]
m_2	[0.2,0.3]	[0.4,0.45]	[0.2,0.4]	1.163	[1.764,2.213]

second place, as well as m_1 is the most precise of these NIBSs. Furthermore, UIV is suitable for measurement for the reason that the difference in calculated values is significant and thus has a degree of discrimination.

Another detail of concern is m_6 and m_7 . The uncertainty of m_7 is low, while the maximum uncertainty degree occurs on m_6 . The cause of this consequence as Song said in [49], "This is caused by the transformation from belief structures to Bayesian belief structures, which will cause information loss." $UIV(m_6)$ is comparatively small because m_6 only distribute the belief to singleton. In addition, m_7 actually is not a normalized interval-valued belief structure. It turns into a NIBS $m_7(\{F_1, F_2, F_3\}) = 1$ by Eqs. (16) and (17). After standardization, $UIV(m_7)$ is a precise number and its uncertainty can be effectively measured.

Example 7. Let a frame of discernment be $\Theta = \{F_1, F_2\}$. Two NIBSs, their IU and UIV are shown in Table 5.

We can calculate that both two Bayesian belief structures of m_1 and m_2 are $m(a) = [0.3, 0.5]$, $m(b) = [0.5, 0.65]$, and IU is not competent to measure the uncertainty in this situation. Because for two unrelated NIBSs with significant differences in the degree of uncertainty, their IU are equivalent. Through the above analysis, it is found that UIV is more reasonable to measure the uncertainty of the interval-valued belief structures.

5 Conclusion

D-S theory has been widely used in information processing and information fusion. In many applications, we can only obtain an interval-valued belief structure instead of a basic probability assignment defined on single values, due to lack of information and some other reasons. It is indispensable to measure the uncertainty of the IBS, there is still an open issue.

The main contribution of this paper is a new method based on Deng entropy, UIV is proposed to measure the uncertainty of an IBS. It is proved that UIV meets some axiomatic properties. Numerical examples are illustrated to show the effectiveness of UIV and discuss its characteristic. Moreover, it is found that UIV is more reasonable and sensitive in comparison with existing methods.

Acknowledgments

The work is partially supported by National Natural Science Foundation of China (Grant No. 61671384), Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016JM6018), Aviation Science Foundation (Program No. 20165553036), Fundamental Research Funds for the Central Universities (Program No. 3102017OQD020).

Bibliography

- [1] Abellán J., Masegosa A. (2008); Requirements for total uncertainty measures in Dempster-Shafer theory of evidence, *International Journal of General Systems* 37, 733–747, 2008.
- [2] Aminravan F., Sadiq R., Hoorfar M., Rodriguez M. J., Najjaran H. (2015); Multi-level information fusion for spatiotemporal monitoring in water distribution networks, *Expert Systems with Applications*, 42(7), 3813–3831, 2015.
- [3] Bhattacharyya A., Saraswat V. K., Manimaran P., Rao S. B.(2015); Evidence theoretic classification of ballistic missiles, *Applied Soft Computing*, 37, 479–489, 2015.
- [4] Bolar A., Tesfamariam S., Sadiq R. (2013); Condition assessment for bridges: a hierarchical evidential reasoning (HER) framework, *Structure & Infrastructure Engineering*, 9(7), 1–19, 2013.
- [5] Bronevich A., Klir G. J. (2010); Measures of uncertainty for imprecise probabilities: An axiomatic approach, *International journal of approximate reasoning*, 51(4), 365–390, 2010.
- [6] Bronevich A., Lepskiy A.(2015); Imprecision indices: axiomatic, properties and applications, *International Journal of General Systems*, 44(7-8) (2015) 812–832, 2015.
- [7] Chao X., Peng Y., Kou G.(2017); A Similarity Measure-based Optimization Model for Group Decision Making with Multiplicative and Fuzzy Preference Relations, *International Journal of Computers Communications & Control*, 12(1), 26–40, 2017.
- [8] Dempster A. P.(1967); Upper and lower probabilities induced by a multivalued mapping, *Annals of Mathematical Statistics*, 38(2), 325–339, 1967.
- [9] Deng X., Han D., Dezert J., Deng Y., Shyr Y.(2016); Evidence combination from an evolutionary game theory perspective, *IEEE Transactions on Cybernetics*, 46(9), 2070–2082, 2016.
- [10] Deng X., Xiao F., Deng Y., Lei L., Quan W.(2017); An improved distance-based total uncertainty measure in belief function theory, *Applied Intelligence* 46(4), 898–915, 2017.
- [11] Deng X., Jiang W., Zhang J.(2017); Zero-sum matrix game with payoffs of Dempster-Shafer belief structures and its applications on sensors, *Sensors*, 17(4), 922, doi:10.3390/s17040922, 2017.
- [12] Deng X., Jiang W.(2017); An evidential axiomatic design approach for decision making using the evaluation of belief structure satisfaction to uncertain target values, *International Journal of Intelligent Systems*, , Article in press, doi:10.1002/int.21929, 2017.
- [13] Deng Y. (2016); Deng entropy, *Chaos Solitons & Fractals*, 91, 549–553, <https://doi.org/10.1016/j.chaos.2016.07.014>, 2016.
- [14] Denœux T. (1999); Reasoning with imprecise belief structures, *International Journal of Approximate Reasoning*, 20(1), 79–111, 1999.
- [15] Denœux T.(2000); Modeling vague beliefs using fuzzy-valued belief structures, *Fuzzy Sets and Systems*, 116(2), 167–199, 2000.
- [16] Du W., Cao X., Hu M., Wang W.(2009); Asymmetric cost in snowdrift game on scale-free networks, *EPL (Europhysics Letters)*, 87(6), 60004, 2009.

-
- [17] Dubois D., Prade H.(1985); A note on measures of specificity for fuzzy sets, *International Journal of General System*, 10(4), 279–283, 1985.
- [18] Dzitac I.(2015); The fuzzification of classical structures: a general view, *International Journal of Computers Communications & Control*, 10(6), 772-788, 2015.
- [19] Frikha A., Moalla H.(2014); Analytic hierarchy process for multi-sensor data fusion based on belief function theory, *European Journal of Operational Research*, 241(1), 133–147, 2014.
- [20] Fu C., Wang Y.(2015); An interval difference based evidential reasoning approach with unknown attribute weights and utilities of assessment grades, *Computers & Industrial Engineering*, 81, 109–117, 2015.
- [21] Han D., Dezert J., Duan Z.(2016); Evaluation of probability transformations of belief functions for decision making, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(1), 93–108, 2016.
- [22] Harmanec D., Klir G. J.(1994); Measuring total uncertainty in dempster-shafer theory: A novel approach, *International journal of general system*, 22(4), 405–419, 1994.
- [23] Hwang C., Yang M.(2015); Belief and plausibility functions on intuitionistic fuzzy sets, *International Journal of Intelligent Systems*, 31(6), 556–568, 2015.
- [24] Jiang W., Yang Y., Luo Y., Qin X.(2015); Determining basic probability assignment based on the improved similarity measures of generalized fuzzy numbers, *International Journal of Computers Communications & Control*, 10(3), 333–347, 2015.
- [25] Jiang W., Wei B., Zhan J., Xie C., Zhou D.(2016); A visibility graph power averaging aggregation operator: A methodology based on network analysis, *Computers & Industrial Engineering*, 101, 260–268, 2016.
- [26] Jiang W., Xie C., Zhuang M., Shou Y., Tang Y.(2016); Sensor data fusion with Z-numbers and its application in fault diagnosis, *Sensors*, 16(9), Article ID 1509, 2016.
- [27] Jiang W., Xie C., Zhuang M., Tang Y.(2017); Failure mode and effects analysis based on a novel fuzzy evidential method, *Applied Soft Computing*, 57, 672–683, 2017.
- [28] Jiang W., Wang S., Liu X., Zheng H., Wei B.(2017); Evidence conflict measure based on OWA operator in open world, *PloS one* 12(5), b) e0177828, 2017.
- [29] Jiang W., Zhan J.(2017); A modified combination rule in generalized evidence theory, *Applied Intelligence*, 46(3), 630–640, 2017.
- [30] Jiang W., Zhuang M., Xie C., Wu J. (2017); Sensing attribute weights: A novel basic belief assignment method, *Sensors*, 17(4), 721, doi: 10.3390/s17040721, 2017.
- [31] Jousselme A. L., Liu C., Grenier D., Bosse E.(2006); Measuring ambiguity in the evidence theory, *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 36(5), 890–903, 2006.
- [32] Klir G. J., Wierman M. J.(1999); *Uncertainty-based information: elements of generalized information theory*, Springer Science & Business Media, 1999.
- [33] Klir G. J.(2005); *Uncertainty and information: foundations of generalized information theory*, John Wiley & Sons, 2005.

-
- [34] Lee E. S., Zhu Q.(1992); An interval Dempster-Shafer approach, *Computers & Mathematics with Applications*, 24(7), 89–95, 1992.
- [35] Lefebvre A., Sannier C., Corpetti T.(2016); Monitoring urban areas with sentinel-2A data: Application to the update of the copernicus high resolution layer imperviousness degree, *Remote Sensing*, 8(7), doi:10.3390/rs8070606, 2006.
- [36] Liu W., Liu H., Li L.(2017); A Multiple Attribute Group Decision Making Method Based on 2-D Uncertain Linguistic Weighted Heronian Mean Aggregation Operator, *International Journal of Computers Communications & Control*, 12(2), 254–264, 2017.
- [37] Mo H., Lu X., Deng Y.(2016); A generalized evidence distance, *Journal of Systems Engineering and Electronics*, 27(2), 470–476, 2016.
- [38] Moosavian A., Khazaee M., Najafi G., Kettner M., Mamat R.(2015); Spark plug fault recognition based on sensor fusion and classifier combination using Dempster-Shafer evidence theory, *Applied Acoustics*, 93, 120–129, 2015.
- [39] Neshat A., Pradhan B.(2015); Risk assessment of groundwater pollution with a new methodological framework: application of Dempster-Shafer theory and GIS, *Natural Hazards*, 78(3), 1565–1585, 2015.
- [40] Nguyen K., Denman S., Sridharan S., Fookes C.(2015); Score-level multibiometric fusion based on Dempster-Shafer theory incorporating uncertainty factors, *IEEE Transactions on Human-Machine Systems*, 45(1), 132–140, 2015.
- [41] Peng M., Chi K. T., Shen M., Xie K.(2013); Fault diagnosis of analog circuits using systematic tests based on data fusion, *Circuits, Systems, and Signal Processing*, 32(2), 525–539, 2013.
- [42] Perez A., Tabia H., Declercq D., Zanotti A.(2016); Using the conflict in Dempster-Shafer evidence theory as a rejection criterion in classifier output combination for 3d human action recognition, *Image & Vision Computing*, 55(2), 149–157, 2016.
- [43] Ran Y., Li X., Lu L., Li Z.(2012); Large-scale land cover mapping with the integration of multi-source information based on the Dempster-Shafer theory, *International Journal of Geographical Information Science*, 26(1), 169–191, 2012.
- [44] Roychowdhury S., Koozekanani D. D., Parhi K. K.(2014); DREAM: Diabetic retinopathy analysis using machine learning, *IEEE Journal of Biomedical and Health Informatics*, 18(5), 1717–1728, 2014.
- [45] Shafer G.(1976); *A Mathematical Theory of Evidence*, New Jersey, Princeton University Press, 1976.
- [46] Shahpari A., Seyedin S. A.(2015); Using mutual aggregate uncertainty measures in a threat assessment problem constructed by Dempster-Shafer network, *IEEE Transactions on Systems Man & Cybernetics Systems*, 45(6), 877–886, 2015.
- [47] Shannon C. E.(1948); A mathematical theory of communication, *Bell Labs Technical Journal*, 5(3), 3–55, 1948.
- [48] Si L., Wang Z., Tan C., Liu X.(2014); A novel approach for coal seam terrain prediction through information fusion of improved D-S evidence theory and neural network, *Measurement*, 54, 140–151, 2014.

-
- [49] Song Y., Wang X., Lei L., Yue S.(2016); Uncertainty measure for interval-valued belief structures, *Measurement*, 80, 241–250, 2016.
- [50] Song Y., Wang X., Wu W., Lei L., Quan W.(2017); Uncertainty measure for Atanassov's intuitionistic fuzzy sets, *Applied Intelligence*, 46(4), 757–774, 2017.
- [51] Vasu J. Z., Deb A. K., Mukhopadhyay S.(2015); MVEM-based fault diagnosis of automotive engines using Dempster-Shafer theory and multiple hypotheses testing, *IEEE Transactions on Systems Man & Cybernetics Systems*, 45(7), 977–989, 2015.
- [52] Wang J., Hu Y., Xiao F., Deng X., Deng Y.(2016); A novel method to use fuzzy soft sets in decision making based on ambiguity measure and Dempster-Shafer theory of evidence: An application in medical diagnosis, *Artificial intelligence in medicine*, 69, 1–11, 2016.
- [53] Wang J., Xiao F., Deng X., Fei L., Deng Y.(2016); Weighted Evidence Combination Based on Distance of Evidence and Entropy Function, *International Journal of Distributed Sensor Networks*, 12(7), <https://doi.org/10.1177/155014773218784>, 2016.
- [54] Wang Y. M., Yang J. B., Xu D. L., Chin K. S.(2007); On the combination and normalization of interval-valued belief structures, *Information Sciences*, 177(5), 1230–1247, 2007.
- [55] Xu S., Jiang W., Deng X., Shou, Y.(2017); A modified Physarum-inspired model for the user equilibrium traffic assignment problem, *Applied Mathematical Modelling*, , In Press, DOI: 10.1016/j.apm.2017.07.032, 2017.
- [56] Yager R. R.(2001); Dempster–Shafer belief structures with interval valued focal weights, *International Journal of intelligent systems*, 16(4), 497–512, 2001.
- [57] Yang J. B., Wang Y. M., Xu D. L., Chin K. S.(2006); The evidential reasoning approach for MADA under both probabilistic and fuzzy uncertainties, *European Journal of Operational Research*, 171(1), 309–343, 2006.
- [58] Yang J. B., Xu D. L.(2013); Evidential reasoning rule for evidence combination, *Artificial Intelligence*, 205, 1–29, 2013.
- [59] Zadeh L. A.(1986); A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination, *AI Magazine*, 7(2), 85–90, 1986.
- [60] Zhang X., Mahadevan S., Deng X.(2017); Reliability analysis with linguistic data: An evidential network approach, *Reliability Engineering & System Safety*, 162, 111–121, 2017.

Phasing of Periodic Tasks Distributed over Real-time Fieldbus

S.-H. Lee, H.-W. Jin, K. Kim, S. Lee

Sang-Hun Lee

Hyundai Mobis
17-2, 240 Mabuk-ro, Giheung-gu, Yongin-si,
Gyeonggi-do 16891, Korea
sanghun@mobis.co.kr

Hyun-Wook Jin*

Department of Computer Science and Engineering
Konkuk University
120 Neungdong-ro, Gwangjin-gu, Seoul 05029, Korea
*Corresponding author: jinh@konkuk.ac.kr

Kanghee Kim

Department of Smart Systems Software
Soongsil University
369 Sangdo-ro, Dongjak-gu, Seoul 06978, Korea
khkim@ssu.ac.kr

Sangil Lee

Agency for Defense Development
Songpa P.O.Box 132, Seoul 05661, Korea
happyjoy@add.re.kr

Abstract: In designing a distributed hard real-time system, it is important to reduce the end-to-end delay of each real-time message in order to ensure quick responses to external inputs and a high degree of synchronization among cooperating actuators. In order to provide a real-time guarantee for each message, the related literature has focused on the analysis of end-to-end delays based on worst-case task phasing. However, such analyses are too pessimistic because they do not assume a global clock. With the assumption that task phases can be managed by using a global clock provided by emerging real-time fieldbuses, such as EtherCAT, we can try to calculate the optimal task phasing that yields the *minimal* worst-case end-to-end delay. In this study, we propose a heuristic to manage the phase offsets in the distributed tasks to reduce the theoretical end-to-end delay bound. The proposed heuristic reduces the search time for a solution by identifying time intervals where actual communication occurs among inter-dependent tasks. Furthermore, to analyze the distribution of end-to-end delays in different phases, we implemented a simulation tool. The simulation results showed that the proposed heuristic can reduce worst-case end-to-end delay as well as jitter in end-to-end delays.

Keywords: task phasing, end-to-end delay, real-time, fieldbus, EtherCAT.

1 Introduction

Fieldbuses are industrial networks widely used in small-to-large-scale distributed control systems, such as vehicles and factory automation systems. While the fieldbuses define only the physical, the data link, and the application layers, they provide real-time features for deterministic communication latency.

Emerging real-time fieldbuses provide highly accurate clock synchronization as a key feature to support a global clock in distributed systems. For instance, the distributed clock of EtherCAT provides accurate clock synchronization with errors smaller than a few tens of nanoseconds [2] [3].

Such accurate clock synchronization provides a novel opportunity to adjust task phasing across distributed computing nodes in order to better synchronize dependent tasks (e.g., multi-axis motion control). However, obtaining optimal task phasing across distributed nodes has not been accorded sufficient attention in the literature, whereas worst-case task phasing in each node has been extensively studied [22] [19] [8]. This is due to coarse-grained clock synchronization on legacy networks; moreover, the performance analysis of distributed tasks by considering jitters in task execution time is not straightforward. Thus, researchers have focused on the analysis of worst-case end-to-end delays while assuming no global clock. However, once we assume a precise global clock for all nodes, we can investigate the problem of finding an optimal phase combination of tasks in order to reduce worst-case end-to-end delays. Though few efforts have been devoted in recent research to enhancing task phasing, these either assume no jitters for tasks [4] or apply phasing to a single node [11] [10].

In this paper, we describe a novel off-line algorithm that heuristically searches for near-optimal task phasing with respect to the worst-case end-to-end delay. One novelty of the proposed algorithm is that it takes into account the jitters in execution times of distributed tasks while trying to reduce search time to find near-optimal task phasing. An exhaustive search needs to consider all combinations of the variable execution times of tasks. To reduce search time, our algorithm deals with feasible time ranges of communication.

To analyze the distribution of end-to-end delays in different phase combinations, we implemented a tool that simulated variable execution times of tasks and the behavior of real-time fieldbus. In the analysis, we assumed a real-time fieldbus, such as EtherCAT, that comprised a master and multiple slave nodes. We modeled each node as a set of periodic tasks scheduled on a uniprocessor by a fixed-priority scheduling algorithm. The results of analysis showed that our algorithm can efficiently search for a phase combination that can significantly reduce both worst-case end-to-end delay and jitter.

The rest of this paper is organized as follows: In Sec. 2, we provide an overview of EtherCAT and discuss related work. In Sec. 3, we describe the system model that we assume in this study. We outline our algorithm to find a near-optimal task phase combination and describe our simulation tool for the analysis of end-to-end delays in Sec. 4 and Sec. 5, respectively. Sec. 6 contains details of a case that involves finding an optimal node phase combination for a given industrial task set. Finally, in Sec. 7, we present conclusions.

2 Background

2.1 EtherCAT

In this paper, we consider EtherCAT as the reference model for real-time fieldbuses. EtherCAT is an open specification that covers the physical, the data link and the application layers of the communication protocol stack. Owing to its low cost, compatibility with the TCP/IP protocol and efficiency, EtherCAT is becoming popular in many real-time applications such as factory automation systems and motion control systems.

An EtherCAT network consists of a single master node and multiple slave nodes allowing various network topologies such as line, daisy chain, and star. The master node generates EtherCAT telegrams, each of which is encapsulated in Ethernet frames. A telegram may include several datagrams for different slave nodes, where a datagram includes an address to specify the corresponding slave node. The master node transmits the resulting Ethernet frames through the standard Ethernet interface. Each slave node then reads from and writes to the datagram on the fly in a specially designed network controller called the EtherCAT slave controller. The EtherCAT slave controller can forward Ethernet frames to the next slave node at the hardware

level while performing such read-and-write operations. The propagation delay at the controller is less than $1\mu s$ [9]. Thus, EtherCAT provides high-speed message relay between adjacent nodes, and can guarantee deterministic communication delays.

EtherCAT provides a global clock synchronization feature called *Distributed Clock* that synchronizes the clocks of the master and slave nodes to the reference clock represented by a selected slave. The distributed clock controller can generate an interrupt called the *Sync* signal as the global reference clock [18]. The signal can operate in either cyclic or single-shot mode, where the period and shift time are specified in nanosecond resolution. Thus, we can adjust the release time (i.e., phase) of a task by means of the *Sync* signal.

2.2 Related work

Sung et al. [18] showed the potential of the global clock provided by EtherCAT for synchronized control processes, but they considered only communication tasks while assuming no preemption by higher-priority tasks. A few researchers recently suggested heuristic approaches that iteratively adjusted task phases for isochronous control on EtherCAT [11] [10]. Although they could improve actuation jitters, the suggested approaches could not be applied at design phase for a system yet to be constructed because those are based on an on-line algorithm. In addition, they targeted only task phasing on the master node by assuming a simple task set in the slave nodes. Craciunas et al. [4] tried to decide the optimal offset of tasks in terms of utility, but did not consider jitters in execution times of tasks. In this study, we propose an off-line algorithm that searches for a near-optimal task phase combination on both master and slave nodes while considering jitters in execution times of tasks.

Traditionally, task phasing was addressed in order to obtain worst-case response times of tasks. Under fixed-priority scheduling, tasks with varying inter-release times were analyzed at a worst-case time instant, called a critical instant [13], since the release time of each task was assumed not to be managed. In a distributed real-time system, such an assumption was extended to every node because the unpredictability of the release time of a task was exacerbated by the jitters introduced by varying response times of the preceding task [19] [8]. In addition, it was assumed that the communicating tasks on different nodes were synchronized by messages. There were also studies to analyze the best-case response time [1] [16], but these studies did not target distributed systems.

Several studies were conducted on the analysis and simulation of real-time industrial networks [17] [6] [7] [22]. These studies, however, did not address task phasing to obtain minimum worst-case end-to-end delays in the design of distributed real-time systems. It is important to reduce analysis time to find an optimal solution. For example, Garcia-Valls et al. [21] [20] proposed selectively checking for dynamic systems in real-time reconfiguration. In this study, we try to reduce analysis time by considering only feasible communication time intervals.

3 System model

We consider a distributed real-time system that interconnects a master node and $N - 1$ slave nodes with a real-time fieldbus, which is common in many industrial applications. For example, in a motion control system, the master node generates position commands to describe motion trajectory and slave nodes serve as motor drives to respond to these commands. In the following, we explain the task and the network models, and define end-to-end delay.

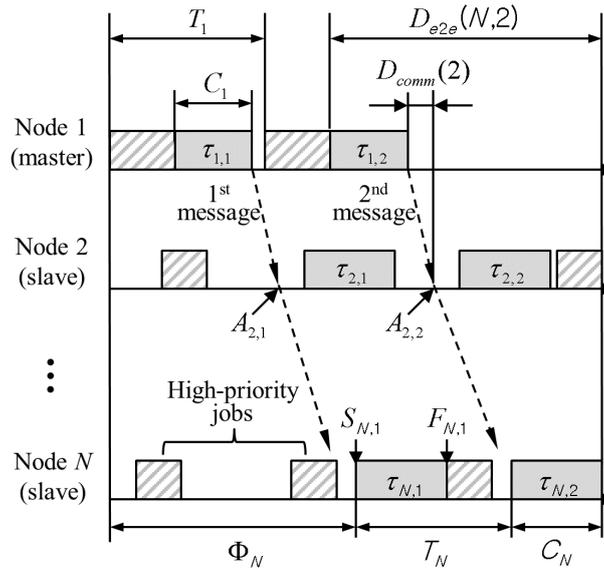


Figure 1: System model

3.1 Task model

We consider a set of periodic tasks for both master and slave nodes and these may have different sets of tasks. We denote a communication task by $\tau_n = (C_n^{min}, C_n^{max}, T_n, P_n, \Phi_n)$, where C_n^{min} and C_n^{max} are its minimum and maximum execution times, T_n is the period, P_n is its priority, and Φ_n is its phase with respect to a certain reference time. The jitter in the execution time represented by $C_n^{max} - C_n^{min}$ is caused not only by jitters in system software, but also by different code branches in tasks. We assume that the relative deadline of each task is equal to its period. Each node has only one communicating task. The rationale for this is that many industrial network protocols do not support multiplexing and de-multiplexing between tasks [5] [14]. Moreover, as the name implies, only the master node is assumed to initiate message transmission. Each slave node may only piggyback some data on message passing through the node on the fly, but cannot initiate message transmission. Thus, there exists a chain of communicating tasks across nodes (i.e., $\tau_1 \rightarrow \tau_2 \rightarrow \dots \rightarrow \tau_n$) for each communication period as shown in Fig. 1.

Each communication task gives rise to an infinite sequence of jobs. $\tau_{n,i}$ denotes the i^{th} job of τ_n . The start time of $\tau_{n,i}$ is denoted by $S_{n,i}$ and the finish time by $F_{n,i}$. A sender job on the master node (i.e., $\tau_{1,i}$) sends a message to the network device at the finish time, $F_{1,i}$, while receiver jobs on the slave nodes (i.e., $\tau_{n,i}$, $1 < n \leq N$) look up the message from the message queue at the start time, $S_{n,i}$. We do not suppose that the receiver jobs are synchronously released by a message; rather, we assume that the jobs are released by a periodic task scheduler as many real systems do. If there is no message arrived, then the receiver job may finish without further processing. The periods of the sender and receiver tasks are identical because both operate on the same message initiated by the master node. The message arrival time of the i^{th} message on node n is represented by $A_{n,i}$. For task scheduling, we assume a fixed-priority scheduling algorithm, such as Rate Monotonic [13]. Since each task is assigned a static priority, the execution of a task assigned a lower priority can be preempted by one assigned a higher priority, which were modeled as the release jitter in several previous studies. Due to such preemption and jitters in execution times of tasks, there can be many potential time points at which $\tau_{n,i}$ starts. Thus, we define the start time of $\tau_{n,i}$ as a set of time points, i.e., $S_{n,i} = \{s_{n,i}^{min}, \dots, s_{n,i}^{max}\}$. Similarly, the

finish time of $\tau_{n,i}$ is defined as a set $F_{n,i} = \{f_{n,i}^{min}, \dots, f_{n,i}^{max}\}$. The one-to-many correspondence $\mathcal{R} : S_{n,i} \rightarrow F_{n,i}$ maps a start time point of $\tau_{n,i}$ to a set of potential finish time points. That is, $\mathcal{R}(s_{n,i}) = \{f_{n,i}^1, \dots, f_{n,i}^m\} \subseteq F_{n,i}$. and $\mathcal{R}(s_{n,i}^{min}) \cup \dots \cup \mathcal{R}(s_{n,i}^{max}) = F_{n,i}$. we also define the arrival time $A_{n,i} = \{a_{n,i}^{min}, \dots, a_{n,i}^{max}\}$.

3.2 Network model

Emerging real-time fieldbuses, such as EtherCAT, guarantee deterministic communication delays between any two nodes through two design choices. The first involves connecting any two adjacent nodes with a dedicated link. That is, there are no interfering nodes on the link and, thus, no packet collisions. The second design choice is such that each slave node n conducts cut-through switching to relay packets between node $n-1$ and $n+1$, instead of, store-and-forward switching. This switching scheme when implemented at the hardware level eliminates the likelihood that the internal software operations of each slave contribute to end-to-end communication delay. As a result, EtherCAT guarantees deterministic communication delays from the master to any slave n , which is analyzed by Prytz et al. [15] as follows:

$$D_{comm}(n) = D_t(s) + (n-1) \times D_f + D_r(s), \quad (1)$$

where $D_{comm}(n)$ is the communication delay from the master's memory to slave n 's memory as shown in Fig. 1, $D_t(s)$ is the total transmission time of a message of size s on the master network device, and D_f is the total forwarding time of a message on the slave side, which is a constant of $1 \mu s$, irrespective of the message size. $D_r(s)$ is the total reception time of a message of size s on the slave network controller. We do not have to consider any delay in piggybacking data on the message at the slave node because this operation is conducted on the fly while the message is being forwarded by the network controller.

3.3 Problem statement

In this paper, we define end-to-end delay of the i^{th} message as the time from the beginning of the sender job at the master to the completion of the receiver job that consumes the i^{th} message at slave. Suppose $\Phi_1 + D_{comm}(n) \leq \Phi_n$, the i^{th} message is consumed by either $\tau_{n,i}$ or $\tau_{n,i+1}$ at slave n because periods of sender and receiver tasks are the same as mentioned in Sec. 3.1.

We define $D_{e2e}^0(n, i)$ to be the set of end-to-end delay values observed when $\tau_{n,i}$ consumes the message on slave, i.e.,
for all $f_{1,i}$ and $s_{n,i}$ that satisfy

$$f_{1,i} + D_{comm}(n) \leq s_{n,i}, \quad (2)$$

$$D_{e2e}^0(n, i) = \{f_{n,i} - s_{1,i} \mid f_{n,i} \in \mathcal{R}(s_{n,i}), s_{1,i} \in \mathcal{R}^{-1}(f_{1,i})\}, \quad (3)$$

where \mathcal{R}^{-1} is the inverse correspondence of \mathcal{R} defined in Sec. 3.1 (i.e., $\mathcal{R}^{-1} : F_{n,i} \rightarrow S_{n,i}$).

We also define $D_{e2e}^{+1}(n, i)$ to be the set of end-to-end delays when $\tau_{n,i+1}$ receives the message as the message arrives after the beginning of $\tau_{n,i}$, i.e.,
for all $f_{1,i}$ and $s_{n,i}$ that satisfy

$$f_{1,i} + D_{comm}(n) > s_{n,i}, \quad (4)$$

$$D_{e2e}^{+1}(n, i) = \{f_{n,i+1} - s_{1,i} \mid f_{n,i+1} \in F_{n,i+1}, s_{1,i} \in \mathcal{R}^{-1}(f_{1,i})\}. \quad (5)$$

Thus, we define the end-to-end delay $D_{e2e}(n, i)$ of the i^{th} message as follows:

$$D_{e2e}(n, i) = D_{e2e}^0(n, i) \cup D_{e2e}^{+1}(n, i) = \{d_{n,i}^{min}, \dots, d_{n,i}^{max}\}. \quad (6)$$

In this study, we want to analyze the worst-case end-to-end delay $\max(d_{N,1}^{max}, \dots, d_{N,H/T_1}^{max})$, where $d_{n,i}^{max} = \max(D_{e2e}(n, i))$, and H is the hyperperiod defined as the least common multiple of the periods of all tasks. Then, we aim to focus on finding an optimal combination of task phases leading to the minimal worst-case end-to-end delay:

$$\begin{aligned} & \text{Minimize } \max(d_{N,1}^{max}, \dots, d_{N,H/T_1}^{max}) \\ & (\Phi_1, \Phi_2, \dots, \Phi_N) \\ & \text{subject to } U_n \leq U_{upper} \text{ for } 1 \leq n \leq N, \end{aligned} \quad (7)$$

where U_n and U_{upper} represent the processor utilization of a given task set on node n and upper-bound of the utilization which guarantees schedulability, respectively. This is challenging given that $D_{e2e}(n, i)$ has very large number of elements due to variable execution times of tasks and preemption by higher-priority tasks. In the next section, we describe the proposed algorithm that searches for a near-optimal task phase combination without calculating all elements of $D_{e2e}(n, i)$.

4 Algorithm for optimal task phasing

In this section, we propose a heuristic algorithm for optimal task phasing with respect to minimal worst-case end-to-end delay. The algorithm can efficiently consider the jitter in execution times of tasks. If we exhaust all possible combinations of the variable execution times of tasks, it requires a long time to analyze the worst-case end-to-end delay for a given task phase combination. To avoid this, prior to the actual analysis of the worst-case end-to-end delay of a phase combination, we first determine the time intervals where communication can occur and efficiently find the worst-case end-to-end delay by using this information.

4.1 Overview

Fig. 2 shows the steps of our algorithm. In order to find the optimal phases of tasks, we heuristically calculate the worst-case end-to-end delays for different phase combinations. A phase combination is denoted by $Phase_i = \{1 + \Delta_{1,i}, 2 + \Delta_{2,i}\}$, where n is an initial sequence of task phases on node n , and $\Delta_{n,i}$ is the i^{th} sequence of variations to be added to n . If we analyze end-to-end delays for all possible phase combinations, we need to examine $H^{\Sigma t}$ cases, where H is the hyperperiod, and Σt is the number of all tasks in the system. To reduce the number of search cases, in the first step, we assume that the slave nodes have the same set of tasks, which is common in many industrial control systems [12] [11]. We do this in order to be able to consider only the master and the first slave nodes, while simply calculating the phase combinations for the other slave nodes n ($2 < n \leq N$) by adding $(n-2) \times D_f + D_r(s)$ to the best phase combination for the first slave node. Moreover, we only consider phases that satisfy $\Phi_1 + D_{comm}(2) \leq \Phi_2$, and do not specifically consider the phases for tasks with lower priority than the communication task. We also deal with phases with identical relative phase combinations, such as $\{1, 2\}$ and $\{1 + \Delta', 2 + \Delta'\}$, as the same phase combinations; thus, we can omit the latter. The granularity of the task phase is also configurable to reduce analysis time, which is discussed in Sec. 6.3.

More importantly, we need to deal with the variable execution times of tasks. If we consider every case of variable execution times of all tasks, we need to examine $\left((C_1^{max} - C_1^{min}) \times H/T_1 \times (C_{m1}^{max} - C_{m1}^{min}) \times H/T_{m1} \times \dots \times (C_{mj}^{max} - C_{mj}^{min}) \times H/T_{mj} \right) \times \left((C_2^{max} - C_2^{min}) \times H/T_2 \times (C_{s1}^{max} - C_{s1}^{min}) \times \right.$

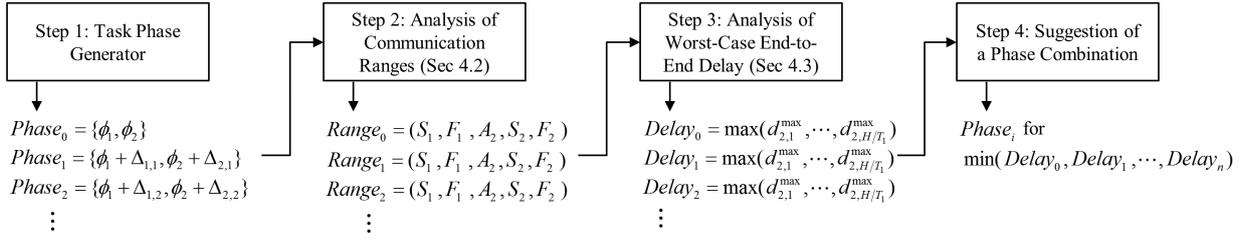


Figure 2: Steps to find optimal phase combination

$H/T_{s1} \times \dots \times (C_{sk}^{max} - C_{sk}^{min}) \times H/T_{sk}$) cases for each phase combination, where j and k are the number of non-communicating tasks that have a higher priority than τ_1 and τ_2 on the master and the slave nodes, respectively. We cannot simply reduce the number of these cases by assuming a larger resolution of execution times, because we can omit the case that generates the worst-case end-to-end delay with a large resolution. In order to reduce analysis time while considering variable execution times, the second step of our algorithm analyzes the communication range, $Range_i$, for each $Phase_i$ by only using the minimum and maximum execution times of tasks. A communication range represents a time interval where actual communication occurs between τ_1 and τ_2 . We describe the second step in finding communication ranges in Sec. 4.2.

In the third step, we efficiently calculate the maximum end-to-end delay $d_{2,i}^{max}$ for every message for a given phase. In Sec. 4.3, we detail the third step to analyze the maximum end-to-end delay for a given communication range. Once we determine the maximum end-to-end delay for each phase combination, we choose the phase combination that provides the minimum worst-case end-to-end delay in the last step.

4.2 Analysis of communication ranges

As we have mentioned in Sec. 4.1, in order to analyze the worst-case end-to-end delay for a given phase combination, we deal with the ranges of communication time points instead of considering all possible combinations of the execution times of tasks. We find the communication ranges by using the minimum and maximum execution times of tasks. First, we calculate the start and finish time ranges for all jobs of τ_1 on the master node, which are represented by $S_1 = \{S'_{1,1}, \dots, S'_{1,H/T_1}\}$ and $F_1 = \{F'_{1,1}, \dots, F'_{1,H/T_1}\}$, respectively, where $S'_{n,i} = \{s_{n,i}^{min}, s_{n,i}^{max}\}$ and $F'_{n,i} = \{f_{n,i}^{min}, f_{n,i}^{max}\}$. It should be noted that $S'_{n,i}$ and $F'_{n,i}$ have only minimum and maximum values, whereas $S_{n,i}$ and $F_{n,i}$ defined in Sec. 3.1 include all possible time points. The time range information F_1 is then translated into the range information of message arrival time on the first slave node, which is denoted by $A_2 = \{A'_{2,1}, \dots, A'_{2,H/T_1}\}$, where $A'_{2,i} = \{a_{2,i}^{min}, a_{2,i}^{max}\}$. The start and finish time ranges on the first slave node represented by $S_2 = \{S'_{2,1}, \dots, S'_{2,H/T_2}\}$ and $F_2 = \{F'_{2,1}, \dots, F'_{2,H/T_2}\}$ are calculated as well.

For example, Fig. 3 shows the communication range for the first message. For each node in the figure, the upper row represents time flow with the maximum execution times of tasks while the lower one shows time flow with the minimum execution times. In the master node of this example, τ_1 is preempted by a higher-priority task as shown in the upper row, which results in a larger $s_{1,1}^{max}$ than $s_{1,1}^{min}$. The difference between $f_{1,1}^{min}$ and $f_{1,1}^{max}$ becomes even greater due to the jitter in execution time (i.e., $C_1^{max} - C_1^{min}$). We can calculate $a_{2,1}^{min}$ and $a_{2,1}^{max}$ by adding $D_{comm}(2)$ to $f_{1,1}^{min}$ and $f_{1,1}^{max}$, respectively. In the slave node, two non-communicating tasks have a higher priority than τ_2 . Although the figure shows that $\tau_{2,1}$ is not preempted by these higher-priority tasks in the case of minimum execution time (i.e., the lower row), $\tau_{2,1}$ is delayed by a higher-

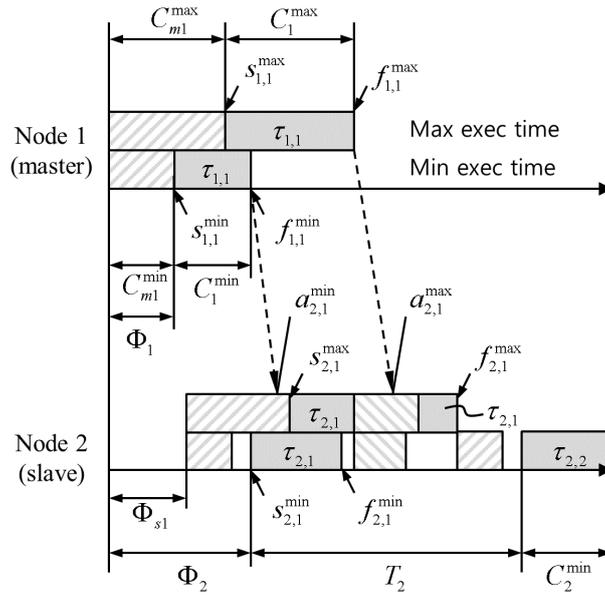


Figure 3: Analysis of communication ranges

priority task at the start point and preempted by the other one during execution in the case of maximum execution time. Consequently, for the given task set and phase combination, the range of message arrival, $[a_{2,1}^{min}, a_{2,1}^{max}]$, and that of the start time of $\tau_{2,1}$, $[s_{2,1}^{min}, s_{2,1}^{max}]$, overlap. By providing such information for the third step, we can efficiently analyze the worst-case end-to-end delay $d_{2,1}^{max}$.

4.3 Analysis of worst-case end-to-end delay

By using the feasible time ranges of communication provided by the second step, we analyze the worst-case end-to-end delay for a given task phase. As we have mentioned in Sec. 3.3, to analyze the worst-case end-to-end delay we need to calculate $D_{e2e}(2, i)$ for $1 \leq i \leq H/T_1$ and find $\max(d_{2,1}^{max}, \dots, d_{2,H/T_1}^{max})$. Since $D_{e2e}(2, i) = D_{e2e}^0(2, i) \cup D_{e2e}^{+1}(2, i)$ by Eq. 6, we describe how to calculate $\max(D_{e2e}^0(2, i))$ and $\max(D_{e2e}^{+1}(2, i))$ in this section, respectively.

As defined by Eq. 2 and Eq. 3 in Sec. 3.3, $D_{e2e}^0(2, i)$ is a set of end-to-end delays calculated supposing that the i^{th} receiver job $\tau_{2,i}$ consumes the i^{th} message. However, since we have only range information represented by minimum and maximum values, we modify Eq. 2 as follows:

$$f_{1,i}^{min} + D_{comm}(2) = a_{2,i}^{min} \leq s_{2,i}^{max}, \quad (8)$$

which is satisfied if there is a possibility that the i^{th} receiver job receives the i^{th} message. In this case, we can calculate the maximum end-to-end delay of the i^{th} message as follows:

$$\begin{aligned} \max(D_{e2e}^0(2, i)) &= f_{2,i}^{max} - \min\left(\mathcal{R}^{-1}(a_{2,i}^{min} - D_{comm}(2))\right) \\ &= f_{2,i}^{max} - \min\left(\mathcal{R}^{-1}(f_{1,i}^{min})\right) \\ &= f_{2,i}^{max} - s_{1,i}^{min}, \end{aligned} \quad (9)$$

which is derived from Eq. 3. In order to calculate the end-to-end delay, we need to know the start time of the sender job (i.e., $s_{1,i}$) and the finish time of the receiver job (i.e., $f_{2,i}$). It is obvious that the end-to-end delay becomes maximum if the sender sends the message in earliest

time (i.e., starts at $s_{1,i}^{min}$) while the receiver is delayed for as long as possible (i.e., finishes at $f_{2,i}^{max}$). In Eq. 9, the start time of the sender job is reversely calculated from the message arrival time $a_{2,i}^{min}$.

On the other hand, if there is a chance that the message arrives after the beginning of the i^{th} receiver job, the message happens to be handled by the next (i.e., $(i+1)^{th}$) job. This occurs when the following equation derived from Eq. 4 is satisfied:

$$f_{1,i}^{max} + D_{comm}(2) = a_{2,i}^{max} > s_{2,i}^{min}. \quad (10)$$

Then the equation for finding the maximum end-to-end delay of the i^{th} message can be derived from Eq. 5 as follows:

$$\max(D_{e2e}^{+1}(2, i)) = f_{2,i+1}^{max} - \min(\mathcal{R}^{-1}(a_{2,i}^* - D_{comm}(2))). \quad (11)$$

Here $a_{2,i}^*$ is the earliest arrival time of the message processed by $\tau_{2,i+1}$. That is,

$$a_{n,i}^* = \max(s_{n,i}^{min} + \epsilon, a_{n,i}^{min}), \quad (12)$$

where ϵ represents a very small value. If the message is arrived at $s_{2,i}^{min} + \epsilon$ while the receiver job starts at $s_{2,i}^{min}$, the message is processed by the next receiver job. However, if $s_{n,i}^{min} + \epsilon < a_{n,i}^{min}$, then $a_{n,i}^{min}$ is the earliest time point. It should be noted that $\min(\mathcal{R}^{-1}(a_{2,i}^* - D_{comm}(2)))$ in Eq. 11 can be neither $s_{1,i}^{min}$ nor $s_{1,i}^{max}$. However, we have only minimum and maximum values that present range information to reduce analysis time. Thus, it is difficult to find the exact $\min(\mathcal{R}^{-1}(f_{1,i}))$ for arbitrary $f_{1,i}$ (i.e., $a_{2,i}^* - D_{comm}(2)$), unless we analyze every preemption point of the sender job by considering variable execution times. Although we could calculate a converged value by using a recurrence equation for each message, to reduce the time needed for analysis, we pessimistically assume that the worst-case preemption always occurs and estimate $\min(\mathcal{R}^{-1}(f_{n,i}))$ as follows:

$$\min(\mathcal{R}^{-1}(f_{n,i})) = \begin{cases} f_{n,i} - C_{n,i}^{max} - P_r^{max}, & \text{if } f_{n,i} - C_{n,i}^{max} - P_r^{max} > s_{n,i}^{min}, \\ s_{n,i}^{min}, & \text{otherwise,} \end{cases} \quad (13)$$

where P_r^{max} is the worst-case preemption time of $\tau_{n,i}$. As we calculate $s_{n,i}^{max}$ and $f_{n,i}^{max}$ assuming the maximum execution time for both communicating task and non-communicating higher-priority tasks, the time range $[s_{n,i}^{max}, f_{n,i}^{max}]$ includes the worst-case preemption time. Hence, P_r^{max} can be expressed as:

$$P_r^{max} = f_{n,i}^{max} - s_{n,i}^{max} - C_{n,i}^{max}. \quad (14)$$

By using Eq. 14, we restate Eq. 13 as follows:

$$\min(\mathcal{R}^{-1}(f_{n,i})) = \begin{cases} f_{n,i} - f_{n,i}^{max} + s_{n,i}^{max}, & \text{if } f_{n,i} - f_{n,i}^{max} + s_{n,i}^{max} > s_{n,i}^{min}, \\ s_{n,i}^{min}, & \text{otherwise,} \end{cases} \quad (15)$$

Therefore, by Eq. 9 and Eq. 11, we can find the worst-case end-to-end delay of i^{th} message as follows:

$$\begin{aligned} d_{2,i}^{max} &= \max(D_{e2e}(2, i)) \\ &= \max(\{ \max(D_{e2e}^0(2, i)), \max(D_{e2e}^{+1}(2, i)) \}). \end{aligned} \quad (16)$$

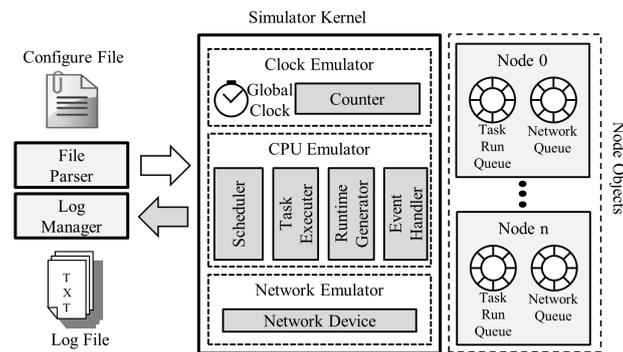


Figure 4: Simulation tool

5 Simulation tool

In real systems, end-to-end delays vary due to jitters in the execution times of tasks. Thus, it is not feasible to evaluate the distribution and the average of end-to-end delays by using a simple mathematical equation. Instead, we implemented a simulation tool consisting of four components: configuration manager, node objects, simulation kernel, and log manager. Fig. 4 shows the overall design of the proposed simulator. The analysis tool is based on discrete-event simulation to simulate task scheduling, message transmission/forwarding, DMA, and I/O event handling.

Configuration parameters such as the number of nodes, task sets, network bandwidth, message size, and event handling mode are defined in the configuration file in XML format. Thus, a user of the tool can easily change the attributes of the system to be analyzed. Furthermore, newly defined attributes can be described simply by extending the XML schema. The configuration module reads the configuration file at the initialization phase and sets the environment variables of the simulation kernel with values extracted from the configuration file. The configuration module also creates node objects as much as described in the configuration file and sets attributes of these objects, such as event handling mode.

The node objects contains a snapshot of a run-time image of each node, such as the state information of the run queue and the message queue. The run queue stores task control blocks, which saves a given task's mission as well as the amount of time for which it executes for the given period. The mission is classified into three categories: send, receive, and other. The message queue stores network messages, which are yet to be sent to the network or yet to be consumed by a receiver task.

The simulation kernel performs the analysis loops. The simulation kernel consists of a clock emulator, a CPU emulator, and a network emulator. The clock emulator simulates a synchronized global clock and increases its counter for every simulation loop.

The CPU emulator run task(s) on each node for every time unit. The scheduler selects a task to run based on the fixed-priority scheduling algorithm. Our current implementation only supports the RM scheduling algorithm but our design is general enough to add other scheduling algorithms. The task executor emulates the behavior of the selected task based on its mission information. If the mission is *send*, the task executor posts a send request to the network device at the finish time of the job. On the contrary, in the *receive* case, the task executor consumes a message from the message queue at the start time of the job. The task executor simply consumes the time for tasks dedicated to *other* missions. The runtime generator generates the execution time of a job according to C_n^{min} and C_n^{max} for every task. In the current implementation, we

Table 1: Task set on slave nodes

Task Name	T_i and C_i	Description
MotorAct	$250\mu s$, $25\sim 35\mu s$	Controls a motor (highest priority)
RtMsg	$250\mu s$, $10\sim 15\mu s$	Receives real-time messages from the network, and shares it with the MotorAct task
NrtMsg	$250\mu s$, $7\sim 10\mu s$	Handles non-real-time messages (lower priority than RtMsg)
HealthMon	$500\mu s$, $6\sim 9\mu s$	Performs health monitoring (lowest priority)

Table 2: Simulation parameters

Parameters	Value	Parameters	Value
Network bandwidth	100 Mbps	Network forwarding delay (D_f)	$1\ \mu s$
Event handling mode	Interrupt	Interrupt handling time	$5\ \mu s$
Packet size	50 bytes	DMA overhead for a 50byte packet	$1\ \mu s$

support uniform distribution and normal distribution for the type of execution time distribution but other distribution type can also be added.

The network emulator simulates operations of the network device (e.g., message transmission, DMA, and raising an interrupt). Event handling can be carried out in either polling or interrupt mode. If the network device is configured as a polling-based device, the event is handled when a receiver task is released. Thus, in this mode, the network emulator silently inserts a received message into the receive queue without any notification when it arrives. When the network device is configured as an interrupt-based device, the network emulator sends an interrupt signal to the CPU emulator so that the event handler is invoked as soon as a message arrives preempting a running task. The network emulator also decides the message transmission speed based on the bandwidth information, and maintains the remaining bytes of a message being sent on each object node.

The log manager stores the results of analysis in output files. The final output represents the total running time of the tool, the end-to-end delay for each message, and the average end-to-end delay. The log manager also saves intermediate information generated during analysis.

6 Industrial case study

In this section, we detail an industrial case to show that our algorithm can efficiently propose enhanced task phasing across distributed nodes for a given task set.

6.1 Analysis environments

Table 1 shows the task set we ran on the slave nodes. We used the task set defined in Kim et al. [12] for the motor drive. Each slave node scheduled these tasks using the fixed-priority scheduling algorithm. We assume that the execution times of tasks followed a uniform distribution, which was not necessary for our analysis. Table 2 shows the configuration parameters for the simulation tool. We borrowed such network parameter values as queue size, bandwidth, and forwarding delay from EtherCAT specifications.

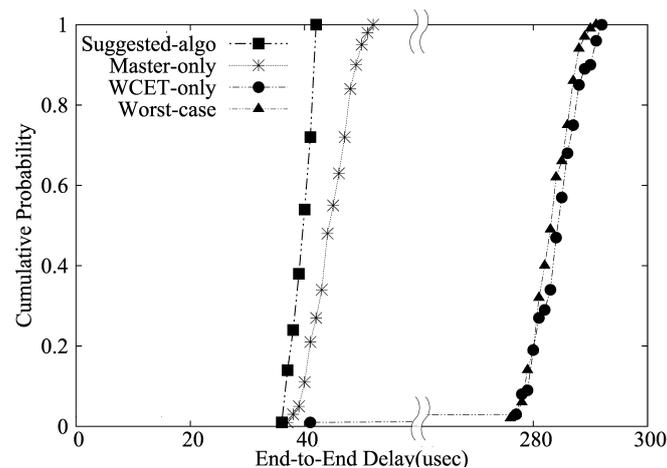


Figure 5: Distribution of end-to-end delays of different phasing schemes

6.2 Comparisons with other phasing schemes

In this section, we compare end-to-end delays of four phase combinations. One of these phase combinations was generated by our algorithm in Sec. 4 (*Suggested-algo*). Another phase combination was limited to the master node by assuming the critical instant on the slave (*Master-only*). The third was generated based on the worst-case execution times (WCET) of tasks without considering the jitters in execution time (*WCET-only*). The other phase combination represented worst-case phasing with respect to end-to-end delay (*Worst-case*). By utilizing the simulator described in Sec. 5, we measured the end-to-end delays, and present the cumulative probability of these and the minimum, average, and maximum end-to-end delays.

The distribution of end-to-end delays of four phase combinations are compared in Fig. 5. It can be seen that the end-to-end delays of *Suggested-algo* were distributed among the lowest values, whereas those of *Worst-case* yielded the highest values. Although *WCET-only* adjusted the phases of tasks, it was largely unable to enhance end-to-end delay in comparison with *Worst-case*, because *WCET-only* did not consider variable execution times of tasks. It is also evident that *Master-only* significantly reduced end-to-end delays, but still generated higher values than *Suggested-algo* due to the absence of phasing on the slave node.

Fig. 6 compares the minimum, average and maximum end-to-end delays of the four phase combinations. Compared with *Master-only* and *WCET-only*, *Suggested-algo* yielded end-to-end delays that were shorter by 19% and 86%, respectively. It also reduced jitter by 67% and 98%, respectively. In the case of *WCET-only*, the minimum end-to-end delay was significantly lower than its average and maximum values. This was because *WCET-only* shows the lowest end-to-end delay when tasks consume their WCET, although its probability is very low, as shown in Fig. 5.

6.3 Impact of phase granularity

As mentioned in Sec. 4.1, we can change the granularity of the task phase to reduce the number of phase combinations to be considered. A higher granularity can reduce search time but harms optimality. To analyze the impact of granularity on search time and optimality, we ran our algorithm by varying granularity and analyzed the worst-case end-to-end delays of the suggested phase combinations.

Fig. 7 shows that the number of search cases can be significantly reduced with a higher phasing granularity. However, Fig. 8 shows that the worst-case end-to-end delays increased as

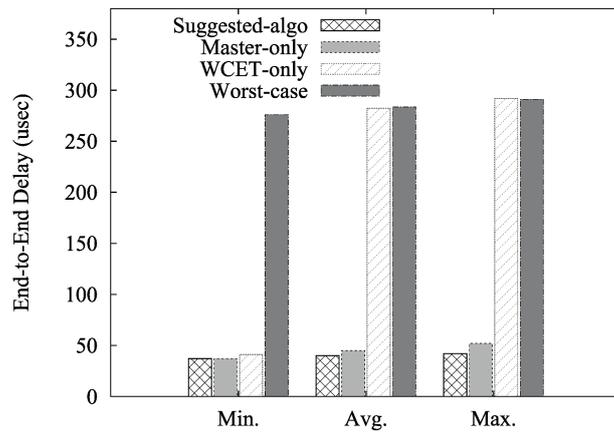


Figure 6: Minimum, average, and maximum end-to-end delays of different phasing schemes

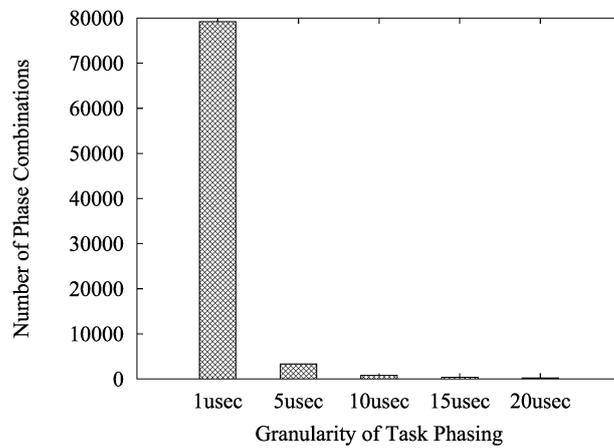


Figure 7: Number of phase combinations with different phase granularities

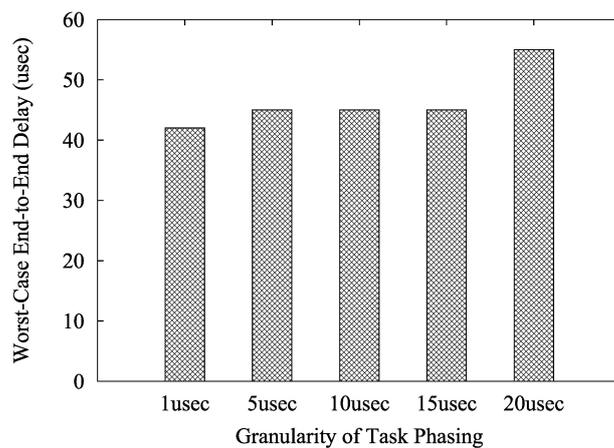


Figure 8: Worst-case end-to-end delay with different phase granularities

granularity increased, which means that the chances of finding an optimal phase combination also decreased with a higher granularity.

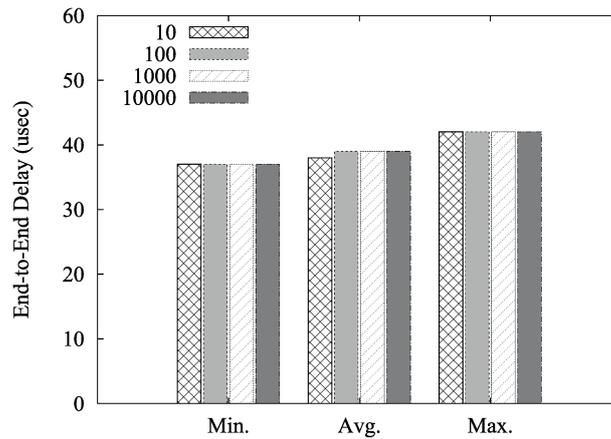


Figure 9: Impact of the number of simulated periods on the analysis accuracy

6.4 Accuracy of delay distribution

To analyze the distribution of end-to-end delays, our simulator varied the execution times of tasks for every period during the simulation. Therefore, the simulator could provide an accurate distribution when a sufficient number of periods had been simulated. Thus, the larger the number of periods simulated, the more accurate the distribution of end-to-end delays. In order to analyze the impact of the number of simulated periods on the accuracy of performance measurement, we ran our simulator by varying the number of simulated communication periods from 10 to 10,000.

Fig. 9 shows the minimum, average and maximum end-to-end delays analyzed with different numbers of periods. We can see that the average end-to-end delay of the 10-periods case was slightly lower than in other cases, while the minimum and maximum end-to-end delays were the same for all cases. In general, if variations in tasks' execution time and the number of tasks are large, a small number of simulated periods may produce a less accurate distribution of end-to-end delays. However, we would not need an excessively large number of periods, as shown in our results.

7 Conclusions

In this study, we proposed a heuristic algorithm that searched for a near-optimal task phase combination on distributed nodes with respect to the minimal worst-case end-to-end delay for emerging real-time fieldbuses, such as EtherCAT. To reduce search time, the algorithm identifies time intervals where communication occurs and efficiently calculates the worst-case end-to-end delay for a given task phase combination without considering all possible execution times of tasks and their combinations. To analyze the distribution of end-to-end delays of different phasing approaches, we also implemented a simulation tool that considered variable execution times of tasks and the behavior of a real-time fieldbus.

We carried out an industrial case study to show that the proposed heuristic can efficiently suggest near-optimal task phasing across distributed nodes for minimum worst-case end-to-end delays. The simulation results clearly showed that our algorithm can reduce end-to-end delay and its jitter by 86% and 98%, respectively, compared with other task phasing approaches. In summary, we need to consider variable execution times of tasks for optimal phasing of distributed tasks, but also need to apply task phasing to both master and slave nodes to achieve minimum worst-case end-to-end delay.

Acknowledgment

This work was supported by the Dual Use Technology Program (UM13018RD1).

Bibliography

- [1] Bril R.J., Steffens E.F.M., Verhaegh W.F.J. (2004); Best-case response times and jitter analysis of real-time tasks, *Journal of Scheduling*, 7(2), 133-147, 2004.
- [2] Cena G., Bertolotti I.C., Scanzio S., Valenzano A., Zunino C. (2012); Evaluation of EtherCAT distributed clock performance, *IEEE Trans. Industrial Informatics*, 8(1), 20-29, 2012.
- [3] Cena G., Bertolotti I.C., Scanzio S., Valenzano A., Zunino C. (2010); On the accuracy of the distributed clock mechanism in EtherCAT, In *Proc. the 8th IEEE Int. Workshop on Factory Communication Systems*, 43-52, 2010.
- [4] Craciunas S.S., Oliver R.S., Ecker V. (2014); Optimal static scheduling of real-time tasks on distributed time-triggered networked systems, *Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation*, 1-8, 2014.
- [5] Farsi M., Ratcliff K., Barbosa M. (1999); An introduction to CANopen, *Computing & Control Engineering Journal*, 10(4), 161-168, 1999.
- [6] Garner G.M., Gelter A., Teener M.J. (2009); New simulation and test results for IEEE 802.1 as timing performance, *Int. Symp. on Precision Clock Synchronization for Measurement, Control and Communication*, 1-7, 2009.
- [7] Harbour M.G., Gutiérrez J.J., Drake J.M., Martínez P.L., Palencia J.C. (2013); Modeling distributed real-time systems with MAST2, *Journal of Systems Architecture*, 59(6), 331-340, 2013.
- [8] Henia R., Racu R., Ernst R. (2007); Improved Output Jitter Calculation for Compositional Performance Analysis of Distributed Systems, *Proc. IEEE Int. Parallel and Distributed Processing Symp.*, 1-8, 2007. doi: 10.1109/IPDPS.2007.370356
- [9] International Electrotechnical Commission (2007); Industrial Communication Networks Fieldbus specifications - Part 3-12: Data-Link Layer Service Definition - Part 4-12: Data-link layer protocol specification-Type 12 elements, *IEC*, 61158-3/4-12 (Ed. 1.0), 2007.
- [10] Kang H., Kim K., Jin H.-W. (2016); Real-Time Software Pipelining for Multidomain Motion Controllers, *IEEE Trans. Industrial Informatics*, 12(2), 705-715, 2016.
- [11] Kim I.; Kim T. (2015); Guaranteeing isochronous control of networked motion control systems using phase offset adjustment, *Sensors*, 15(6), 13945-13965, 2015.
- [12] Kim K., Sung M., Jin H.-W. (2012); Design and implementation of a delay-guaranteed motor drive for precision motion control, *IEEE Trans. Industrial Informatics*, 8(2), 351-365, 2012.
- [13] Liu C., Layland J. (1973); Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, *Journal of the ACM*, 20(1), 46-61, 1973.

- [14] Potra S., Sebestyen G. (2006); EtherCAT protocol implementation issues on an embedded linux platform, *Proc. IEEE Int. Conf. on Automation, Quality and Testing, Robotics*, 420-425, 2006.
- [15] Prytz G. (2008); A performance analysis of EtherCAT and PROFINET IRT, *Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation*, 408-415, 2008.
- [16] Redell O., Sanfridson M. (2002); Exact Best-Case Response Time Analysis of Fixed Priority Scheduled Tasks, *Proc. Euromicro Conf. on Real-Time Systems*, 165-172, 2002.
- [17] Steinbach T., Kenfack H.D., Korf F., Schmidt T.C. (2011); An extension of the OMNeT++ INET framework for simulating real-time Ethernet with high accuracy, *Proc. the 4th Int. ICST Conf. on Simulation Tools and Techniques*, 375-382, 2011.
- [18] Sung M., Kim I., Kim T. (2013); Toward a Holistic Delay Analysis of EtherCAT Synchronized Control Processes, *International Journal of Computers Communications & Control*, 8(4), 608-621, 2013.
- [19] Tindell K., Clark J. (1994); Holistic schedulability analysis for distributed hard real-time systems, *Microprocessing and Microprogramming*, 40(2), 117-134, 1994.
- [20] Valls M.G., Alonso A., de la Puente J.A. (2012); A dual-band priority assignment algorithm for dynamic QoS resource management, *Future Generation Computer Systems*, 28(6), 902-912, 2012.
- [21] Valls M.G., López I.R., Villar L.F. (2013); iLAND: An enhanced middleware for real-time reconfiguration of service oriented distributed real-time systems, *IEEE Trans. Industrial Informatics*, 9(1), 228-236, 2013.
- [22] Zeng H., Di Natale M., Giusto P., Sangiovanni-Vincentelli A. (2009); Stochastic analysis of CAN-based real-time automotive systems, *IEEE Trans. Industrial Informatics*, 5(4), 388-401, 2009.

Mining Users' Preference Similarities in E-commerce Systems Based on Webpage Navigation Logs

P. Li, C.X. Wu, S.Z. Zhang, X.W. Yu, H.D. Zhong

Ping Li

College of Biological and Environmental Sciences,
Zhejiang Wanli University
No. 8 South Qianhu Rd., Ningbo, Zhejiang, 315100,
P. R. China, liping_kaixin@163.com

Chunxue Wu

School of Optical-Electrical and Computer Engineering,
University of Shanghai for Science and Technology
No. 516 Jun Gong Road, Shanghai 200093,
P. R. China, tyfond@126.com

Shaoyong Zhang

School of Electronic and Computer Science,
Zhejiang Wanli University
No. 8 South Qianhu Rd., Ningbo, Zhejiang, 315100,
P. R. China, dlut_z88@163.com

Xinwu Yu

The Information Center,
Zhejiang Wanli University
No. 8 South Qianhu Rd., Ningbo, Zhejiang, 315100,
P. R. China, herrison@163.com

Haidong Zhong*

Logistics and E-commerce School,
Zhejiang Wanli University
No. 8 South Qianhu Rd., Ningbo, Zhejiang, 315100,
P. R. China

*Corresponding author: zhonghaidong@zww.edu.cn

Abstract: Mining users' preference patterns in e-commerce systems is a fertile area for a great many application directions, such as shopping intention analysis, prediction and personalized recommendation. The web page navigation logs contain much potentially useful information, and provide opportunities for understanding the correlation between users' browsing patterns and what they want to buy. In this article, we propose a web browsing history mining based user preference discovery method for e-commerce systems. First of all, a user-browsing-history-hierarchical-presentation-graph to established to model the web browsing histories of an individual in common e-commerce systems, and secondly an interested web page detection algorithm is designed to extract users' preference. Finally, a new method called UPSAWBH (User Preference Similarity Calculation Algorithm Based on Web Browsing History), which measure the level of users' preference similarity on the basis of their web page click patterns, is put forward. In the proposed UPSAWBH, we take two factors into account: 1) the number of shared web page click sequence, and 2) the property of the clicked web page that reflects users' shopping preference in e-commerce systems. We conduct experiments on real dataset, which is extracted from the server of our self-developed e-commerce system. The results indicate a good effectiveness of the proposed approach.

Keywords: web browsing history mining, e-commerce, preference, recommendation.

1 Introduction

E-commerce (or electronic commerce) usually refers to products or services exchange oriented activities based on the Internet technology, which covers online payment, information security, logistics distribution, etc [25, 29].

Pushed by the widespread availability of the Internet, more and more consumers prefer to shift from traditional face-to-face transactions to web-based commercial activities [16]. Meanwhile, with the advances of network technology and fast-growing e-commerce systems, such as Amazon, Stimulated by network technology advances, the rapid growth of networking systems and the boom in netizens, an ever-increasing number of traders and entrepreneurs have participated in e-commerce [19].

Nowadays, supply chain management, online transaction processing and many other e-commerce relevant industries have attracted thousands of workers and companies to provide a great many products or services for the online transactions. With online information growing exponentially, it will eventually result in "Information Overload" and "Information Loss" in e-commerce systems [11], which seriously hinder the development of e-business or e-commerce industries.

Since the born of collaborative filtering approach in the 1990s, recommendation systems have become an intensively investigated independent discipline and deemed as an effective means to ease the "Information Overload" problem. Generally, the existing recommendation approach can be divided into three categories [2, 26, 28]: (1) collaborative filtering recommendation, which is based on the idea of similarity of users to make predictions (filtering) about the interests of an individual by calculating preferences from other users (collaborating) who have brought the same items as the target user; (2) content-based recommendation, which is the technology of choosing items as recommended for a target user according to other similar users' preferred items; (3) hybrid recommendation, which is proposed to combine different recommendation technologies according to different mixed strategies (e.g., weighted, switch, mixed characteristics, combination, series, meta level hybrid, etc.). Although, the specific steps of these recommendation methods vary, the fundamental principles of them are similar: finding users with similar preferences with target consumer and using them to make recommendations.

However, there are still many challenges, such as data scarcity and cold-start [20, 21], in utilizing these recommendation approaches. Searching for candidate users more precisely is deemed as an important solution to improve these challenges. In the majority of the existing researches, insufficient effort has been made to solve the problem that users' preference is time-varying and can be measured in different granularity.

From the view point of Srivastava, web usage logs, which is an important part of web data, contain abundant information and can be exploited in many Web personalization applications [24]. They provide demographic data (for instance name, age, country, marital status, education, interests, etc.) of each website user, also implicit knowledge about users' behavior patterns and other preferences. To discover the information explicitly from a novel perspective, a web page navigation logs mining based method is proposed to extract users' preferences dynamically.

The paper directs toward excavating users' browsing histories in typical e-commerce systems and tries to establish a hierarchical presentation model for the data, which will have wide utilization potentiality in e-commerce system intelligence, monitoring users' preferences and analyzing on different levels. Based on the proposed hierarchical presentation model and interested web page detection algorithm, a user's shopping preference measure algorithm UPSAWBH is put forward.

2 Related work

2.1 Web page navigation logs mining

Web page navigation logs generally refers to the detailed information that can be gathered from the Internet browsing of users, which are lists of links network clients clicked and the elapsed time between them [22,30]. They can be represented as a quintuple $WebLog = \{time, remote\ host, method, page, request\ status\}$.

The format of the web browsing log may vary slightly in different application servers, but the elements listed in the quintuple are essential. The meaning of each element in the quintuple is explained as follows [32]:

"*time*" denotes the time the server responds to the user's request and returns the requested resources. In the article, time between two URLs is the time interval between the request of a URL and the followed URL page, which contains of web page load time and page time as shown in Fig.1 [1].

"*remote host*" denotes the logic name or IP address of the Network server that a user visits. A proxy server may exists between the user and the Web server, so "*remote host*" may represent the final proxy server that the user has visited.

"*method*" denotes the request method, which include GET, POST, HEADER, OPTIONS, PUT, and so on of the user. Among these methods, GET and POST are the most usually adopted.

"*page*" denotes the requested web page. The "page" can be functionally divided into two types: navigation pages and content pages. Navigation pages act as "guiding people" in the Internet, while the content page is the place where people usually spend most of their time.

"*request status*" denotes the status code that request the user to return to the server. The status code consists of three digits, which represent the status response of the server to the browser's request.

Web page navigation log data contains a lot of valuable information, such as the records of links that a user has visited and the elapsed time between them, the number of clicks on each web page, the complete visit path and the time spent on the web pages. This kind of statistical information can be explored by a variety of methods and used for many scenarios, such as users' preference prediction and the prefetching of pages to improve users' browsing experience.

A great many scholars have paid the much attention on website access pattern investigation of users by their browsing logs with the help of statistical analysis methods to reduce server-side response time and improve access efficiency of web pages [27]. Magdalini Eirinaki and Michalis Vazirgiannis rely on the application of statistical analysis and intelligent data mining methods (for instance, clustering, association rule mining, sequential pattern discovery and classification) to the Web log data, resulting in a set of valuable patterns that imply individuals' access patterns, and the knowledge is then employed to personalize pages for users according to their navigational behavior and profile [9]. Based on the theory of probability, Borges and Levene put forward a data mining method that captures users' web page access patterns: individuals' navigation sessions are treated as hypertext probabilistic grammar whose higher probability strings correspond to the interested tails of an individual and the last N visited web pages affect the affect the probability of the following page to be navigated [4]. Ezeife and Lu proposed a Web access pattern tree (WAP-tree) approach to explore frequent visit sequences for users, which can response dynamically without numerous re-constructions of WAP-tree during knowledge mining [10]. To overcome the weakness of ineffective content management of websites and the incapability of providing personalized web page services for the users in traditional web usage mining approaches, Yao-Te Wanga and Anthony J. T. Lee introduced the concept of throughout-surfing patterns and present an advanced access pattern mining model [27]. Also, they put forward a compact graph

model, termed a web page navigation path traversal graph, to store knowledge about the web page access paths of the website users.

2.2 User similarity measurement

In website access scenario, a user profile contains static part, which changes seldom (such as demographic information) and dynamic part that changes frequently. The ability to find users with similar preference or distinguish between different individuals is a matter of cardinal significance in various information system applications, especially in e-commerce systems. The process of finding similar users is usually conducted based on users' profile mining, which focus on knowledge about web page access preferences and characteristics of the users. Due to the convenience of collecting users' web page navigation and other potential valuable information from server-side, users' profile exploration has attracted much attention of scholars all over the world in recent years. Personalized recommendation in e-commerce systems is deemed as one of the most popular applications that based on users' profile and preference extraction.

User similarity measurement is one of the research aspects of users' preference mining. Although, much effort has been paid on this topic, it is still a problem-rich area. The existing researches focus mainly on how to measure user similarity in different circumstance. Some investigations follow the perspective of geography to probe users' similarity. For example, Li, Zheng, Xie, et al. found it important to discover valuable knowledge from large scale spatio-temporal data, and proposed hierarchical-graph-based similarity measurement (HGSM) framework to model an individual's trajectories. The model considered both the sequence property of people's movement behaviors and hierarchy attribute of geographic feature, which proved to be an effective way of measuring similarities among users [17]. Guy, Jacovi, Perer, et al. studied nine kind of sources (friending, communities, blogs, forums, et al.) that can be used for users' similarity measurement in social media applications [13]. Their research shows that the aggregation of sources may be valuable to measure the similarity between people. All these approaches are based on the hypothesis that the more two users share the same geographical overlap areas, the more likely they can be similar to each other. However, it may be challenge to evaluate the similarity of two users who are living very close. Take two persons, an old man and a young one, living in the same community as an example: they may share the same geographical overlap area (both stay at home) in a period of leisure time (week end or holiday), but it is hard to say that they are similar. While still other works pay much attention to semantic analysis to exploit users' similarity. Ying, Lu, Lee, et al. argued that geographically close uses' trajectories may not have to be similar, because the activities implied by nearby landmarks that they passed through may vary and so they put forward a MSTPS (Maximal Semantic Trajectory Pattern Similarity) method, which measures the similarity between users based on the calculation of semantic similarity of their trajectories [31]. Lee and Chung proposed a method to calculate user similarity according to the semantics of frequently visited locations and the user's potential preference [15]. Still others studies focus on mining users' purchase history or web page access record to measure their similarity. For example, Eckhardt A. put forward a collaborative filtering based user preference model to explore users' similarity [8]; Wei, Shijun, Yunlu, et al. held the opinion that users sharing similar purchase in history are very likely to have similar preference in the future, and constructed a user similarity network to get rid of the negative affect of popular objects or items for personalized recommendation [12].

In this work we focus on web usage logs mining to find users' preferences, based on the opinion that web page navigation patterns resemble users may have similar interests. We combine the approach of hierarchical-graph-based similarity measurement with Web usage mining techniques on web page access records.

2.3 E-commerce recommendation

With the boom in e-commerce in recent years, the structure of the e-commerce system has become much more complicated than ever before as it provides a great many customized services for both customers and enterprises. Meanwhile, a wide variety of goods is provided by sellers in e-commerce virtual shops, which makes it impossible for a user to view all the products when he/she has something to buy. Under this circumstance, the demand for understanding users' preferences in e-commerce systems and find useful knowledge to make recommendations has greatly increased. By providing precise and useful suggestions to a potential consumer, recommendations in e-commerce systems have made good profits for many popular e-commerce enterprise, such as Tabao.com and Amazon.com [18], who recommend new products related to the items purchased previously by similar users of the target user.

Since the collaborative filtering approach first proposed in the mid-1990s, scholars all over the world have devoted their effort on recommender system, and make it a high-profile research area [33]. The interaction between a recommendation system and the user can be divided into explicit approach, using users provided registration information to get a general picture of their static profiles, as well as implicit methods, by exploring the web page access records to infer users' preferences [14]. The former make recommendations based on the users' input conditions, while the latter would automatically collect or observe users' behavior to detect their profile. In e-commerce systems, recommendation seems extremely import because it aim sat customizing/personalizing a given product according to interests of the consumers and help them make decisions.

Although, recommendation methods (such as Collaborative filtering, Content-based filtering and rule-based filtering) may different from one another, the process of e-commerce recommendation generally involves three main steps [3,5-7]:

- (1) the users' buying records are collected, processed and analyzed to find their preferences;
- (2) based on the conclusion of the above first step, commence recommendation for a target user and
- (3) provide a recommended goods list for the target users to buy. In addition to mining customers' properties and filtering unnecessary information, an e-commerce recommendation system focuses, as far as possible, on the matter of the ability to suggest items of interest to the user.

Over the last decade many new algorithms and methods have been put forward to improve recommendation accuracy and efficiency in both practical application and theoretical research. However, it still faces many challenges, such as cold start data sparse. At present, recommender system related research is still a popular issue because it constitutes problem-rich research areas concerning not only about finding accurate recommendation algorithms, but also a great many crucial factors, such as diversity, recommender persistence, robustness, serendipity, privacy etc.

In many e-commerce systems, web page browsing pattern mining plays a very important role in generating accurate recommendations. When a user accesses to a web page or a browser-server based e-commerce system, the URLs of the pages visited will be stored in the server access log. It plays a crucial role in conveying knowledge of customers' activities and preferences, which are very useful for personalized recommendation. Furthermore, it dynamically reflects their interests in a sense and the similar click sequence of two different users may imply that they have similar preferences at that time, which is very import for real-time recommendations. Preference similarity is one of the most useful pieces of knowledge that can be extracted by many kind of data mining models [13]. This knowledge will help in finding groups of visitors with similar preferences and making effective recommendations. In recent years, many scholars have turned

their attention to web browsing history mining based e-commerce recommendation, for example, Qinbao Song and Martin Shepperd put forward a vector analysis and fuzzy set theory based model to explore similar users, frequently visited web pages and navigation paths and designed a web browsing mining based recommendation model for e-commerce systems [23].

3 Users' preference similarities exploration

In this section, we conduct a detailed introduction to the processes involved in user preferences extraction, which includes web browsing trajectory definition, user browsing history hierarchical presentation and users' preference similarity measure algorithm.

3.1 Preliminary

Definition 1. Web Browsing Trajectory. A web browsing trajectory ($WBTraj$) is a clicked URL sequence of pages viewed by a user across the entire web visit process: Each $WBTraj$ contains a page URL ($l_i.URL$) request time ($l_i.RTime$) and leave page time ($l_i.LTime$). Thus, a web browsing trajectory can be represented as $WBTraj = l_1 \rightarrow l_2 \rightarrow \dots \rightarrow l_{n-1} \rightarrow l_n$, where $l_i.RTime < l_i.LTime$ and $l_i.LTime < l_{i+1}.RTime$.

Definition 2. Interested Web Page. Generally, an interested web page ($IURL$) can be represented by a URL where a user stays on longer than a certain time interval. Therefore, the main factor involved in extraction of the interested web page depends on the time threshold (θ_t), which implies the time a user stays on a certain web page. Formally, a set of interested web pages can be defined as $IURL = \{l_i \in WBTraj, |l_i.LTime - l_i.RTime| \geq \theta_t\}$.

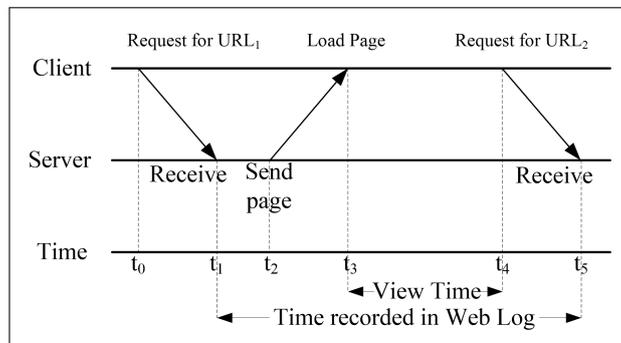


Figure 1: Time of request between two URLs

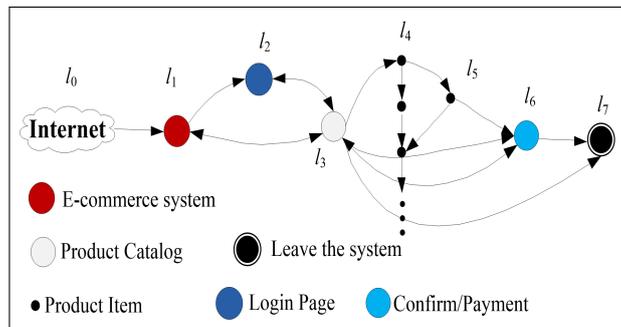


Figure 2: Web browsing trajectory of a general e-commerce system

As demonstrated in Figure 2, $l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow l_4 \rightarrow l_5 \rightarrow l_6$ formulates a general web browsing trajectory in a common e-commerce system, and the interested web page can be extracted according to the web page browsing time threshold.

Typically, interested web pages may occur in the following conditions: (1) a person enters an e-commerce system, opens a web page and then diverts his (or her) attention away, and (2) a user opens more than one product item web page to compare their characteristics and decide which one to buy, exceeds a time limit at a certain web page. It is impossible to grasp customers' interested produce web page navigation logs under the former circumstance. We focus on the second case, and put forward an Interested-Webpage-Detection algorithm. Detailed process of the algorithm can be described below.

Algorithm 1 Interested Webpage Detection

Require: Web browsing trajectory $WBTraj$ and a time threshold θ_t

Ensure: A set of URLs ($IURL=\{L\}$) of the interested web pages

```

1:  $i = 0, iUrlCount = |WBTraj|$  //The number of WebUrl in WBTraj
2: while  $i < iUrlCount$  do
3:   if  $|l_i.LTime - l_i.RTime| \geq \theta_t$  then  $IURL.insert(l_i)$ 
4:   end if
5:    $i++$ 
6: end while
7: return  $IURL$ 

```

To capture interested web pages in a server-browser based e-commerce system as accurate as possible, and we need to find a suitable time threshold to detect every stay on a certain web page. A too small time threshold can lead to too many navigated web pages over-detected as IURLs. A small time threshold value, for example 1 second, might be more capable of identifying much IURLs for an e-commerce system; however, this could cause too many IURLs detected, making us get lost and don't know which is the real web page or produce that users interested. In addition, time interval between the server response and the requested web page shown in the user's screen may be greater than 1 second (depending on network situation). This is obviously not in accordance with people's intuitiveness, as the web page has not shown in that short time interval. Meanwhile, too large time threshold (θ_t) value is not appropriate either. It could result in many interested web pages, which indicate users' real preference, cannot be detected efficiently.

3.2 User browsing history hierarchical presentation

From the above section, we can ascertain that the more interested web pages two people share in an e-commerce system, the more likely it is that they may have the same preferences and the similar product purchase inclination. However, it is subjective to measure the similarity of two customers' preferences directly based on the web pages of interest that they have in common. Moreover, it doesn't make sense to judge users' preference similarities just by yes or no. Therefore, we aim to measure the degree of similarity of two users' interest quantitatively, and then rank a group of people according to the preference similarities among them. To solve the key point of the issue, we put forward a hierarchical graph to present users' browsing histories in an e-commerce system, as shown in Figure 3. Three procedures need to be preformed before building such a graph for an e-commerce system browse path.

(1) Formulate a set of user click logs according to the time sequence in which he (or she) visits an e-commerce system and form the web browsing trajectory;

(2) Filter out the common web URLs that everyone will click in the e-commerce system, such

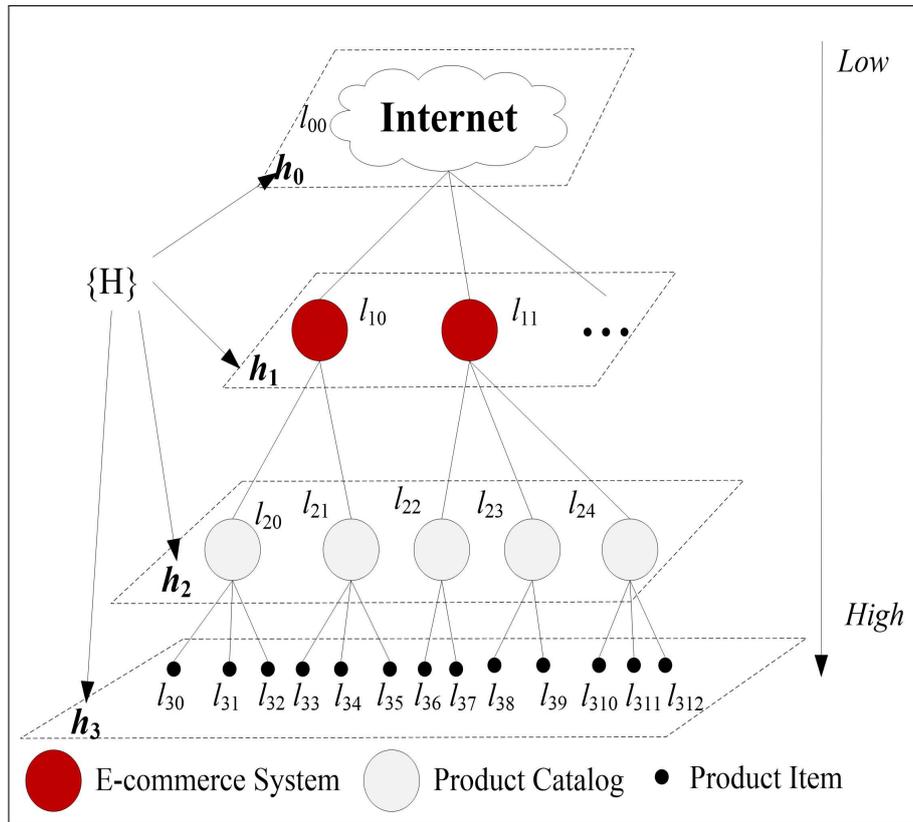


Figure 3: Hierarchical structure of the user browsing trajectory

as login, logout, product payment or other related URLs;

(3) Construct the hierarchical graph according to the web page properties of the clicked URL. We are not concerned with the web page clicked time sequence while creating the graph (in other words, a web page clicked more than one time can be repetitively treated as a multiple node in the graph). Also, a web page clicked many times can be represented as multiple nodes in the hierarchical graph.

Definition 3. Hierarchical Graph (HG). HG is a collection of clicked URLs in an e-commerce system, with a hierarchical structure $HG = \{H, L\}$, where $H = \{h_1, h_2, \dots, h_{n-1}, h_n\}$ represents the collection of layers of the hierarchy graph. $L = \{l_{ij} | 0 \leq i \leq |H|, 0 \leq j \leq |L_i|\}$, where l_{ij} denotes the j th nodes on the layer h_i ($h_i \in H$), and L_i is the set of nodes on layer l_i

3.3 Users' preference similarity measurement

Concepts of similar web click sequences

Definition 4. Similar Web Click Sequence (SWCS). A similar web click sequence represents for two users (u_p and u_q) who have viewed the same sequence of URLs within the same period. Formally, a pair of web-click-sequences, $webclickseq_i^p$ and $webclickseq_i^q$, for two users, u_p and u_q ,

$$webclickseq_i^p = \left[url_1^p(t_1^p) \xrightarrow{\Delta t_1^p} url_2^p(t_2^p) \xrightarrow{\Delta t_2^p} url_3^p(t_3^p) \dots \xrightarrow{\Delta t_{n-1}^p} url_n^p(t_n^p) \right]$$

$$webclickseq_i^q = \left[url_1^q(t_1^q) \xrightarrow{\Delta t_1^q} url_2^q(t_2^q) \xrightarrow{\Delta t_2^q} url_3^q(t_3^q) \dots \xrightarrow{\Delta t_{n-1}^q} url_n^q(t_n^q) \right]$$

where $url_j(1 \leq j \leq n)$ are the graph vertices that u_p and u_q share on the layer l_i , $t_j(1 \leq j \leq n)$ stands for the times the visitor successively stay on the web page url_j and $\Delta t_j(1 \leq j \leq n)$ represents the times interval that the user transfer from url_j to url_{j+1} . $webclickseq_i^p$ and $webclickseq_i^q$ are similar web click sequences only if they meet the following conditions:

(1) The two users, u_p and u_q , share the same vertex in one layer of their HGs. Formally, $\forall 1 \leq j \leq n, url_j^p = url_j^q$.

(2) The two users, u_p and u_q , have assemble transition times between the orderly accessed web pages. Formally, $\forall 1 \leq j \leq n, 0 \leq \frac{|\Delta t_j^p - \Delta t_j^q|}{\max(\Delta t_j^p, \Delta t_j^q)} \leq T_{threshold}$, where $T_{threshold}$ is a predefined time threshold.

If the above two conditions hold, a similar web click sequence, $simwebclickseq_i^{q,p}$, included in $webclickseq_i^q$ and $webclickseq_i^q$ can be extracted as:
 $simwebclickseq_i^{p,q} = \langle url_1^{p,q}(\min(\Delta t_1^p, \Delta t_1^q)) \rightarrow url_2^{p,q}(\min(\Delta t_2^p, \Delta t_2^q)) \rightarrow \dots \rightarrow url_n^{p,q}(\min(\Delta t_n^p, \Delta t_n^q)) \rangle$, where $\min(\Delta t_1^p, \Delta t_1^q)$ stands for the minimum of the time intervals Δt_1^p and Δt_1^q .

Definition 5. *n*-Length Similar Web Click Sequence. If there is *n* nodes in the similar web click sequence $simwebclickseq_i^{q,p}$ for two users, u_p and u_q , we call the sequence *n*-length similar web click sequence.

Similar web click sequences extracting

It can be found in Figure 3 that the bottom nodes in the hierarchy graph reveal more specific preferences than those at the top. Also, from the top to the bottom, the customers' shopping intentions increase gradually. Therefore, the hierarchical characteristic of this graph is efficient in depicting individuals' preference similarity. Customers who share the same web browsing trajectory on a lower layer are more likely to have similar shopping preferences than those who have web browsing trajectories in common on a higher layer.

According to the user browsing history hierarchical presentation model, we propose the following users' preference similarity (*UPS*) measure approach and define preference similarity between user *p* and *q* as:

$$UPS(p, q) = \frac{SumScore_{p,q}}{|HG_p| + |HG_q|} \quad (1)$$

where HG_p and HG_q denote the hierarchical graphs of the user *p* and *q*, $SumScore_{p,q}$ denotes the sum of the score for the same node in each layer, $|HG_p|$ is the number of nodes in HG_p , $|HG_q|$ is the number of nodes in HG_q . The sum score ($SumScore_{p,q}$) value of two users (*p* and *q*) need to be calculated according to the URLs they share in each layer. The detailed process of acquiring $SumScore_{p,q}$ can be described in the following algorithm.

Obviously, the weighted factor α , in the algorithm, can impact the result of users' preference similarity to a certain degree. We set $f(i) = \frac{1}{|H|-i+1}$ ($|H|$ is the total layers of the hierarchical graph) to normalize the preference similarity value to $[0,1]$. Additionally, the greater $UPS(p, q)$ value means more preference similarities between users *p* and *q*. The process of calculating the total length of the shared trajectory for users *p* and *q* in each layer is illustrated in the following algorithm (**GetSimilarClickSeqLength**), and the demonstration of similar web page click sequence matching is presented in Figure 4.

4 Experiment evaluation

To examine the effectiveness of the proposed web page navigation logs mining based user preference similarity measure approach, we conduct the experiments on dataset collected from

Algorithm 2 UPSAWBH**Require:** Hierarchically structured web browsing trajectory HG_p, HG_q of users p and q **Ensure:** $UPS(p, q)$ which shows the preference similarities of users p and q

```

1:  $SumScore_{q,q} = 0, UPS(p, q) = 0$ 
2: while  $h_i \in H$  do
3:    $score_h = 0$  // preference similarity on a layer
4:    $\alpha = f(i)$  //  $\alpha$  is an  $i$ -dependent factor
5:   while  $l_i \in L$  do
6:      $len = GetSimilarSeqLength(HG_p^i, HG_q^i)$  // get similar web page length
7:      $score_h = 2len * \alpha$  // get the total length of a similar web page
8:   end while
9:    $SumScore_{q,q} = SumScore_{q,q} + score_h$ 
10: end while
11:  $UPS(p, q) = \frac{SumScore_{p,q}}{|HG_p| + |HG_q|}$  // calculate  $UPS$  according to formula (1)
12: return  $UPS(p, q)$ 

```

Algorithm 3 GetSimilarClickSeqLength**Require:** Web browsing list set L_p, L_q **Ensure:** the total length ($TotalLength$) of the similar web page click sequence that user p and q share in L_p and L_q

```

1: Sort  $L_p, L_q$  according to the web click time order, and form the web browsing trajectories  $Traj_p$  and  $Traj_q$ 
2:  $indictor = 0, TotalLength = 0$  // variable initialization
3: while  $url_p$  in  $Traj_p$  do
4:   while  $indictor < |Traj_q|$  do
5:     if  $url_p == Traj_q[indictor]$  then  $TotalLength ++$ 
6:     end if
7:      $indictor ++$ 
8:   end while
9: end while
10: return  $TotalLength$ 

```

the server of our self-developed e-commerce system. The dataset covers all the system click logs during a whole month. Table 1 depicts the profile of the dataset of the dataset that we used in the experiment, and Figure 5 details the click trajectories we extracted. It suggests that most of the system users has trajectory length of 1, which means many users just access a web page of the system and exit within a very short time, because the search engine www.Baidu.com is chosen to promote our system, and many link-clicks from the search engine. Meanwhile, the numbers of users decrease sharply with the increase of the click trajectory length in our system.

All the proposed algorithms and models are implemented and run in a computer with Intel (R) Core™ i3-2310 CPU @2.10 GHz (4 CPUs), 6GB RAM, Windows 7 Ultimate 64-bit using visual Programming Language C# (in Microsoft Visual Studio 2010 Professional). To differentiate the significance of similar web page sequences with various length values on different layers, we set $\alpha = \frac{1}{|H|^{-i+1}}$. Here α increases in accordance with the layer i in the hierarchical graph, since we intuitively observe that the likelihood of two individuals' preference similarity rises sharply. There are thousands of combinations of time threshold (θ_t) with factor (α) in algorithm UPSAWBH. Honestly, it is a great challenge to determine what time interval is proper to detect interested

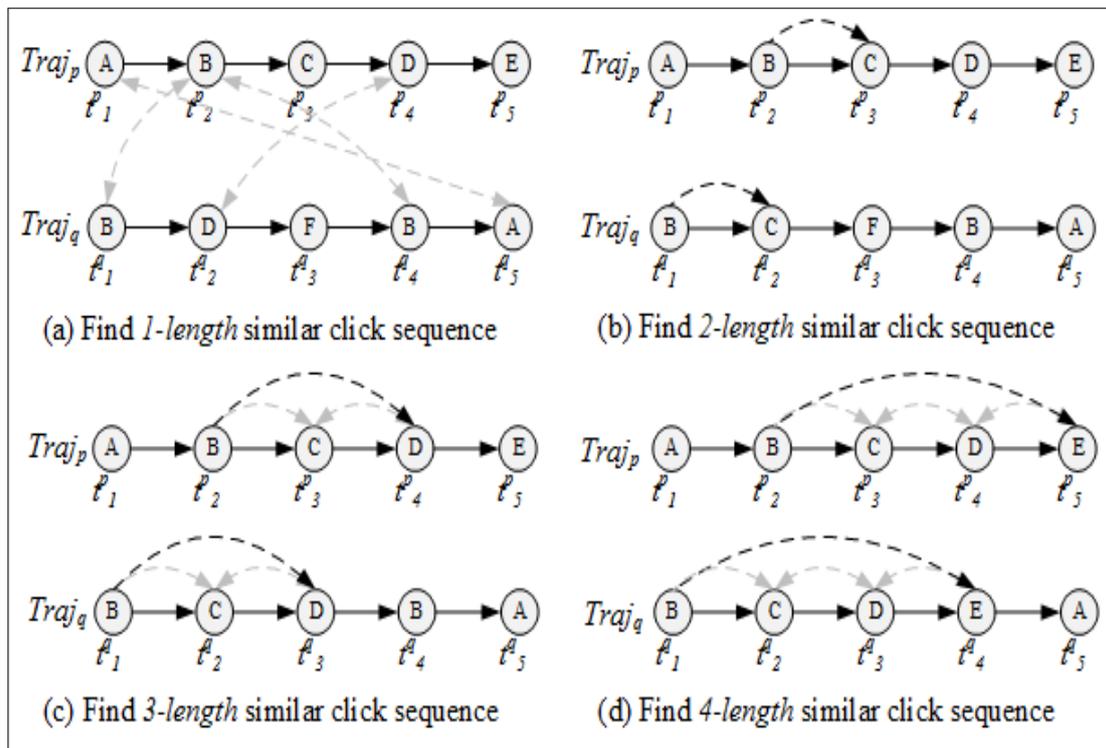


Figure 4: Schematic of similar web page click sequence matching

Table 1: Statistical information of the web browsing log data

Item	Count
Total request(Number of entries)	1685
Aver request/Hour	86.46
Average bytes transferred(MB)	4198.56
Average bytes transferred per hour	164.61
No of request after pre-processing	5417
Number of users	305
Number of user sessions	317

web pages of an individual. Therefore, we set the values of θ_t with reference to commonsense knowledge in real-world web system design. In our experiment, θ_t is set to 5 seconds, 10 seconds, 15 seconds, 20 seconds, 25 seconds, 30 seconds, 35 seconds, 40 seconds and 45 seconds. The result, as shown in Figure 6, reveals that that the number of interested web pages decrease rapidly as the time threshold θ_t increases. Take $\theta_t = 5$ as an example, nearly 290 interested web pages are detected, but the value drops to less than 80 when θ_t is set to 10 seconds.

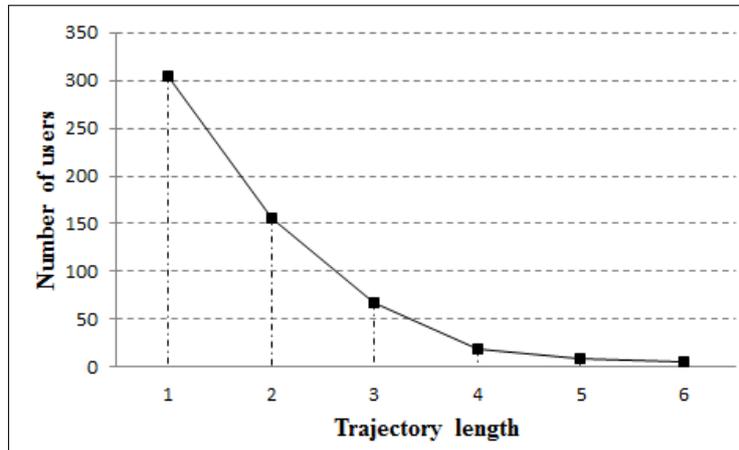


Figure 5: Relationship between trajectory length and number of users

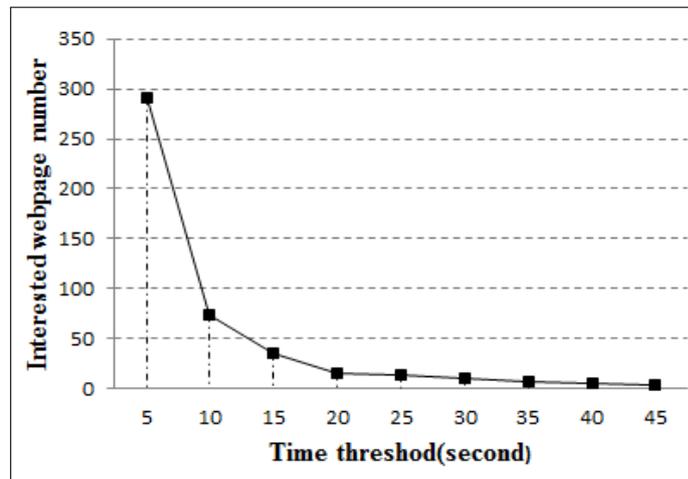


Figure 6: Number of interested web page changing over time threshold

We mainly focus on verifying the feasibility and effectiveness of hierarchical graph, interested-webpage-detection algorithm, web click sequence, in measuring users' similarity in e-commerce systems. To further explore the accuracy of proposed UPSAWBH algorithm, six candidate total layer values in the hierarchical graph are tested in the experiment. As we can see in figure 7, with the increase in the total layer in the hierarchical structure of user the browsing trajectory, the accuracy of UPSAWBH nearly increases linearly, and the calculated preference similarity value (*UPS*) rises simultaneously. Intuitively, the more different layer levels in hierarchical graph considered the more similar web click sequences can be checked, that is, both preference similarity and average accuracy are better while 6 total layers are considered. All these prove that our approach has good effectiveness in mining users' preference similarity knowledge using webpage navigation logs of users in e-commerce systems.

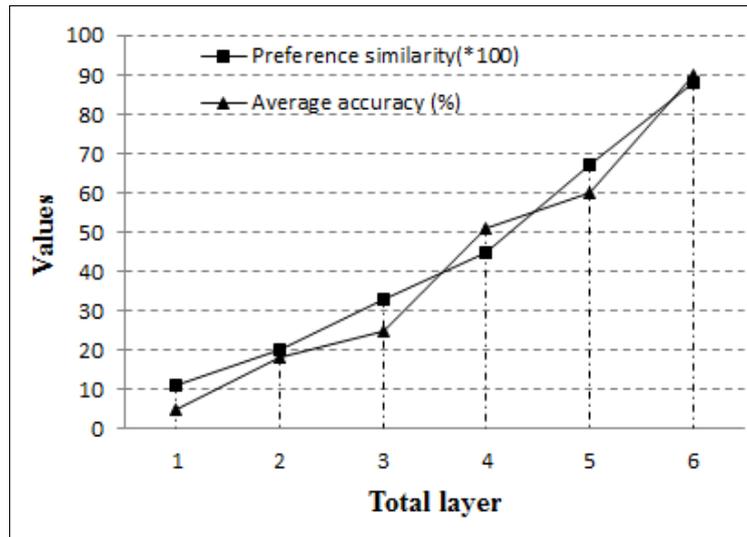


Figure 7: The influence of total layer parameter on the accuracy of the UPSAWBH algorithm

5 Conclusion and future work

Users' preference similarities mining is an important research area in many information systems, especially in e-commerce application. To some extent alleviate challenges, such as cold start and data sparsity, in e-commerce environment, the article presents a method to explore users' preference similarities based on web browsing history data mining. In this research, we introduce a web page-navigation-log data extraction based users' similarity calculation approach which (1) starts from a perspective of geography, treating individuals' visits of web pages in e-commerce systems as trajectories, (2) devotes to explore the users' preference based on web page click sequence discovering and (3) tries to check users' preferences similarity in terms of their web click pattern resemblance in distinct levels. In order to construct a complete a research framework of our approach, a hierarchical graph model is proposed to present the structure of web navigation history and an interested web page detection algorithm is put forward to explore users' preference patterns. Meanwhile, an algorithm, UPSAWBH, is put forward to figure out the preference similarities among users, and the experimental results on dataset from real-world e-commerce system sever prove good effectiveness of our approach. Two traits, the sequence peculiarity of user click and hierarchy feature of webpage browsing levels, have been considered in this similarity measure.

One limitation of our method is that it has been tested using only dataset from our self-developed e-commerce system and we just set the value of weighted factor (α), by intuition. In the future, we will pay more attention to collect web page navigation dataset from other B/S based e-commerce systems to compare the result and improve our approach. We plan to focus on the following two areas: (1) further explore the impact of weighted control factor (α) on the result of UPSAWBH quantitatively, and (2) investigate how to improve the veracity of the proposed user preference similarity computation method to meet the requirement of personalization recommendation. We have just put forward a rough research paradigm to mine users' preference similarities using web page navigation logs in e-commerce system from another perspective, and still have a long way to go.

Acknowledgments

This work is partly supported by the Ministry of Education, Humanities and Social Sciences Research Project (Grant No. 14YJC630210), the Zhejiang Public Technology Research and Application Project (Grant No. 2015C33065), the Zhijiang Youth Action Project: study on mobile e-commerce recommendation (Grant No. G306), the Ningbo Education Science Planning Key Project (Grant No. 2017YZD010), the Zhejiang Business Economics Association Project (Grant No. 2016SJYB01), the China National Natural Science Foundation Project (Grant Nos. 61202376, 71071145 and 41201550), the Natural Science Foundation of Zhejiang (Grant No. LY16G020012), the Major Research Projects of Humanities and Social Sciences in Colleges and Universities of Zhejiang (Grant No. 2014GH015), the Ningbo Huimin Project of Science and Technology (Grant No. 2016C51040), the China Press and Publication Administration Key Laboratory Project and Shanghai Key Lab of Modern Optical System, the Modern Port Service Industry and Culture Research Center of the Key Research Base of Philosophy and Social Sciences of Zhejiang Province and the China Scholarship Council.

Bibliography

- [1] Abraham S., Lai P.S. (2011); Spatio-temporal Similarity of Web User Session Trajectories and Applications in Dark Web Research, *Proceedings of the Pacific Asia Workshop on Intelligence and Security Informatics*, Beijing, China, 2011. doi:10.1007/978-3-642-22039-5_1.
- [2] Adomavicius G., Tuzhilin A. (2005); Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Transaction on Knowledge and Data Engineering (TKDE)*, 17(6), 734-749, 2005. doi:10.1109/TKDE.2005.99.
- [3] Becchetti L. et al. (2014); A lightweight privacy preserving SMS - based recommendation system for mobile users, *Knowledge and Information Systems*, 40(1), 49-77, 2014.
- [4] Borges J., Levene M. (2000); *Web usage analysis and user profiling, Chapter: Data mining of user navigation patterns (92-112)*, San Diego, CA, USA; Springer Berlin Heidelberg, 2000.
- [5] Chen D.-N. et al. (2010); A Web-based personalized recommendation system for mobile phone selection: Design, implementation, and evaluation, *Expert Systems with Applications*, 37(12), 8201-8210, 2010. doi:10.1016/j.eswa.2010.05.066.
- [6] Cheng A.-J. et al. (2011); Personalized travel recommendation by mining people attributes from community-contributed photos, *Proceedings of the 19th ACM international conference on Multimedia*, Scottsdale, Arizona, USA; ACM, 83-92, 2011.
- [7] Dao T.H., Jeong S.R., Ahn H. (2012); A novel recommendation model of location-based advertising: Context-Aware Collaborative Filtering using GA approach, *Expert Systems with Applications*, 39(3), 3731-3739, 2012. doi:10.1016/j.eswa.2011.09.070.
- [8] Eckhardt A. (2012); Similarity of users' (content-based) preference models for Collaborative filtering in few ratings scenario, *Expert Systems with Applications*, 39(14), 11511-11516, 2012. doi:10.1016/j.eswa.2012.01.177.
- [9] Eirinaki B.M., M Vazirgiannis M. (2003); Web mining for web personalization, *ACM Transactions on Internet Technology*, 3(1), 1-27, 2003.

-
- [10] Ezeife C.I., Lu Y. (2005); Mining web log sequential patterns with position coded pre-order linked WAP-tree, *Data Mining and Knowledge Discovery*, 10(1), 5-38, 2005. doi:10.1007/s10618-005-0248-3.
- [11] He S., Fang M. (2008); Personalized recommendation based on ontology inference in E-commerce, *Proceedings of the International Conference on Management of e-Commerce and e-Government*, 192-195, 2008. doi:10.1109/icmecg.2008.24.
- [12] Gan M., Jiang R. (2013); Constructing a user similarity network to remove adverse influence of popular objects for personalized recommendation, *Expert Systems with Applications*, 40(10), 4044-4053, 2013. doi:10.1016/j.eswa.2013.01.004.
- [13] Guy I. et al. (2010); Same places, same things, same people?: mining user similarity on social media, *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, Savannah, Georgia, USA, 41-50, 2010. doi:10.1145/1718918.1718928.
- [14] Leavitt N. (2006); Recommendation technology: Will it boost E-Commerce, *Computer*, 39(5), 13-16, 2006. doi:10.1109/MC.2006.176.
- [15] Lee M.-J., Chung C.-W. (2011); A user similarity calculation based on the location for social network services, *Proceedings of the Database Systems for Advanced Applications*, Hong Kong, China, Springer, 38-52, 2011. doi:10.1007/978-3-642-20149-3_5.
- [16] Li P. et al. (2007); Preference update for e-commerce applications: Model, language, and processing, *Electronic Commerce Research*, 7(1), 17-44, 2007. doi:10.1007/s10660-006-0061-0.
- [17] Li Q. et al. (2008); Mining user similarity based on location history, *Proceedings of the ACM SIGSPATIAL GIS*, Irvine, CA, USA, 2008. doi:10.1145/1463434.1463477.
- [18] Linden G., Smith B., York J.(2003); Amazon.com recommendations: item-to-item collaborative filtering, *IEEE Internet Computing*, 7(1), 76-80, 2003. doi:10.1109/MIC.2003.1167344.
- [19] Papazoglou M.P. (2001); Agent-oriented technology in support of e-business - Enabling the development of "intelligent" business agents for adaptive, reusable software, *Communications of the ACM*, 44(4), 71-77, 2001. doi:10.1145/367211.367268.
- [20] Park S.T., Pennock D., Madani O. (2006); Collaborative filtering for robust cold-start recommendations, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, 2006.
- [21] Sarwar B.M. (2001); *Sparsity, scalability and distribution in recommender systems*, Minneapolis, USA, University of Minnesota, 2001.
- [22] Shahabi C. et al. (1997); Knowledge discovery from users web-page navigation, *Proceedings of the Seventh International Workshop on Research Issues in Data Engineering*, Birmingham, England, 1997. doi:10.1109/RIDE.1997.583692.
- [23] Song Q., Shepperd M. (2005); Mining web browsing patterns for E-commerce, *Computers in Industry*, 57(7), 622-630, 2006. doi:10.1016/j.compind.2005.11.006.
- [24] Srivastava J. et al. (2000); Web usage mining: Discovery and applications of usage patterns from web data, *ACM SIGKDD Explorations Newsletter*, 1(2), 12-23, 2000.
- [25] Turban E. et al. (2009); *Electronic Commerce*, NJ, USA: Prentice Hall Press, 2009.

- [26] Waga K., Tabarcea A., Franti P. (2011); Context aware recommendation of location-based data, *Proceedings of the 15th International Conference on System Theory, Control, and Computing (ICSTCC)*, Sinaia, Romania, 1-6, 2011.
- [27] Wang Y.-T., Lee A.J.T. (2011); Mining web navigation patterns with a path traversal graph, *Expert Systems with Applications*, 38(6), 7112-7122, 2011. doi:10.1016/j.eswa.2010.12.058.
- [28] Woerndl W., Brocco M., Eigner R. (2009); Context-aware recommender systems in mobile scenarios, *International Journal of Information Technology and Web Engineering*, 4(1), 67-85, 2009. doi:10.4018/jitwe.2009010105.
- [29] Wu C., Hou F. (2011); Design and Optimization of Redundant ControlNet Networking Control System, *Process Automation Instrumentation*, 32(3), 50-56, 2011.
- [30] Wu C., Yu Z. (2005); Data transmission with data package dropout and control method on NCS, *Control & Automation*, 10, 39-41, 2005.
- [31] Ying J. J.-C. et al. (2010); Mining user similarity from semantic trajectories, *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, San Jose, California, 19-26, 2010. doi:10.1145/1867699.1867703.
- [32] YU X.B., GUO S.S., HUANG X.R. (2010); Intelligent e-commerce based on Web usage mining and its application (in Chinese), *Computer Integrated Manufacturing Systems*, 16(2), 439-448, 2010.
- [33] Zhong H. et al. (2014); Study on Directed Trust Graph Based Recommendation for E-commerce System, *International Journal of Computers Communications & Control*, 9(4): 510-523, 2014.

A New Deep Learning Approach for Anomaly Base IDS using Memetic Classifier

S. Mohammadi, A. Namadchian

Shahriar Mohammadi*, **Amin Namadchian**

Department of Industrial Engineering
aK.N. Toosi University of Technology,
Tehran, Iran

*Corresponding author: Mohammadi@kntu.ac.ir
anamadchian@mail.kntu.ac.ir

Abstract: A model of an intrusion-detection system capable of detecting attack in computer networks is described. The model is based on deep learning approach to learn best features of network connections and Memetic algorithm as final classifier for detection of abnormal traffic.

One of the problems in intrusion detection systems is large scale of features. Which makes typical methods data mining method were ineffective in this area. Deep learning algorithms succeed in image and video mining which has high dimensionality of features. It seems to use them to solve the large scale of features problem of intrusion detection systems is possible. The model is offered in this paper which tries to use deep learning for detecting best features. An evaluation algorithm is used for produce final classifier that work well in multi density environments.

We use NSL-KDD and Kdd99 dataset to evaluate our model, our findings showed 98.11 detection rate. NSL-KDD estimation shows the proposed model has succeeded to classify 92.72% R2L attack group.

Keywords: Deep learning, KDD99, memetic algorithm, NSL-Kdd, classification function, anomaly base intrusion detection, intrusion-detection system (IDS).

1 Introduction

In recent years, security industry has been actively playing roles in dealing with security threats against computer organizations and networks through employing various technologies such as encryption, authentication, and access control. However, all these technologies are also involved with their specific limitations, allowing an attacker to enter the system. IDSs¹ are security mechanisms which intelligently monitor computer and network systems in real time to detect intrusions and take quick appropriate measures in response [3].

Anomaly- and signature-based are two main methods used in IDSs. The former is based on the statistical description of users or application programs which is ultimately intended to detection any activity deviating from the profile of normal behavior, which in fact is an indication of an abnormal behavior conducted by users or application programs [7].

Signature-based IDSs work based on collecting and storing the signature of known attacks, and the IDS attempts to search the logged events for patterns matching the signature of stored attacks. Both methods have their specific advantages and drawbacks. Signature-based systems have an appropriate accuracy in detection of known attacks and generate few number of false positives. On the other hand, they are only capable of detecting previously modeled attacks. On the contrary, anomaly-based IDSs are capable of detecting new attacks, but also produce a large number of false positives and may identify a normal behavior as suspicious due to deviation from the defined threshold. Another challenge in using these systems is their difficulty in adapting to dynamic environments [10].

¹Intrusion detection systems

Many challenges are faced in order to improve the IDSs. The large number of application programs has led to extensive features in describing normal behavior. Moreover, there are more complex attacks developed everyday taking advantage of several vulnerabilities in different application programs. As a result, many parameters of high dimensions are involved in the IDSs. As a solution, attempts were made in many studies to further simplify the problem through selecting effective features and decreasing their numbers [8].

Different studies attempted to select appropriate features for intrusion detection and to develop a learning model through an algorithm. [1]. classified these studies into four categories, namely classification, clustering, statistical, and information theory. In [12, 14], cross-entropy analysis was employed to derive the appropriate features for each attack type. All the mentioned methods are highly dependent on the dataset as they attempt to extract the appropriate features according to those of the dataset.

Similar to intrusion detection, we are also faced with the problem of selecting appropriate features from among a large number of features in the real environment in applications such as image, video, and sound processing. Recent studies have managed to achieve acceptable results in feature selection through a deep-learning approach [15].

The main idea behind deep-learning is the assumption that data are composition of factors or features created in a hierarchical manner. Many other general assumptions can further improve deep learning. These seemingly simple assumptions allow exponentially finding relationships between some of the regions and samples. This can be a solution to some of the high-dimensional challenges of the problem [6].

In the proposed algorithm, We used a deep learning algorithm for regression function is obtained through deep Auto-encoder. Then use Memetic algorithm to generate a linear classification function to detect attack. This algorithm helps system to bypass local minima and become convergent, faster. Therefore, unlike papers which used genetics [14], [13] we used all KDD training datasets and enhanced our precision as well.

The architecture and components of proposed system are introduced in section 2. Section 4 deals with deep learning algorithm and its characteristics for obtain linear classification function for each class. In section 5 you know how Memetic algorithm works to combine results of classification functions is implied. and finally discussion and conclusion are presented in section 7.

2 The proposed algorithm

Our proposed algorithm is composed of three phases. In first phase we prepare and normalized dataset. In second phase we use a deep learning Auto-Encoder model to produce a regression function and in last phase we use a Memetic for produce a classifier function by those model. As shown in Fig 1 our dataset compose from three parts: training and validation part for learning Auto-Encoder model and test part for evaluation the proposed algorithm. Each record in dataset represent by data and it's label, normalized data feed into input layer of deep Auto-encoder model, and labels are assigned as input the stochastic guardian descent to find optimal model that fit with validation data. After that, we apply Memetic algorithm on reached features as final classification function that can detect normal and abnormal traffic. Finally at test phase, this model is evaluated with test data. The details of proposed architecture will be in the next sections.

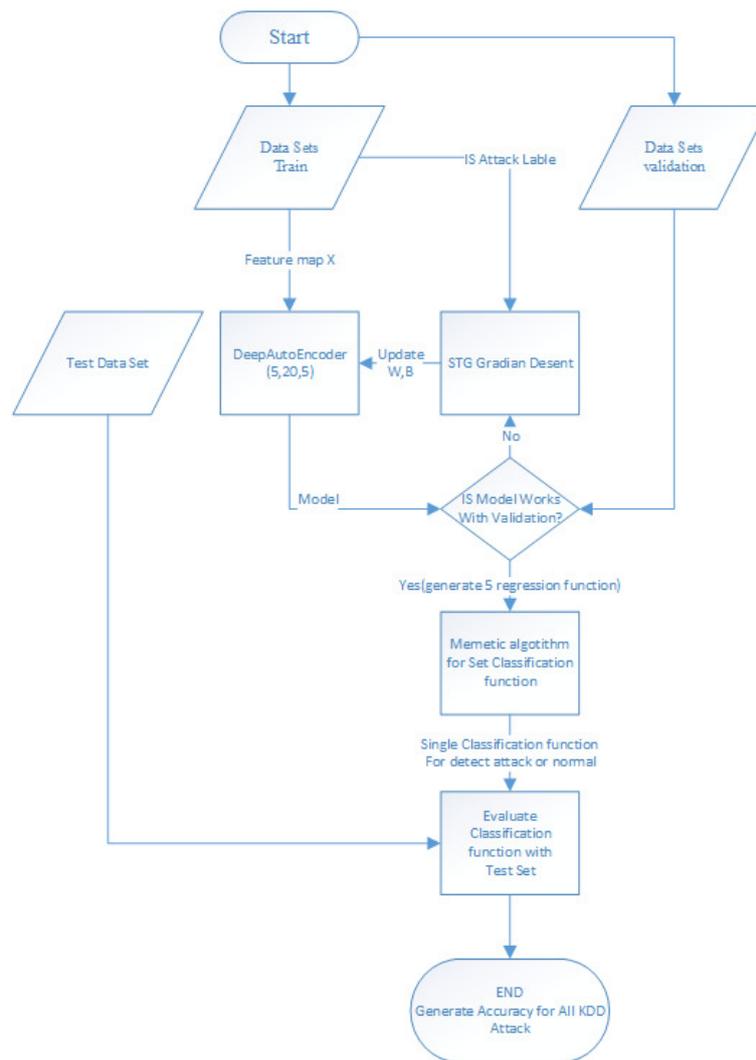


Figure 1: Architecture of the proposed algorithm

In first phase we must prepare our data for learning. For this purpose all extracted features from data must be normalized to real number with method that is described in section 3.

In next phase we train a deep learning for producing a model as regression function that can assign attack score which is explained with detail in section 4.

Our objective in the third phase is to combine the mentioned functions to learn classification that is able to report an attack is occurred or not occurring. For generating this function, the Memetic algorithm is used. This algorithm works based on an evaluation function.

After obtaining a proper classification, system may isolate attacks from ordinary traffic through this function. KDD99 dataset was used to test this algorithm.

3 Normalize dataset

The quality of analyzed data plays a significant role in enhancing the precision of data mining algorithms. We use a gradient descent optimization algorithms in our deep neural network model then need data with zero mean and equal variance. Z-score is one method that have a distribution with a mean of zero and a standard deviation of one and perfect for normalization. X is feature data, μ is mean and σ is standard deviation in equation 1.

$$X_{normalized} = \frac{X - \mu}{\sigma} \quad (1)$$

KDD use text and number. For text feature we assign a number for every value of text can give features. For example for service feature we use 0 for Ftp and 1 for Http and so on. After that all feature has a number value. We use this formula to assign a number between -1 and 1 to each features.

4 Deep learning approaches

Deep learning is a branch of machine learning based on several layer of non-linear operations that attempt to provide high-level abstractions for complex data. Several algorithms can be building blocks that put on each other as stacks to shape a deep architecture like: Convolutional neural networks, Deep Belief Network, Boltzmann machine, Restricted Boltzmann machine, neural networks, Auto-encoder, Gated Auto-encoder; and many models of this group are based on unsupervised learning. The output of each layer becomes the input of the next layer; therefore features that learned in upper layers are more abstracted [6].

The aim of an Auto-encoder is to learn a compressed representation for a set of data, typically for the purpose of dimensionality reduction. Auto-encoder is based on the concept of Sparse coding [11]. AE can be considered as a discriminative DNN in which the target output would be similar to the input, and the number of hidden layer nodes is lower than input. Therefore, it can be an unsupervised or supervised method. Its training measure is usually composed of two terms: minimization of the construction error and regularization policy. Considering different relations for regularization, various types of Auto-encoders such as de-noising AE, contractive AE, sparse AE can be created [4,5,9]. After training of the network is finished the hidden layer result considered as compressed representation of the input data.

We have five group data in KDD include attack and normal traffic. We use deep Auto-encoder that encode 41 features to 5 features group that show the score of is member of five group Dos , Normal , Probe , U2R , R2L. As shown in Fig 2, we design an Auto-encoder deep model composed of three layers of encoder-decoder and a regression at last layer. This supervised model received vectors with 41 dimension as input and attempt to learn a representation that can discriminant them according to their label. After training of this neural network is done, we have a feature vector with five dimension at last hidden layer of encoders as representation of input data attack score. This encoders can work as a regression function.

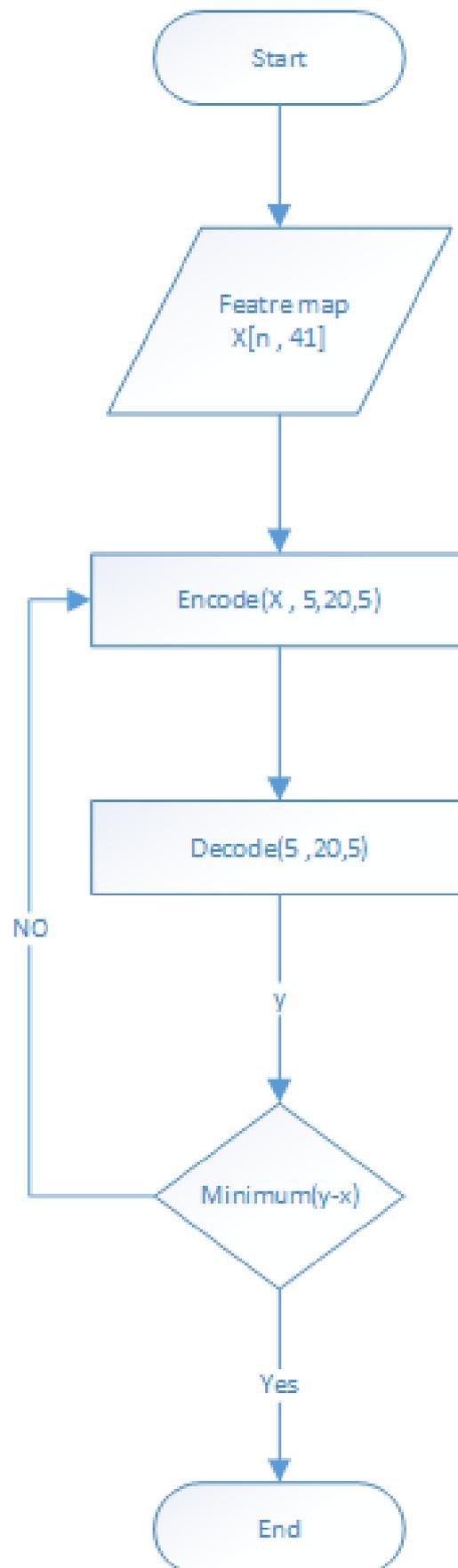


Figure 2: Architecture of the auto-encoder deep learning for generate regression function for five groups

Deep Auto-encoder can learn features as well as regression function. So we can use this algorithm without any feature selection. Our goal in this step is find best representation of data for five group in KDD after this step we need to combine this functions and generate one classification function.

5 Memetic as final classifier

Whenever connection information is bestowed to Auto-encoder of each class, it may have different features due to the mentioned connection belonging to attacks or normal traffic. Thus, we need an algorithm which concludes such results and makes decision about normal or abnormal condition of a connection. To do this, a linear memetic classifier were employed.

In the training phase, we evaluated data of each connection using regression function of each class and map to five numbers that show score of attack probabilities. Then this new features is given to the Memetic classifier as input. We try in Memetic algorithm learn from high level features that show attack probabilities final decision about the normal or attack traffic.

As it can be seen in Fig 1 in our proposed algorithm has been used to obtain linear classification functions. Values of each gene are coefficients of linear classifier function and are in the range of (-1023,1023). Evaluation function for each chromosome is as 2 which reflects classification precision:

$$CR = \frac{TP + TN}{SizeofDataSet} \quad (2)$$

Algorithm 1 Architecture of the memetic algorithm for combine five regression function for detect normal or attack

Data: Training Data Set

Result: Memetic classifier for intrusion detection

Initializes P randomly (P = Population);

Perform local search and find best fitness in its neighbourhood explored chromosome of each individual in P;

repeat

 Select parents from P;

 Generate offspring applying recombination To the parent selected;

if an individual is selected to undergo mutation **then**

 Then apply local search;

end if

 Evaluate fitness of current individual and its neighbours;

 Adopt best chromosome;

until best chromosome $fitness > \sigma$

For local search in algorithm 1 we use a simple local search algorithm like hill claiming, We try to find best fitness by explore neighbourhood of chromosome by changing the gene value.

Where σ (a double value that can be in 0,1 range) which shows precision rate. The more this value is close to 1 result more take runtime and the more precise classification functions. The fitness function for algorithm 1 can compute equation 1 shows in 2.

Algorithm 2 Fitness function for memetic algorithm

Data: individual chromosome gene[0-n] and learning dataset
Result: fitness of individual
sum=0;
for all connection record k in learning data set **do**
 SelectedFeature_k = deepEncode(x);
 if connection k is attack **then**
 if $\sum_{i=0}^{n-1} gene_i * Normalized(k) < gene_n$ **then**
 sum = sum + 1
 end if
 else
 if $\sum_{i=0}^{n-1} gene_i * Normalized(k) \geq gene_n$ **then**
 sum = sum + 1
 end if
 end if
end for
fitness = $\frac{sum}{SizeofDataSet}$;

As you can see in algorithm 2 if correction rate for a classification function compute as fitness each classification function is "N" gens that multiple to normalize(according section 3 method) feature of connection if the compute value less than last gen value classified as attack otherwise classified as normal.

6 Evaluation result

For evaluating intrusion detection, a proper dataset should be selected in the first phase which either meet the necessary standards or be comparable with other works. Kd99 [17] is a proper dataset. [16] produce a dataset called NSL-Kdd ² was introduced that have some benefit to evaluate intrusion. We use both KDD and NSL-KDD datasets to test our algorithm in order to both preserve our work's comparability with its previous counterparts and to negate some drawbacks of KDD99 in our evaluation.

We use correction rate percent that show in equation 2 on testing part of KDD99 For evaluate our model. Results of the proposed algorithm are presented and compared with the similar algorithm in table 1.

Our deep Auto-encoder have four layer in each encoder and decoder phases. Memetic algorithm parameters is number of population, mutation probability and crossover probability were selected as 100, 0.03 and 0.9 respectively and DSCG local searching function was used in our simulation.

We compare our algorithm with four algorithms on KDD99 dataset in table 1 on the first column we use only Memetic algorithm without deep Auto-encoder on our model as you can see result improve with select five best regression function for all group. More abstraction and manifold learning feature of the deep Auto-encoder and diminution reduction cause better result. Tao Xia, et al. [18] try to use genetic with information theory on KDD99 we compare with their result on column "GA" you can see 42.49 percent better result in R2L attack. We have a little records on train for R2L and as result show feature generation and manifold learning of deep Auto-encoder can solve this problem better than information theory. Nahla Ben Amor, et al. [2]

²<http://nsl.cs.unb.ca/NSL-KDD/>

use the native Bayesian network and show its better result than decision Tree. Their result shows on "Native Bayes" and "Decision Tree" columns to show deep learning and Memetic have very competitive results than this ordinary neural networks.

Table 1: Comparison of our CR results with relevant studies

Class	our(deep learning)	Memetic	GA	Native Bayes	Decision Tree
Normal	98.11	97.22	98.34	97.68	99.50
DOS	98.75	98.4	99.33	96.65	97.24
Probe	83.34	81.23	93.95	88.33	77.92
R2L	48.35	42.23	5.86	8.66	0.52
U2R	74.28	66.22	63.64	11.84	13.60

The number of selected NSL-Kdd records is based on the classification hard scattering in kdd99, thus training algorithms precision is analyzed in a larger and more precise span. In NSL-KDD, numbers of records in both test and training sets are reasonable, thus intricate methods would be implemented without a random selection of dataset records [16].KddTrain+, was used for training phase and KddTest+, was used to evaluate the algorithm.Seven algorithms run three times on each record on NSL-KDD.SuccessfulPrediction is a number in the range of 0 to 21 which indicates how many time that algorithm be to succeed on correct detection that record.SuccessfulPrediction field is a criterion to detect difficulty of classification of present records of NSL-Kdd.

In Fig 3 - 8, system precision in terms of SuccessfulPrediction field is shown. For instance, in Fig 6 for R2L class, our system has succeeded to classify 90.72% of records correctly. It is worthy to say that 0 in the mentioned field indicates that all seven algorithms have failed to classify these records in all three times.

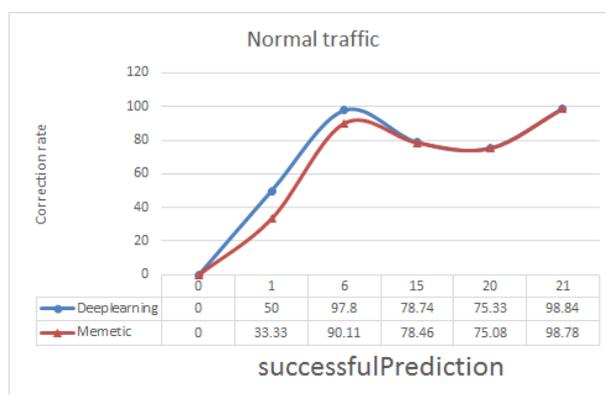


Figure 3: Comparison correction rate on NSLKDD normal attack our model with memetic algorithm

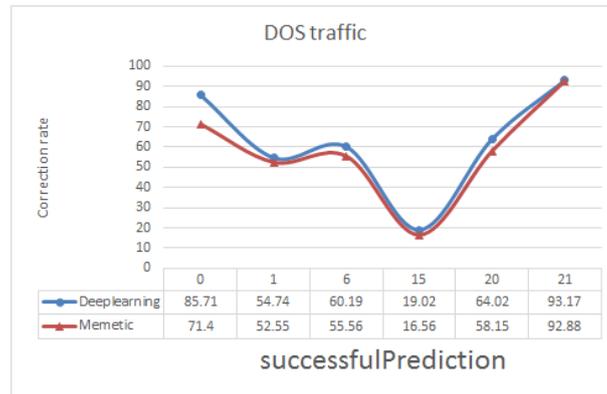


Figure 4: Comparison correction rate on NSLKDD DOS attack our model with memetic algorithm

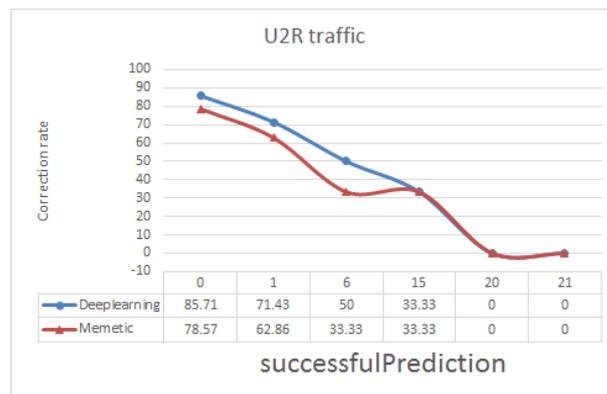


Figure 5: Comparison correction rate on NSLKDD U2R attack our model with memetic algorithm

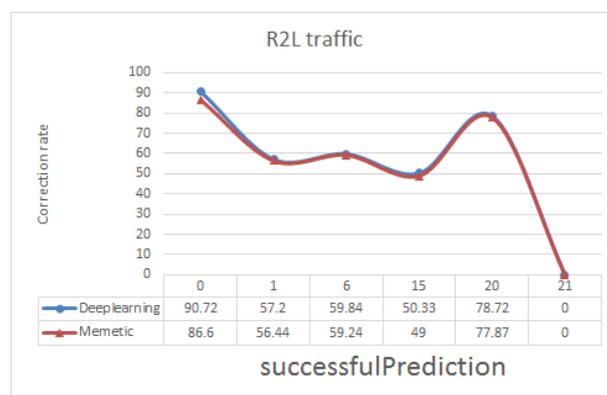


Figure 6: Comparison correction rate on NSLKDD R2L attack our model with memetic algorithm

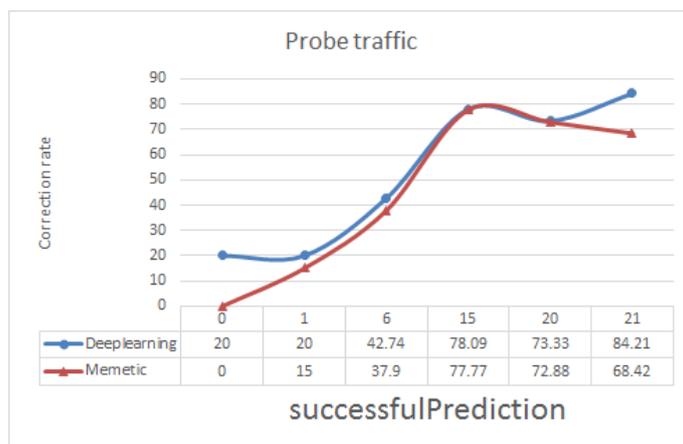


Figure 7: Comparison correction rate on NSLKDD probe attack our model with memetic algorithm

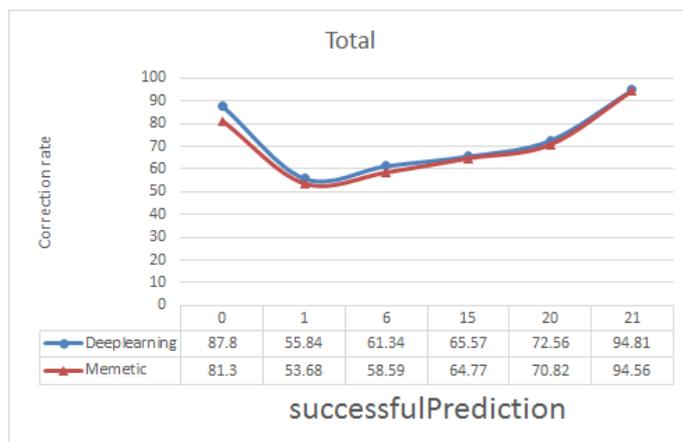


Figure 8: Comparison correction rate on all NSLKDD groups our model with memetic algorithm

7 Conclusion

Anomaly-based intrusion detection system was presented which can learn feature of attacks using DNN-Auto-Encoder and use the memetic algorithm for final classifier. The performance of present model show it can detect most of attacks correctly. Deep learning, which is proper for large scale features and Memetic algorithm which can solved local minimum optimization. In order to evaluate the performance of the proposed algorithm, results were obtained on KDD and NSL-Kdd datasets.

This study aimed at offering a deep learning model for anomaly detection engine. However future works include using deep recurrent network for on-line model. For example deep LSTM model for add signature-based intrusion detection systems as a supervisor on operational environment.

Signature-based intrusion detection systems have a trivial false positive rate but their detection rate is very low because they are able to detect only those attacks which follow their pattern. Therefore, when a signature-based intrusion detection system recognizes a connection

as a single attack, it is reliable mostly and it may even be used as an attack label to train an LSTM detection system.

Bibliography

- [1] Ahmed M., Naser Mahmood A., Hu J. (2016); A survey of network anomaly detection techniques, *Journal of Network and Computer Applications*, 60, 19–31, 2016.
- [2] Amor N. B., Benferhat S., Elouedi Z. (2004); Naive bayes vs decision trees in intrusion detection systems, *Proc. of the 2004 ACM Symposium on Applied Computing, NY, USA. ACM*, 420–424, 2004.
- [3] Axelsson S. (2000); Intrusion Detection Systems : A Survey and Taxonomy, *Computer Engineering*, 1–27, 2000.
- [4] Bengio Y. (2013); Deep learning of representations: Looking forward, *Intl. Conf. on Statistical Language and Speech Processing*, 1–37, 2013.
- [5] Bengio Y., Courville A. C., Vincent P. (2012); Unsupervised feature learning and deep learning: A review and new perspectives, *CoRR*, *abs/1206.5538*, 1, 2012.
- [6] Bengio Y., Goodfellow I. J., Courville A. (2016); *Deep Learning*, The MIT Press, 2016.
- [7] Bhuyan M. H., Bhattacharyya D. K., Kalita J. K. (2014); Network anomaly detection: methods, systems and tools, *Communications Surveys & Tutorials, IEEE*, 16(1), 303–336, 2014.
- [8] Dang Y., Wang B., Brant R., Zhang Z., Alqallaf M., Wu Z. (2017); Anomaly detection for data streams in large-scale distributed heterogeneous computing environments, *ICMLG2017 5th Intl. Conf. on Management Leadership and Governance*, 121–121, 2017.
- [9] Erhan D., Manzagol P.-A., Bengio Y., Bengio S., Vincent P. (2009); The difficulty of training deep architectures and the effect of unsupervised pre-training, *Artificial Intelligence and Statistics*, 153–160, 2009.
- [10] García-Teodoro P., Díaz-Verdejo J., Maciá-Fernández G., Vázquez E. (2009); Anomaly-based network intrusion detection: Techniques, systems and challenges, *Computers & Security*, 28(1-2), 18–28, 2009.
- [11] Ng A. (2011). Sparse autoencoder, *CS294A Lecture Notes*, 72, 1–19, 2011.
- [12] Nguyen H., Franke K., Petrović S. (2010); Improving effectiveness of intrusion detection by correlation feature selection, In *ARES 2010 - 5th Intl. Conf. on Availability, Reliability, and Security*, 17–24, 2010.
- [13] Owais S., Snasel V., Kromer P., Abraham A. (2008); Survey: Using Genetic Algorithm Approach in Intrusion Detection Systems Techniques, *2008 7th Computer Information Systems and Industrial Management Applications*, 300–307, 2008.
- [14] Qu G., Hariri S., Yousif M. (2005), A new dependency and correlation analysis for features, *IEEE Transactions on Knowledge and Data Engineering*, 17(9), 1199–1206, 2005.
- [15] Schmidhuber J. (2015), Deep learning in neural networks: An overview, *Neural Networks*, 61, 85–117, 2015.

- [16] Tavallae M., Bagheri E., Lu W., Ghorbani A. A. (2009), A detailed analysis of the KDD CUP 99 data set, *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, 2009.
- [17] University of California, I. KDD Cup 1999, 1999.
- [18] Xia T., Qu G., Hariri S., Yousif M. (2005), An efficient network intrusion detection method based on information theory and genetic algorithm, *Performance, Computing, and Communications Conference, 2005. IPCCC 2005, 24th IEEE Intl.*, 11–17, 2005.

Gravitation Theory Based Model for Multi-Label Classification

L. Peng, Y. Liu

Liwen Peng, Yongguo Liu*

Knowledge and Data Engineering Laboratory of Chinese Medicine,
School of Information and Software Engineering,
University of Electronic Science and Technology of China,
Chengdu, 610054, PR, China

*Corresponding author: liuyg@uestc.edu.cn

Abstract: The past decade has witnessed the growing popularity in multi-label classification algorithms in the fields like text categorization, music information retrieval, and the classification of videos and medical proteins. In the meantime, the methods based on the principle of universal gravitation have been extensively used in the classification of machine learning owing to simplicity and high performance. In light of the above, this paper proposes a novel multi-label classification algorithm called the interaction and data gravitation-based model for multi-label classification (ITDGM). The algorithm replaces the interaction between two objects with the attraction between two particles. The author carries out a series of experiments on five multi-label datasets. The experimental results show that the ITDGM performs better than some well-known multi-label classification algorithms. The effect of the proposed model is assessed by the example-based F1-Measure and Label-based micro F1-measure.

Keywords: data gravitation theory, interaction, multi-label classification.

1 Introduction

As a major area of research into machine learning systems, the traditional supervised classification learning technology attributes each instance to one class (label). With x as the set of all instances and l as the label space, the traditional supervised classification learning system aims to get the function $f : x \rightarrow l$ from the training dataset $\{(x_i, l_j) : 1 \leq i \leq m, 1 \leq j \leq k\}$, where m and k denote the number of instances and the number of labels, respectively. In other words, the purpose of the system lies in predicting the one correct label for each test instance. There are many algorithms for single-label classification, such as the decision tree C4.5 [4], the support vector machine (SVM) [1], the rough set [11], the neural network (NN) [8] and so on.

The traditional single-label classification may apply to some fields, but many instances do not belong to a specific class in reality, namely text categorization, music information retrieval and video classification. The limitation of the traditional single-label classification has contributed to the growing popularity in multi-label classification algorithms.

In general, multi-label classification methods are roughly divided into two categories [2]: the problem transformation method and the algorithm adaptation method. The former transforms multi-label dataset into other well-established learning scenarios, and processes the transformed dataset with the single-label classification algorithm. Typical examples are the binary relevance algorithm (BR), the label combination method (LC), high-order classifier chains [13], and random k label sets (RAkEL) [17]. By contrast, the algorithm adaptation method directly applies adaptive learning methods to solve the problem of multi-label learning. The representative algorithm adaptation methods include the k -nearest neighbor algorithm for multi-label learning (ML-kNN) [24], the back-propagation multi-label learning (BPMLL) [23], the bichromatic reverse k -nearest neighbor algorithm (BRkNN) [15] and the dependent multi-label k -nearest neighbor

algorithm (DML-kNN) [21]. As an extended version of the KNN method, the ML-kNN approach employs the maximum posteriori probability to determine the label set of test instances. Based on the label sets of the neighbor instances, the BRkNN method calculates the confidence of each label, and decides the k nearest neighbors of each test instance. The BPMLL is another classic back-propagation multi-label classification algorithm which proposes a new error function for multiple labels.

As far as we know, the methods based on the principle of universal gravitation has been extensively used in the classification or clustering of machine learning owing to simplicity and high performance. The existing physical gravitation-based classification or clustering methods mostly focus on two important factors: the data distance and density. For example, Reyes et al. [14] proposed a multi-label lazy algorithm based on the data gravitation model, named the MLDGC. It is the first algorithm developed for data gravitation model on multi-label classification. However, the data gravitation-based methods often ignore the interaction of data mass, and directly define data gravitation with density, resulting in inconsistency with human perception. In view of the necessity to include the attraction between two particles in the data gravitation-based classification, this paper proposes a novel algorithm based on gravitation theory called the interaction and data gravitation-based model for multi-label classification (ITDGM).

The rest of this paper is organized as follows: Section 2 briefly introduces gravitation and reviews the related research on data gravitation, Section 3 presents the proposed algorithm, Section 4 describes the experiment, Section 5 shows and discusses the experimental results, and Section 6 wraps up the research with some conclusions.

2 Literature review

Whenever there is an interaction between two objects, there is a force upon each of the objects. One of the forces can be characterized by Newton's law of universal gravitation. The universal gravitation-based methods are widely used in classification or clustering.

Wright [18] was the first to apply such a method in cluster analysis. Gomez et al. [6] proposed a clustering algorithm based on feature space that considers each of the instances. On account of gravitational collapse, Wang et al. [19] put forward an improved algorithm based on k-NN classifier. In light of the theoretical model of data gravitation and data gravitation field, Yang et al. [22] presented a new classification model called data gravitation-based classifier (DGC), and used it in an intrusion detection system (IDS). Li et al. [9] developed a nonlinear classification algorithm to handle data gravity as data vector. Shafiq et al. [16] proposed a method based on the gravitational potential energy of particles named gravitation-based classification (GBC) algorithm, sought for the equilibrium condition of the classifier, and proved that the algorithm had strong robustness but high computational complexity. Wena et al. [20] came up with the cognitive gravitation model (CGM) algorithm, which uses the self-information of each instance instead of the mass based on the law of gravitation and the cognitive laws. Starting from the DGC method, Peng et al. [12] created the improved DGC (IDGC) in an attempt to handle imbalance datasets with amplified gravitation coefficient (AGC), which could strengthen and weaken the gravitational field of the minority and majority instances. Reyes et al. proposed an algorithm based on lazy learning and data gravitation model for classification which simultaneously measured classification accuracy and label ranking.

Among the aforementioned data gravitation-based algorithms, most of them are applied in single-label learning systems. The only exception is the MLDGC, which is adopted for multi-label classification. Overall, few scholars have integrated the data gravitation model and multi-label classification.

3 The proposed algorithm based on gravitation theory

So far, the laws of physics have been extensively applied to the information domain. It is stressed that the interaction of data instances must conform to the physical laws. The laws are implemented to simulate actual data classification in many novel data gravitation-based classification algorithms. However, the simulation results may not fully correspond with the reality as the data gravitation is directly defined with the density instead of the interaction between data instances.

To solve the problem, this paper proposes a multi-label learning algorithm based on data gravitation model (ITDGM). In a multi-label dataset, an instance i can be transformed into a data particle expressed as (X_i, Y_i, m_i) . The feature vector of the data particle i is defined as the label set of the particle and denoted by X_i and Y_i .

The symbol m_i means the data mass of particle i . For any two particles i and j , the distance between them can be calculated as

$$d_F(i, j) = \sqrt{\sum_{\forall f \in F} \delta(x_{if}, x_{jf})^2}.$$

Since the Heterogeneous Euclidean-Overlap Metric (HEOM) can handle both nominal and continuous features, the above heterogeneous distance function is adopted in the experiment.

Therefore, the term $\delta(x_{if}, x_{jf})^2$ should be solved under three different conditions:

- $\delta(x_{if}, x_{jf})^2 = 1$ when $x_{if} \neq x_{jf}$ and the data attribute is discrete;
- $\delta(x_{if}, x_{jf})^2 = 0$ when $x_{if} = x_{jf}$ and the data attribute is discrete;
- $\delta(x_{if}, x_{jf})^2 = \frac{|x_{if} - x_{jf}|}{\max(f) - \min(f)}$ when the data attribute is continuous;

Where x_{if} and x_{jf} are the f -th eigenvalues of i and j , respectively; F is the feature space; $\delta(x_{if}, x_{jf})$ is the type of f -th feature; $\max(f)$ and $\min(f)$ are the maximum and minimum values, respectively. Hence, it is possible to identify the k -nearest neighbors of particle i by the $d_F(i, j)$ function. The k -nearest neighbors of particle i can calculate by $N_i = (i_1, i_2, \dots, i_k) | d_F(i, i_1), d_F(i, i_2), \dots, d_F(i, i_k)$, which is sorted by ascending order.

According to the Newton's law of universal gravitation, the greater the mass of two objects, the greater the resulting interaction between them. In other words, the gravitation increases with the interaction between the two objects. In this research, a new concept is defined: the interaction-based gravitation coefficient (IGC). The coefficient is created to evaluate the attraction between two particles like the interaction between two objects. Data has not mass. For the reason that data has no mass, the data mass is usually set as one. However, the setting is inconsistent with human perception without considering the interactions between data particles. Thus, the author defines the IGC and a data gravitation-based model to measure the strong or weak gravitational force of particles. The dissimilarity between the label set of the particle i and that of particle j is expressed by the operator $\Theta(Y_i, Y_j)$ and the interaction between the two particles $M(x_{i,j})$ is defined as:

$$P(i, j) = \frac{\Theta(Y_i, Y_j)}{q} \times \frac{1}{f_j} \quad (1)$$

$$M(x_{i,j}) = \log\left(\frac{1}{P(i, j)}\right) = \log\left(\frac{qf_j}{\Theta(Y_i, Y_j)}\right) \quad (2)$$

Where $P(i, j)$ is the dissimilarity degree between the label set of the particle i and that of particle j , and f_j is the number of the instance labels, and q is the number of the total labels.

For the sake of data stability, $M(x_{i,j})$ is introduced to measure the dissimilarity in the classification between particles. From the above formulas, it is concluded that $P(i,j)$ is inversely proportional to $M(x_{i,j})$, that is, the smaller the $P(i,j)$, the more intense the interaction between the particle i and j .

Based on neighborhood density concept of Reyes et al. [14], the IGC value of particle i is calculated by the formula $m_i = d_i^{w_i}$, where d_i is the neighborhood density or the distribution of the neighbor particles of i , and w_i is the neighborhood weight of particle i . The greater the value of d_i , the more similarity between the label set of another particle and that of particle i . The neighborhood density is computed by the following formula:

$$d_i = 1 + \sum_{j \in N_i} (1 + M(x_{i,j})) / d_F(i,j) \quad (3)$$

Based on the above formula, the neighborhood weight calculation method is improved as below. First, $p_{dY}^i = \frac{\sum_{j \in N_i} I(x_{i,j})}{k}$ and $p_{dF}^i = \frac{\sum_{j \in N_i} d_F(i,j)}{k}$ are identified as the nearest particles of i based on the principle of prior probability and the prior probability of the nearest particles of i in the feature space, respectively. The symbol k is the initial number of nearest neighbors, and N_i is the nearest neighbors set of the particle i . Then, the prior probability of the nearest neighborhood of i with similar label sets can be calculated by $p_{dY|dF}^i = \frac{\sum_{j \in N_i} I(x_{i,j}) \cdot d_F(i,j)}{k}$. Finally, the neighborhood weight (w_i) is obtained by the formula $w_i = \frac{p_{dY|dF}^i p_{dF}^i}{p_{dY}^i} - \frac{(1 - p_{dY|dF}^i) p_{dF}^i}{1 - p_{dY}^i}$.

Through the above steps, it is possible to get the gravitational force between two instances. Taking an instance from dataset, the k -nearest neighborhood of particle i can be computed as below.

$$F(i,j) = g \frac{IGC}{d_F(i,j)^2} = g \frac{m_i}{d_F(i,j)^2} \quad (4)$$

In this experiment, the gravitational constant g is set to 1 for the calculation of the gravitation of instances. Thus, the gravitational force $f(i,j) = g \frac{m_i m_j}{r^2}$ is rewritten as $F(i,j) = \frac{IGC}{d_F(i,j)^2}$, where the IGC replaces the product of the mass of the objects.

After obtaining the gravitational force between instances, it is necessary to determine which labels belong to the test instances. To this end, a simple statistical approach is implemented. The number of the y -th label that belongs to the k -nearest particles is expressed as $n(i) = \sum_{j \in F_i} F(i,j)$, $y \in Y_j$, and the number of the y -th label that does not belong to the k -nearest particles is expressed as $n'(i) = \sum_{j \in F_i} F(i,j)$, $y \notin Y_j$, F_i is the set of the nearest neighbors. If $n(i) > n'(i)$, the y -th label belongs to the instance; otherwise, the y -th label does not belong to the instance.

The proposed method is divided into three main steps. First, calculate the interaction between instances with the IGC coefficient; second, compute the gravitation of each particle i ; third, obtain the k -nearest neighbors of the instance based on gravitation. Figure 1 illustrates the basic procedure of the ITDGM method.

The proposed algorithm seeks for k -nearest neighbors of each multi-label instance, and aims to transform the instances to particles in the learning phase. If linear search strategy is adopted for the k -nearest neighbors of instance i , the optimal time complexity of the learning phase is $O(n^2 \cdot d)$ for all training instances, i.e. it takes $O(n \cdot k \cdot d)$ steps at most to classify k instances, where n is the number of instances and d is the cardinality of the feature space.

Algorithm 1 The ITDGM algorithm

Input: $T_r \rightarrow$ multi-label training dataset; $T_s \rightarrow$ multi-label testing dataset; $k \rightarrow$ number of nearest neighbours

Output: label set of instances

Begin

```

1: Learning phase
2: for each  $i \in T_r$  do
3:    $N_i \leftarrow k$  Nearest neighbors ( $i, T_r, k$ )
4:    $m_i \leftarrow IGC(i, N_i)$ 
5: end
6: Test phase
7: for each  $i \in T_s$  do
8:    $F_i \leftarrow k$  Nearest neighbors ( $i, T_s, k$ )
9:    $Y_i \leftarrow \emptyset$ 
10:  For each  $y \in Y$  do
11:     $n(i) = \sum_{j \in F_i} F(i, j), y \in Y_j$ 
12:     $n'(i) = \sum_{j \in F_i} F(i, j), y \notin Y_j$ 
13:  If  $n(i) > n'(i)$  then
14:     $Y_i \leftarrow Y_i \cup y;$ 
15: end

```

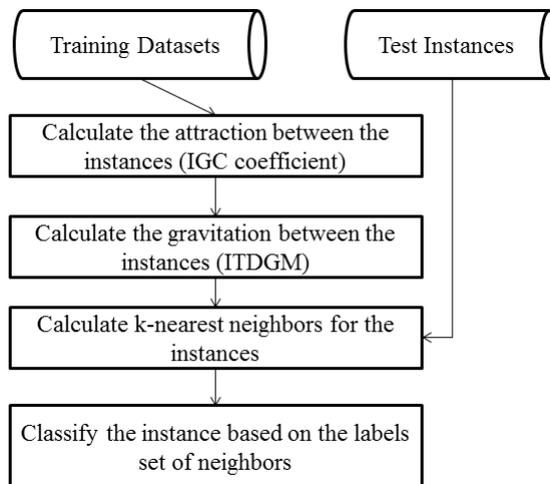


Figure 1: Basic procedure of the ITDGM method

4 Experiment

Due to the one-to-many relationship between instances and labels, the evaluation of multi-label classification performance is much more complicated than the traditional supervised single-label learning. The existing evaluation measures for multi-label classification mainly fall into two groups: the example-based measure [7] and the label-based measure [25]. In this research, the experimental results are verified by a string of different public datasets and discussed in details below.

4.1 Datasets

In this paper, five public multi-label datasets of different scales, labels and properties are selected from various domains to validate the proposed algorithm. The ‘‘Genbase’’ datasets [3] store information on the gene functions; the ‘‘Birds’’ contain bird species examples for acoustic classification; Corel16k001, Corel16k002 and Corel16k006 encompass examples of Corel images. The statistics on benchmark datasets are shown in Table 1, where n is the number of instances in the datasets; d is the number of features; q is the number of labels; d_s and l_c are the label cardinality and the label density, respectively.

Table 1: Statistics of the public multi-label benchmark datasets

Dataset	Domain	n	d	q	d_s	l_c
Genbase	biology	662	1186	27	32	1.252
Birds	audio	645	260	19	133	1.014
Corel16k001	image	13766	500	653	4937	2.867
Corel16k002	image	13761	500	654	4937	2.867
Corel16k006	image	13859	500	652	4937	2.867

4.2 Evaluation metrics

The performance of multi-label classification is measured differently from the single-label classification learning system. Let $X \in R_d$ be instances of data sets and express each instance as a d -dimensional vector $x = \{x_1, x_2, \dots, x_d\}, x \in X$. Let $Y = \{l_1, l_2, \dots, l_m\}$ be a finite label set. Every instance belongs to a subset of Y and the subset is represented by an m -dimensional vector $y = \{y_1, y_2, \dots, y_m\}$; if $y_k = 1$, label l_k belongs to instance x ; if $y_k = 0$, the label does not belong to the instance. For a given training dataset $T = \{(x_i, y_i) | 1 \leq i \leq n, x_i \in X, y_i \in Y\}$ composed of n instances, where x is the instance and y is the label subset of the instance, there is a corresponding test dataset $D = \{(x_i, y_i) | 1 \leq i \leq p, x_i \in X, y_i \in Y\}$ composed of p instances based on the evaluation of the classification performance on test instances and the test set that returns the mean value.

In this paper, the instance-based F1-measure ($F1_{Ex}$) is implemented to present the experimental evaluation results:

$$F1_{Ex} = \frac{1}{m} \sum_{i=1}^m \frac{2|Y_i \cap Z_i|}{|Y_i| + |Z_i|} \quad (5)$$

Where m is the number of test instances, and Z_i is the predicted label set of a given test instance i . The greater the value of $F1_{Ex}$, the better the performance of the multi-label classification algorithm.

Different from the above example-based measure, the label based micro F1-measure relies on the harmonic mean between recall and precision. The precision refers to the relevance of the predicted labels and the recall reflects the percentage of the predicted labels that are relevant. Such a measure is used in this research to present the algorithm performance. The F1-measure and the micro averaging are expressed by the following formulas:

$$F1 - \text{measure} = \frac{1}{N} \sum_{i=1}^N \frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \quad (6)$$

$$M_{micro} = M \left(\sum_{l=1}^M tp_l, \sum_{l=1}^M fp_l, \sum_{l=1}^M tn_l, \sum_{l=1}^M fn_l \right) \quad (7)$$

Where tp_l is the true positives; fp_l is the false positives; tn_l and fn_l are the true and false negatives for l labels after a binary evaluation, separately. The greater the micro F1-measure, the better the classification performance of the multi-label algorithm.

4.3 Experiment setting

The performance of the proposed algorithm ITDGM is compared with that of five classic multi-label classification algorithms: the ML-kNN, the BR, the RAKEL, the BRkNN and the BPMLL. In the RAKEL and BR algorithms, the C4.5 decision tree is taken as the classifier. In this research, the number of the nearest neighbors is 2~7 for the ML-kNN, the ITDGM and the BRkNN; the smoothing factor of the ML-kNN is 1.0; the learning rate parameter of the BPMLL is 0.05; the Epoch parameter is 100; 20% of the input units are hidden units. The experiments are conducted on an Intel Core-i5 2.3 GHz processor with 8G memory, with the aid of MATLAB 2012 and the Java-based open resource software MULAN, which contains lots of classic and popular multi-label algorithms, evaluation targets and measures. The results are subjected to 5-fold cross validation.

5 Results and discussion

In this research, the example-based F1-measure ($F1_{Ex}$) and micro F1-measure are taken as the multi-label classifier evaluation criteria. In any case, the best results are highlighted and bolded. As mentioned above, the greater the $F1_{Ex}$ and micro F1-measure, the better the performance of the corresponding evaluation method.

Tables 2~3 show the results of multi-label classification algorithms on test datasets for $F1_{Ex}$ and micro F1-measure. It is seen that the proposed algorithm ITDGM has better performance than the other methods.

Based on the $F1_{Ex}$ values in Table 2, the ITDGM outperforms the other five classic methods on the five public datasets. The $F1_{Ex}$ index of the proposed algorithm surpasses that of the other algorithms more than 3% on the "Bird" dataset, and 2% on the remaining datasets.

Table 3 displays the micro F1-measure values of the proposed algorithm and the other algorithms. The results show that the proposed algorithm boasts the best performance on the Corel16k001, Corel16k002 and Corel16k006 datasets, and keeps a 2% lead over the other algorithms in "Genbase" and "Birds" datasets.

Table 2: Example-based F1-measure ($F1_{Ex}$) values of all algorithms on the five public datasets

Algorithm	Dataset				
	Genbase	Birds	Corel16k001	Corel16k002	Corel16k006
MLKNN	0.9649±0.0099	0.5330±0.0436	0.0071±0.0027	0.0063±0.0042	0.0109±0.0039
RAKEL	0.9705±0.0169	0.5608±0.0334	0.0459±0.0093	0.0739±0.0059	0.0798±0.0070
BRKNN	0.9565±0.0090	0.5649±0.0385	0.0410±0.0105	0.0522±0.0096	0.0567±0.0071
BPMLL	0.0772±0.0457	0.4436±0.0594	0.0289±0.0011	0.0287±0.0026	0.0282±0.0017
BR	0.9705±0.0169	0.5608±0.0334	0.0535±0.0054	0.0849±0.0088	0.0798±0.0070
ITDGM	0.9766±0.0121	0.5661±0.0544	0.0851±0.0076	0.0946±0.0125	0.0995±0.0066

Figures 2~11 show the relationship between the values of $F1_{Ex}$ and micro F1-measure and the number of nearest neighbors in all algorithms. It is demonstrated that the proposed algorithm

Table 3: Micro F1-measure values of all algorithms on the five public datasets

Algorithm	Dataset				
	Genbase	Birds	Corel16k001	Corel16k002	Corel16k006
MLKNN	0.9447±0.0142	0.2728±0.0300	0.0129±0.0050	0.0094±0.0056	0.0134±0.0039
RAkEL	0.9625±0.0094	0.3456±0.0445	0.0606±0.0062	0.1000±0.0054	0.1077±0.0109
BRKNN	0.9446±0.0123	0.3972±0.0303	0.0629±0.0144	0.0717±0.0087	0.0787±0.0044
BPMLL	0.0870±0.0472	0.1528±0.0518	0.0414±0.0003	0.0408±0.0008	0.0396±0.0008
BR	0.9625±0.0094	0.3456±0.0445	0.0718±0.0066	0.1108±0.0081	0.1077±0.0109
ITDGM	0.9686±0.0118	0.4314±0.0571	0.1066±0.0082	0.1138±0.0101	0.1195±0.0065

has the best classification effect with the fewest k-nearest neighbors in the testing datasets. In Figures 2 and 3, all algorithms perform well except the BPMLL ($F1_{Ex}$: 0.0772, micro F1: 0.0870). The ITDGM achieves the best overall result (micro F1: 0.9686) in Figure 3, followed closely by the simply transformation methods the RAkEL and the BR (micro F1: 0.9625).

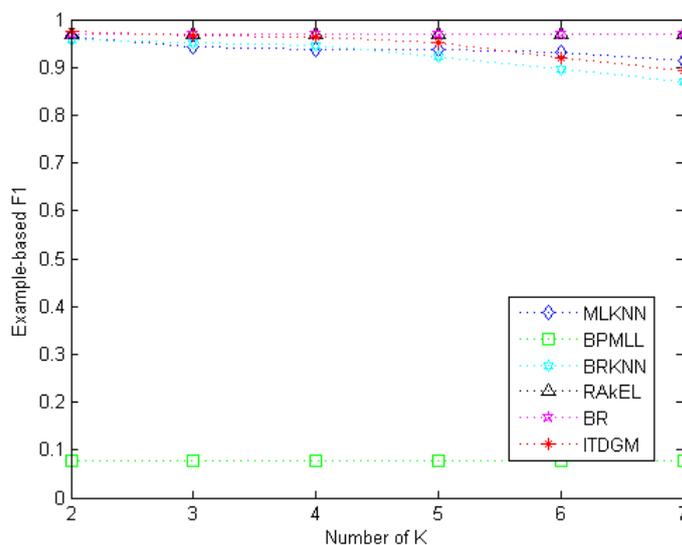


Figure 2: Relationship between $F1_{Ex}$ and the number of nearest neighbors in “Genbase” of all algorithms

The ITDGM is still the best performing algorithm in Figures 4 and 5. Although the ML-kNN and the BRkNN achieve fairly good performance at $k=2$ ($F1_{Ex}$: 0.5330) and $k=3$ ($F1_{Ex}$: 0.5649), respectively, the ITDGM has a slight edge over the two algorithms at $k=2$. As shown in Figure 5, the proposed algorithm has the best performance at almost any number of nearest neighbors on micro F1 metrics. According to Figure 4 and Figure 5, the RAkEL algorithm also has an outstanding performance. This is attributable to the random selection of label sets and unique strategies for training models in the algorithm.

Figures 6~11 describe the metrics of all six algorithms on three Corel16k datasets. All selective multi-label classification methods perform poorly in these datasets, owing to the low label density and huge amount of distinctive label sets in the three datasets. In this case, the proposed algorithm also outperforms the other algorithms.

On the contrary, the ML-kNN shows the worst performance in the three datasets due to the capacity of its classifier. Despite the good performance in some cases, such as Corel16k001 ($F1_{Ex}$:

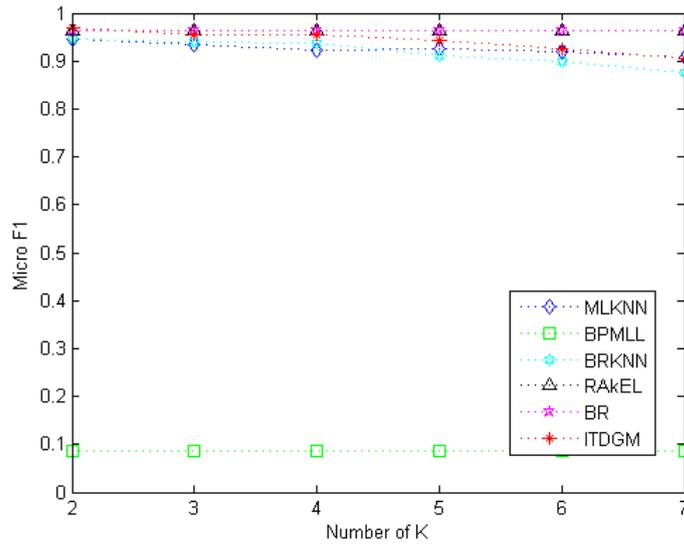


Figure 3: Relationship between micro F1 and the number of nearest neighbors in “Genbase” of all algorithms

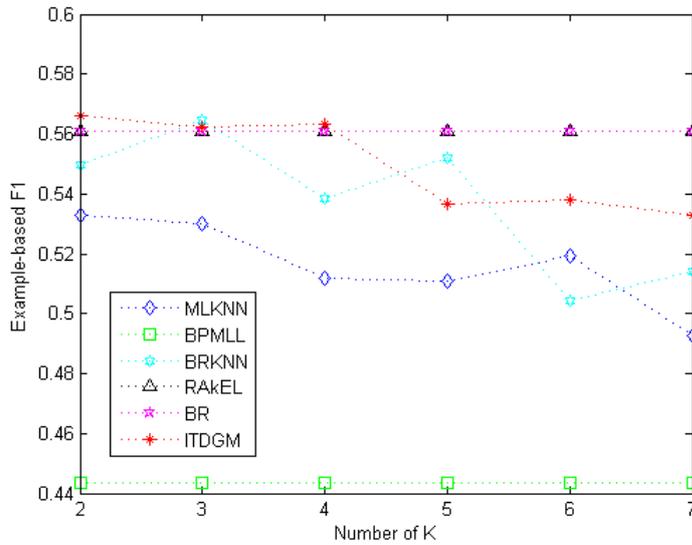


Figure 4: Relationship between $F1_{Ex}$ and the number of nearest neighbors in “Birds” of all algorithms

0.0535, micro F1: 0.0718), Corel16k002 ($F1_{Ex}$: 0.0849, micro F1: 0.1108), Corel16k006 ($F1_{Ex}$: 0.0798, micro F1: 0.1077), the simple transformation method BR costs much more training time than other algorithms. In comparison, the ITDGM, the BRkNN and the ML-kNN consumes less training time thanks to their roots in the lazy learning system.

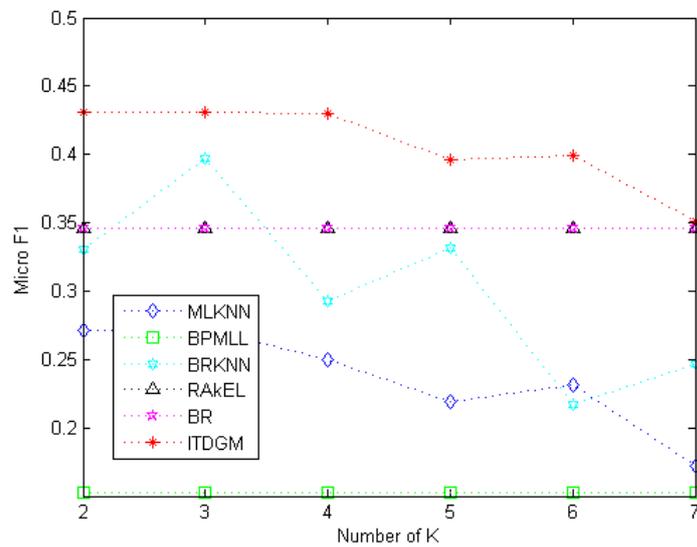


Figure 5: Relationship between micro F1 and the number of nearest neighbors in "Birds" of all algorithms

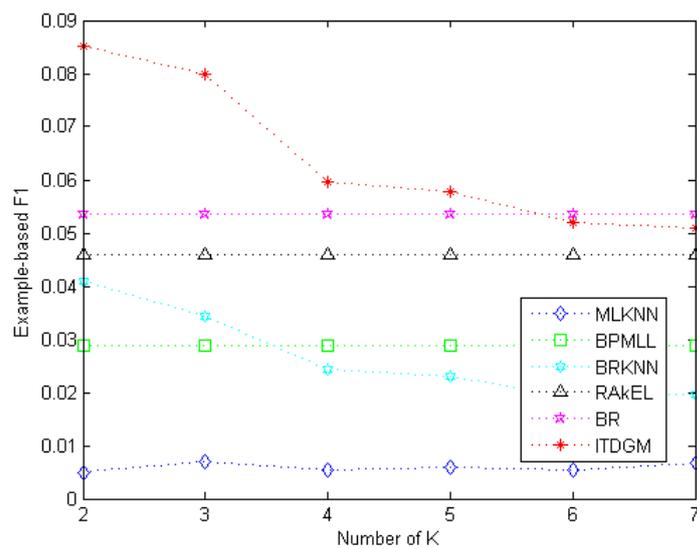


Figure 6: Relationship between $F1_{Ex}$ and the number of nearest neighbors in Corel16k001 of all algorithms

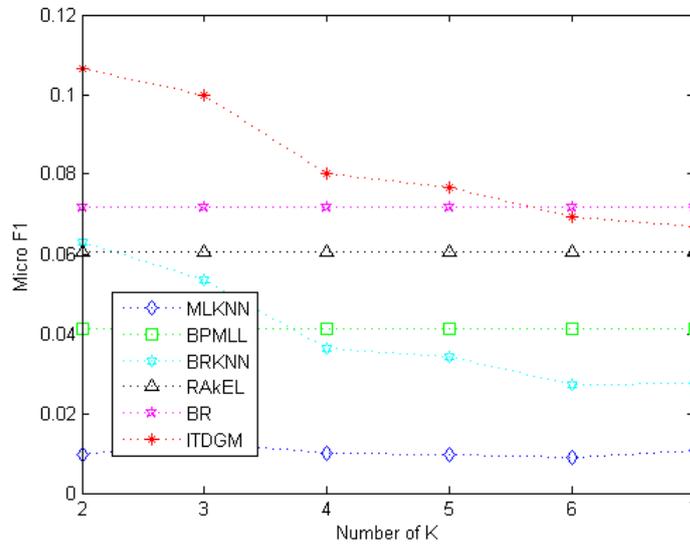


Figure 7: Relationship between micro F1 and the number of nearest neighbors in Corel16k001 of all algorithms

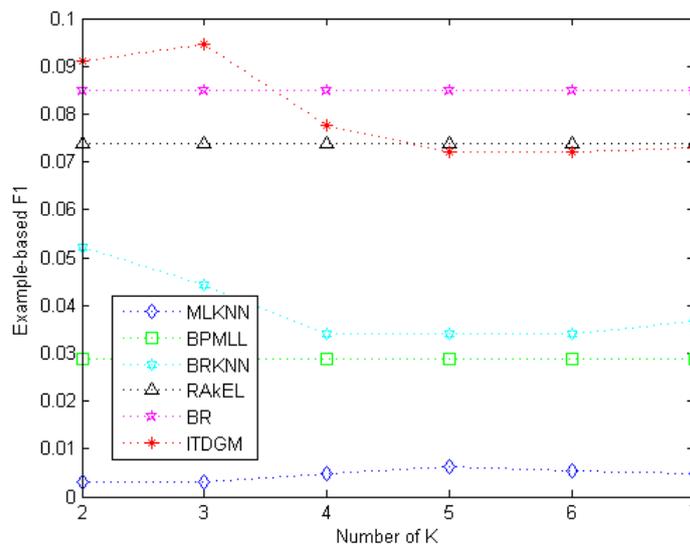


Figure 8: Relationship between $F1_{Ex}$ and the number of nearest neighbors in Corel16k002 of all algorithms

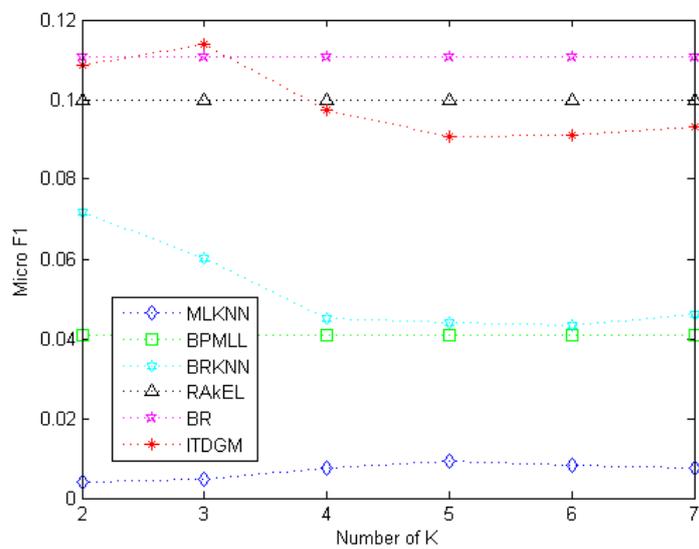


Figure 9: Relationship between micro F1 and the number of nearest neighbors in Corel16k002 of all algorithms

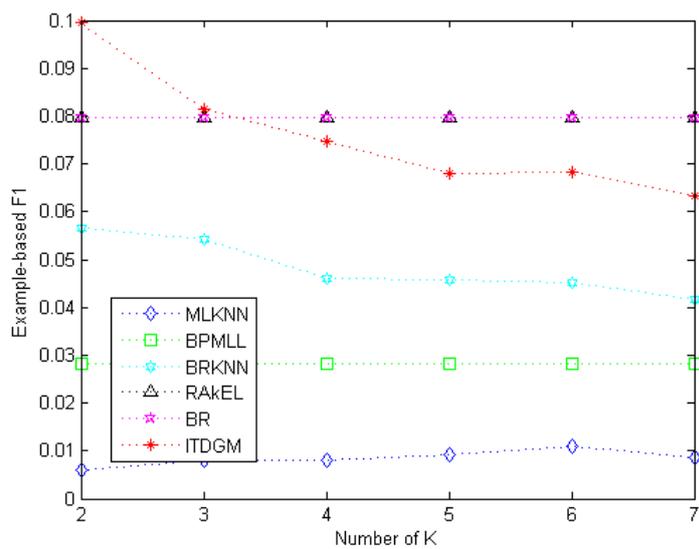


Figure 10: Relationship between $F1_{Ex}$ and the number of nearest neighbors in Corel16k006 of all algorithms

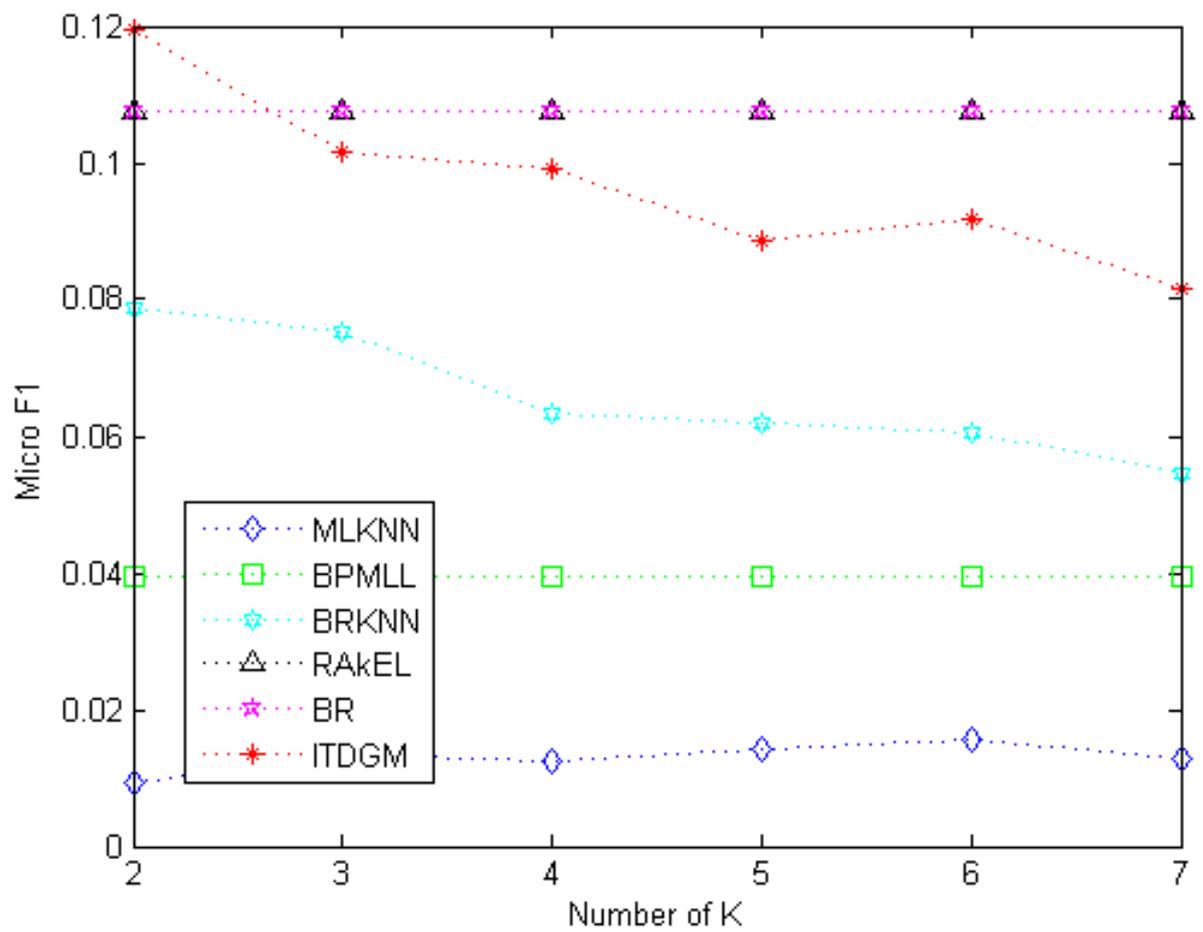


Figure 11: Relationship between micro F1 and the number of nearest neighbors in Corel16k006 of all algorithms

6 Conclusion

This paper presents a new multi-label classification algorithm based on data gravitation theory, introduces the interaction-based gravitation coefficient (IGC) to the proposed algorithm, and uses the algorithm to calculate the gravitational force instead of the mass of the particles. The average classification accuracy of the proposed method is evaluated by experiments on five public open datasets, in comparison with five classic multi-label algorithms: the BR, the RAKEL, the ML-kNN, the BPMLL and the BRkNN. The six algorithms are applied, under the example-based F1-measure ($F1_{Ex}$) and micro F1-measure, to the five public multi-label datasets at different number of nearest neighbors. In the experiment, the proposed algorithm outperforms all the other five algorithms, showcasing ideal capability and rationality. This research creates several new topics for future studies: the proposal of new multi-label classification method based on data gravitation model, integration with other fields [5] [10], and the improvement of the classification effect.

Acknowledgement

This research was supported in part by the National Natural Science Foundation of China (NSFC) under grants 60903074, the National High Technology Research and Development Program of China (863 Program) under grant 2008AA01Z119.

Bibliography

- [1] Boser B.E., Guyon I.M., Vapnik V.N. (1992); A training algorithm for optimal margin classifiers, *Proce. of the 5th Annual ACM Conf. on Computational Learning Theory*, 144–152, 1992.
- [2] Boutell M.R., Luo J., Shen X., Brown C.M. (2004); Learning multi-label scene classification, *Pattern Recognition*, 37(9), 1757–1771, 2004.
- [3] Diplaris S., Tsoumakas G., Mitkas P., Vlahavas I. (2005); Protein classification with multiple algorithms, *Proc. of the 10th Panhellenic Conference on Informatics*, 3746, 448–456, 2005.
- [4] Freund Y. (1995); Boosting a weak learning algorithm by majority, *Information and Computation*, 121(2), 256–285, 1995.
- [5] Fichera A., Fortuna L., Frasca M., Volpe R. (2015); Integration of complex networks for urban energy mapping, *International Journal of Heat and Technology*, 33(4), 181–184, 2015.
- [6] Gomez J., Dasgupta D., Nasraoui O. (2013); A new gravitational clustering algorithm, *Proc. of the SIAM Intl. Conf. on Data Mining*, 2013.
- [7] Ghamrawi N., McCallum A. (2005); Collective multi-label classification, *Proc. of the 14th ACM Intl. Conf. Information and Knowledge Management*, 195–200, 2005.
- [8] Lu H., Rudy S., Liu H. (1996); Effect data mining using neural networks, *IEEE Transaction on Knowledge and Data Engineering*, 8(6), 957–961, 1996.
- [9] Li J., Fu H. (2009); Data classification based on supporting data gravity. *Proc. of the IEEE Intl. Conf. on Intelligent Computing and Intelligent Systems*, 1, 22–28, 2009.

-
- [10] Lin T., Wu P., Gao F.G., Yu Y., Wang L.H. (2015); Study on SVM temperature compensation of liquid ammonia volumetric flowmeter based on variable weight PSO, *International Journal of Heat and Technology*, 33(2), 151–156, 2015.
- [11] Pawlak A. (1991); *Rough Sets: Theoretical Aspects of Reasoning about Data*, Dordrecht, 1991.
- [12] Peng L., Zhang H., Yang B., Chen Y. (2014); A new approach for imbalanced data classification based on data gravitation, *Information Sciences*, 288(C), 347–373, 2014.
- [13] Read J., Pfahringer B., Holmes G., Frank E. (2009); Classifier chains for multi-label classification, *Machine Learning and Knowledge Discovery in Databases*, 254–269, 2009.
- [14] Reyes O., Morell C., Ventura S. (2016); Effective lazy learning algorithm based on a data gravitation model for multi-label learning, *Information Sciences*, 2016, s340–341, 159–174, 2016.
- [15] Spyromitros E., Tsoumakas G., Vlahavas I. (2008); An empirical study of lazy multilabel classification algorithms, *Proc. of the 5th Hellenic Conference on Artificial Intelligence*, 401–406, 2008.
- [16] Shafigh P., Hadi S.Y., Sohrab E. (2013); Gravitation based classification. *Information Sciences*, 220(1), 319–330, 2013.
- [17] Tsoumakas G., Vlahavas I. (2007); Random k-Labelsets: An ensemble method for multilabel classification, *Machine learning: ECML 2007*, 406–417, 2007.
- [18] Wright W. E. (1977); Gravitational clustering, *Pattern Recognition*, 9(3), 151–166, 1977.
- [19] Wang C., Chen Y.Q. (2005); Improving nearest neighbor classification with simulated gravitational collapse, *Proc. of the Intl. Conf. on Advances in Natural Computation*, 3612, 845–854, 2005.
- [20] Wena G., Wei J., Wang J., Zhou T., Chen L. (2013); Cognitive gravitation model for classification on small noisy data, *Neurocomputing*, 118(11), 245–252, 2013.
- [21] Younes Z., Abdallah F., Denceux T. (2008); Multi-label classification algorithm derived from k-nearest neighbor rule with label dependencies, *Proc. of the 16th European Signal Processing Conference*, 297–308, 2008.
- [22] Yang B., Peng L., Chen Y., Liu H., Yuan R. (2006); A DGC-based data classification method used for abnormal network intrusion detection, *Proc. of the Intl. Conf. on Neural Information Processing*, 4234, 209–216, 2006
- [23] Zhang M.L., Zhou Z.H. (2006); Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10), 1338–1351, 2006.
- [24] Zhang M.L., Zhou Z.H. (2007); ML-KNN: A lazy learning approach to multi-label learning, *Pattern Recognition*, 40(7), 2038–2048, 2007.
- [25] Zhang M.L., Zhou Z.H. (2014); A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8), 1819–1837, 2014.

Delay Tolerant Networks over Near Field Communications: The Automatic Multi-packet Communication

R.O. Schoeneich, P. Sadło

Radosław Olgierd Schoeneich*

Institute of Telecommunications

Warsaw University of Technology, Poland

*Corresponding author: rschoeneich@tele.pw.edu.pl

Piotr Sadło

Institute of Informatics

Warsaw University of Technology, Poland

p.sadlo@stud.elka.pw.edu.pl

Abstract: The Near Field Communication (NFC) is designed for sending small data, which size does not exceed the several dozen bytes, for distance not higher than two inches. The NFC can be used in store-carry-forward scenarios which are bases for Disruptive Tolerant Networks (DTN). The aim of this paper is proposal how to use NFC in transmission of large data in DTN. We propose an algorithm of fragmenting, sending and receiving data. We described software which was used in our prototype and discussed about implementation. Finally we showed results of tests performed on our model. Obtained results presents well characteristics of our solution.

Keywords: mobile communication, radio communication, wireless networks, personal communication networks, cooperative communication.

1 Introduction

Nowadays, an attention of scientists working on ad-hoc networks is focused on Delay and Disruptive Tolerant Networks (DTN). The DTN is a specific network in which nodes are mobile, therefore the network topology is often changeable what can be reason of lack of end-to-end path between source and destination nodes. The communication of DTN is done by storing-carrying and forwarding messages.

Applications of DTN are specific: previously it was proposed for interplanetary communication [1], but now it is used in military [14], search and rescue [15], wildlife monitoring [10], body sensing [17] and in many other solutions [21] [13].

The DTN transmission is based on wireless interfaces which are usually long range like radio satellite communications and medium range e.g. WiFi and Bluetooth standards. One of the possible communication interface, which fulfills DTN requirements of wireless medium and easy applicability of store-carry-forward paradigm is a very short range - Near Field Communication (NFC) [24].

NFC is used to transfer small packets on a very short-range. Transmission throughput is very low. This standard is widely used for cashless payment and short tag exchange. It is applicable not only on cards and tags but also on small mobile devices e.g. smartphones. Unlike other wireless communication solutions, NFC is able to setup sessions between devices automatically. The usage of NFC can be effective implemented in the DTN environment. Automatic and secret communication ability is advantage which supports the implementation of DTN objectives.

The aim of this paper is to present solution how to use the NFC to communicate DTN devices using store-carry-forward without data size limitation. For this purpose, we proposed a simple way of communication with the automatic exchange data between devices. We designed and implemented solution how to apply NFC without size limit of the transferred data.

The rest of this paper looks as follows: in section 2 we presented the state of the art: we discuss NFC technology, basics of DTN, and other telecommunication issues. In section 3 we present the algorithm of automatic establishment communication session and a mechanism for transferring big packets of data. In the section 4 we present results of tests performed on our prototype. Chapter 5 contains a conclusions.

2 State of the art

The Near Field Communication is a radio communication standard, based on RFID technology (Radio-Frequency Identification). It allows to wirelessly transfer data on a distance up to 20 cm [2]. In practice, the exchange of information over distances is smaller than 4 cm.

NFC can operate in two modes: active and passive. In the active mode, device alternatively communicate with each other. Both devices need power energy. The device which is at standby state does not emit signals. In passive mode, only one device can initiate the connection. Passive devices cannot read data by themselves; however, they can store it. NFC applications are focused on widely used wireless cashless payment. Contactless payment [16] is done by cards and smartphones. Other area of NFC usage are applications working with NFC tags with web addresses [23], tags for indoor navigation or information and wide aspects of smart homes e.g. office and home secure, body area communications [25]. All of this applications are using very short NFC messages and no one extends the basic communication functionality.

Delay and Disruptive Tolerant Networks firstly was proposed by K. Fall [7] for interplanetary communication. The basic idea is to provide communications in scenarios in which traditional telecommunication approach does not work. The DTN networks evaluate based on wireless ad-hoc and sensor network (MANET and WSN) approach: every node acts as a part of network and can route a message. Based on this for MANET and WSN there was proposed several routing protocols [20] [19] and algorithms which supports data handling [8] [9] [12] [5] [3] and transmission [22]. The DTN adds the new assumption to the Ad-hoc approach: store-carry-forward messages. Storing and forwarding messages is done by nodes called message ferries or mules. As a message ferry can act mobile device associated with human [4], animal or mobile machine [6]. By analogy to MANET, DTN requires specific solutions for routing and data handling [11] [18].

All this solutions are working with popular communication standards like WiFi and Bluetooth. As authors know none of these solution is designed for very short range solution like NFC.

3 DTN over NFC - MultiPacket communication

This section describes algorithms for sending and automatic receiving large data. Algorithms are implemented on Android environment. We propose automatic solution that splits large data into elementary packets and avoids "touch to send" restriction that is mandatory in Android NFC API.

3.1 Message fragmentation into NFC packet

Each message before sending must be adopted to the size of NFC packet. It means, the fragmentation process must be taken. Afterwards, NFC session is set up between devices (nodes) and the packets in sequence are sent to the receiver. Sequence of steps is shown in Figure 1.

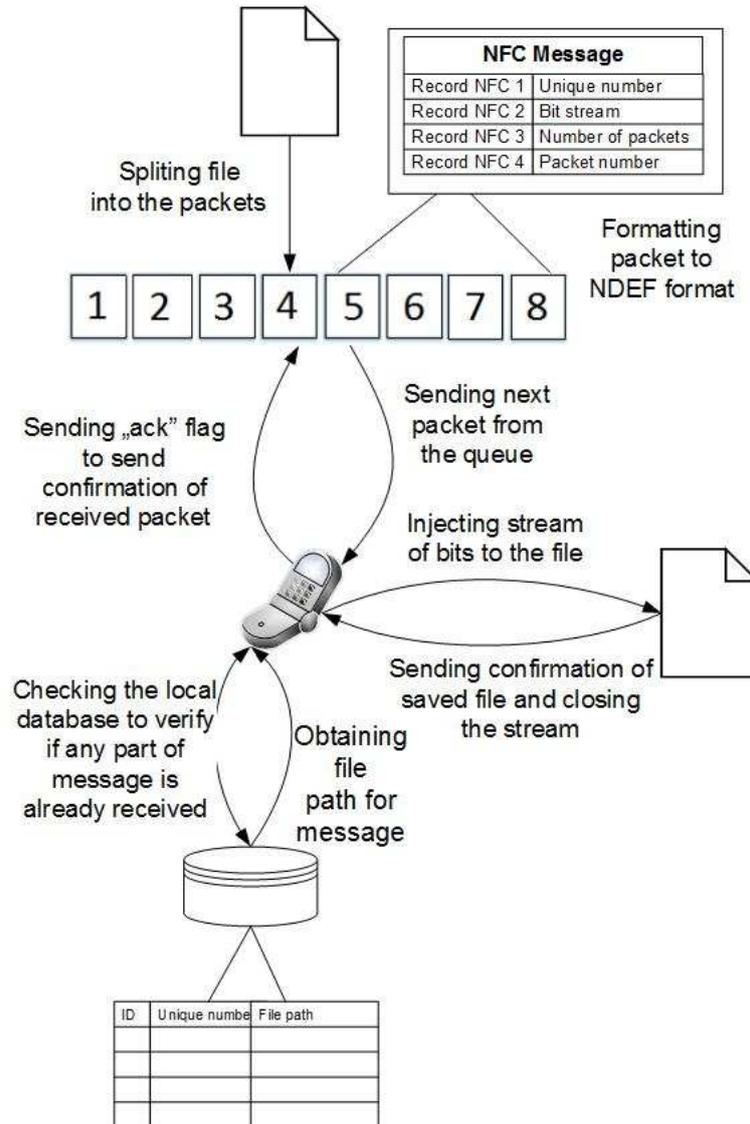


Figure 1: Sending process

Table 1: The detailed description of each record

<i>Item</i>	<i>Description</i>
Unique number	Unique number for the message
Bitstream	Payload
Number of packets	Number of parts generated by splitting the message
Packet number	Succeeding number of part

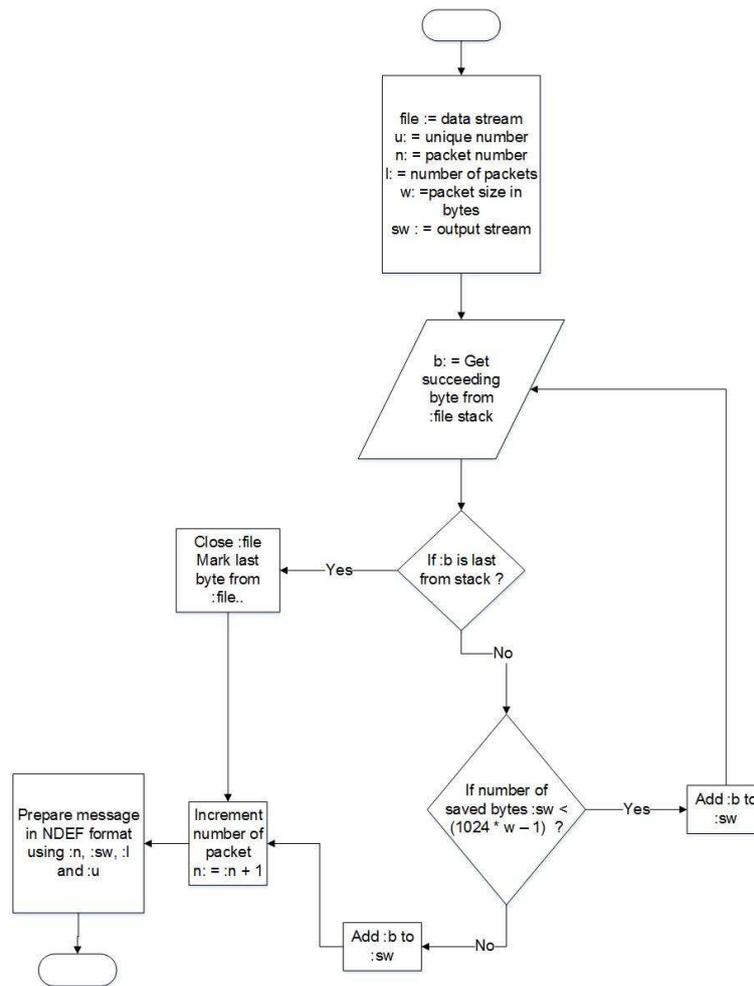


Figure 2: Block diagram of the algorithm responsible for sending the NFC message

In the example the message is divided into 8 fragments. Each fragment is built from four NFC records, which are saved in NFC data exchange format (NDEF). Detailed description of each record is presented in Table 1.

The receiver analyzes each record of the received packet. The unique number is used to check in the local database if exists any received data of the message. The payload is appended to this data in case of existance another data of the message. After this operation the receiver sends the confirmation flag to inform the sender that the packet has been received and processed succesfully. The sender takes the next packet from the stack and sends it to the receiver. This process is repeated until the stack is empty.

3.2 Data sending

Besides specifying a unique part number and payload, the algorithm determines packet number and calculate the total number of packets of the message. The number of packets is calculated by dividing the size of the message by the size of the single packet.

The message is passed to the algorithm responsible for sending data as an input. Additional parameters are: an unique number for transmission which was generated in the previous step, the number of packets to send, the number of the current being sent packet, and the defined packet size. The message is mapped to the byte stream described as the variable file. After

reading each byte, algorithm determines if the examined byte is the last in the stream and/or whether the output buffer is not full. The algorithm stops injecting data into the input buffer if: (a) the amount of stored data reaches the maximum size of the packet, or (b) all data has been read from the file buffer. The packet is send after the completion of this process. Algorithm of sending NFC message is presented in Figure 2.

3.3 Data receiving

Algorithm describing the behavior of the system during the process of packets receiving is presented in Figure 3. In first step the algorithm analyzes received records. It is verified whether the payload is not empty and checks the database to determine whether another packets of the same message were not transmitted. Data is appended to the stream in case, when another part of the message is found. If the application determines that the received packet is the last one, it is marked in the database. The algorithm decides whether the packet is the final one by comparing the parameters: the number of the package and the total number of packets.

If the number of packet is equal to the number of message packets, the algorithm assumes that the message was sent. Parts of the message are sent sequentially. The sending node will retransmit the packet, if a confirming flag is not sent by the receiver.

3.4 The automatic NFC multi-packet transfer

The basic mechanism to NFC data communication is Android Beam. As a standard procedure before sending a message to another device, Android Beam enforces to tap mobile device screen. This procedure prevents device from unwanted automatic data send. Automatic data send is mandatory for multi packet data transfer, therefore we propose the method to handle this problem. Automatic tap functionality was implemented using Xposed framework. The framework enables dynamic simulation of the tap action. Xposed modules allow to perform actions immediately before and after triggering a captured method. In order to capture the system methods we re-implemented methods: `handleLoadPackage` and `findAndHookMethod`. This methods are called with parameters: the package name, the name of the hooked method and object of an `XCMETHODHOOK` abstract class. The object have the appropriate method responsible for modifying the behavior of the hooked method.

3.5 Delay tolerant networks support

The NFC multi-packet communication system was proposed as a part of DTN system for a smartphone. The full functionality of DTN device consists of a group of functions responsible for communication and storage. In terms of architecture, this set of functionalities was implemented in an integrated way.

The first component includes a group of functions responsible for autonomous sending and receiving NFC multi-packet messages. The functionality of communications in our implementation consists of the use of communication NFC interfaces. DTN paradigm store-carry-forward is implemented in classical way based on simple Epidemic Routing idea. This means that all messages are replicated and send to every connected device.

4 Results

The goal of our tests was to compare the effectiveness of the NFC multi-packet data transfer process for approaches with manual and automatic handling. Manual handling requires Tap To Send (TTS), the automatic handling is performed without TTS.

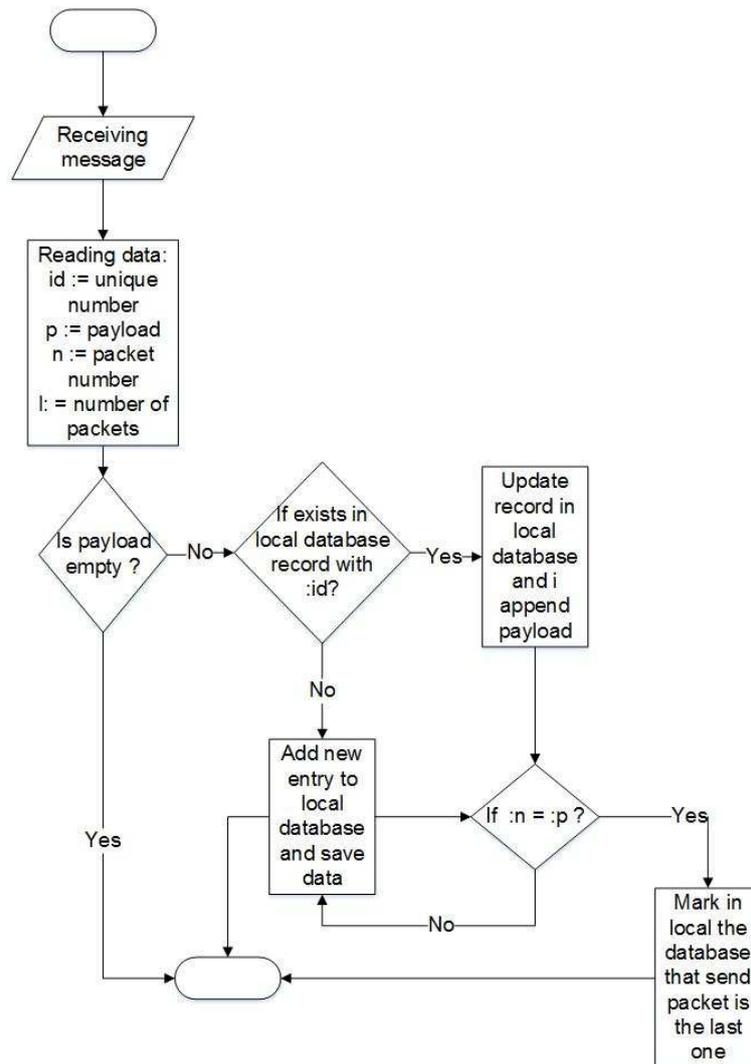


Figure 3: Block diagram of the algorithm responsible for receiving the NFC message

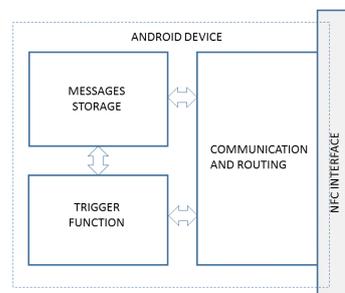


Figure 4: Modular architecture for DTN via NFC using Android device

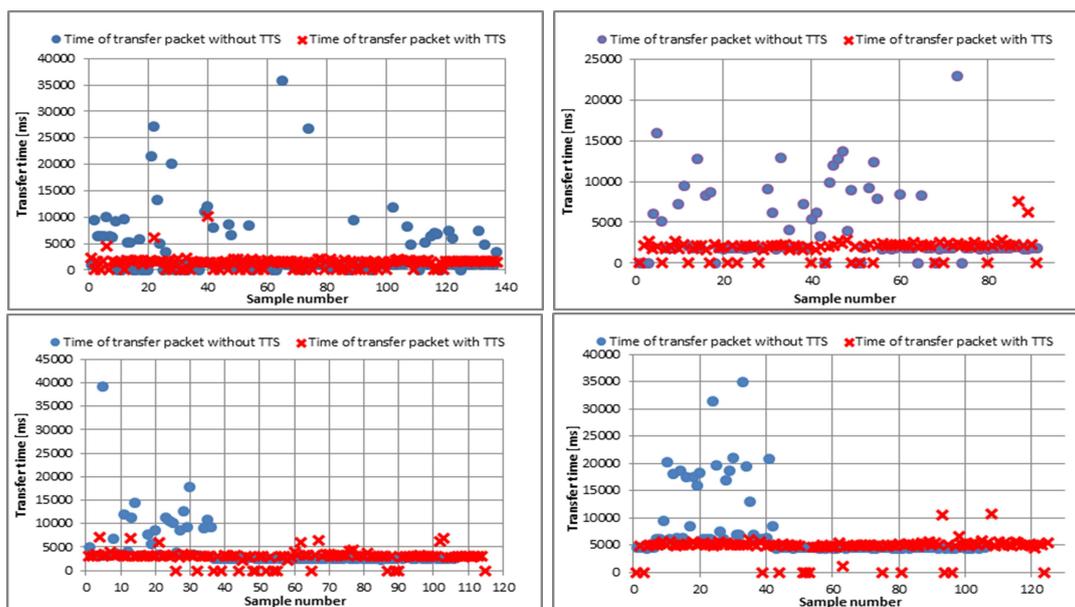


Figure 5: The total mean transfer time for automatic and manual transfer for packets: a) 2KB, b) 4KB, c) 8KB and d) 16K

For tests we proposed packets with sizes: 2 KB, 4KB, 8 KB, 16 KB, 32 KB and 64 KB. For each message more than 100 measurements were done. For each measurement the transfer execution time and the total mean throughput was reported. The total transfer time was defined as a sum of: (a) preparation of the NFC message, (b) user interaction (during tests with TTS), (c) data transfer. For TTS we manually perform screen tapping as soon as it is possible.

Tests were done on Samsung Galaxy S3 and LG L65 mobile phones. Both devices have got built-in NFC module. Samsung Galaxy S3 was working on Android 4.0 operation system, LG L65 was working on 4.4 Android operation system. Tests were done in 12 square meters room.

At first we perform measurements of packet transfer time (see Figure 5). The time increases with the size of the packet as expected for TTS and without TTS. Some transfer attempts were failed during tests. Sessions were interrupted by the Logical Link Control Protocol (LLCP). LLCP is a second layer protocol responsible for setting up session between two NFC enabled devices in peer-to-peer communication. LLCP was returning errors during setting up session (marked in Figure 5 as 0 ms transfer time sample) or setting up session longer than expected. The second circumstance had big influence on the transfer time, which was getting longer.

The effectiveness defined as transmission success ratio of both solutions was presented in Figure 6. The effectiveness is increasing with the size of the packet.

Success ratio was higher than 80 % almost in each test which we have done. In most cases approach without TTS was more efficient. Furth more there was 100% correctly sent samples for 8 KB and 16 KB packets.

Although, the efficacy of transferring data without the TTS was higher, there were more samples with extreme values. Transferring time reached tens of seconds.

Xposed is not officially supported by Google. This framework can be unstable. Release of the next Android version cause that Xposed authors has to adopted framework to the new version. We suspect that Xposed instability has got influence on measurements without TTS tests.

The variation and standard deviation were much larger for transmission with bypassed TTS. Because of that we decided to focus on median value to calculate throughput for tests. Figure 7 presents the median time of transferring packets. Results presents the better transfer time for

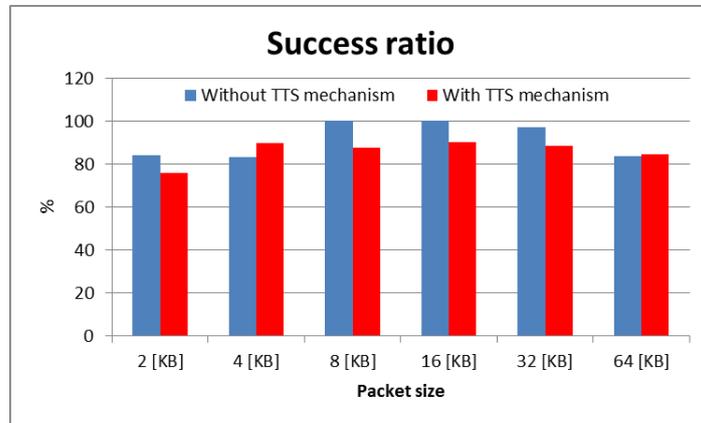


Figure 6: The success ratio of multi-packet transfer

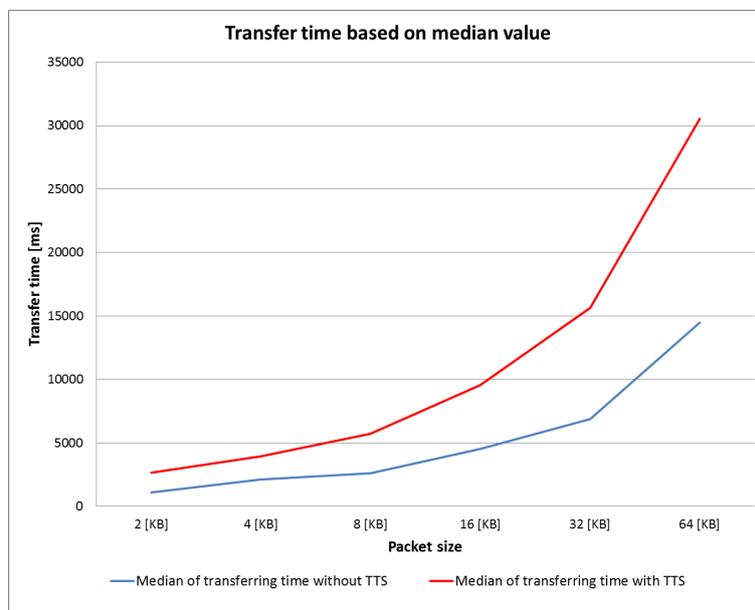


Figure 7: The median of mean transfer time

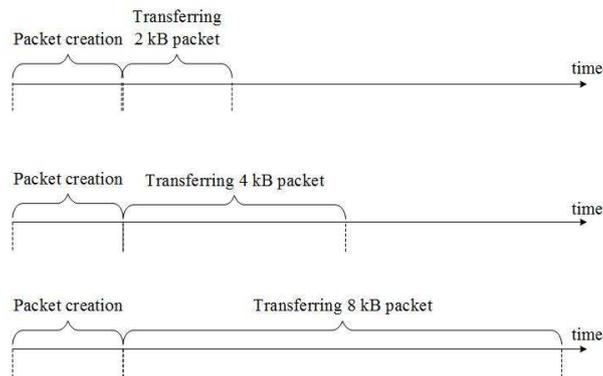


Figure 8: Analysis of time needed to perform the operation while transferring the file

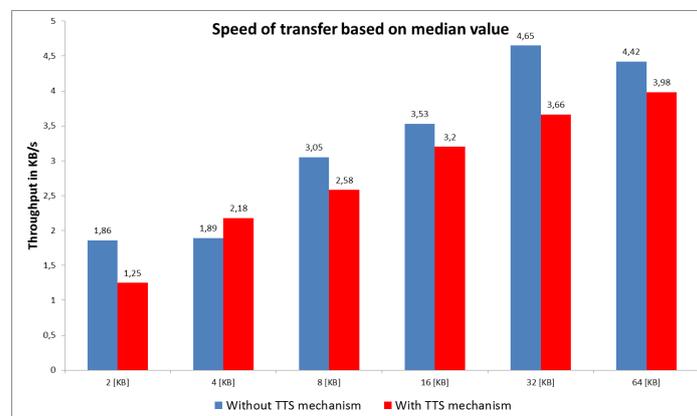


Figure 9: Throughput in function packet size

automatic solution - without TTS.

The second parameter measured was throughput - the total speed of data transfer. We determine the throughput as a message size divided by total transfer time. The total transfer time is the sum of preparation message time, user interaction time and data transfer time. Data transfer time increases with the size of the packet. Creation time is constant for all samples in the packet size group and has huge impact on the results. The throughput increases with the size of the packet. Packet creation time is more negligible with larger packet size. This phenomenon is illustrated on Figure 8. Moreover the automatic mechanism is more effective through the TTS mechanism elimination. Results was presented in Figure 9.

5 Conclusions

In this paper we have presented the idea of NFC multi packet communication for DTN networks. We decided to create this work due to the lack of data about NFC usage in DTN networks especially for larger message transmission which requires fragmentation. We described in detail algorithms for sending, receiving and automatic handling of fragmented message. The algorithms was implemented on Android device and extensively tested. We examined our automatic approach comparing with human-involved standard Android Tap-To-Send method. All test was conducted with different packet sizes. Obtained results confirm the good performance of our solution. The automatic approach has lower total transfer time and higher throughput then manual one.

Bibliography

- [1] Burleigh S., Hooke A., Torgerson L., Fall K., Cerf V., Durst B., Scott K., Weiss H. (2003); Delay-tolerant networking: an approach to interplanetary internet. *IEEE Commun. Mag.*, 128-136, 2003. doi:10.1109/MCOM.2003.1204759
- [2] Curran K., Millar A., Mc Garvey C. (2012); Near field communication, *International Journal of Electrical and Computer Engineering*, 2(3), 2012.
- [3] Domaszewicz J., Koziuk M., Schoeneich R.O. (2008); Context-Addressable Messaging Service with Ontology-Driven Addresses, in *Proc. On the Move to Meaningful Internet Systems: OTM 2008*, 1471-1481, 2008. doi:10.1007/978-3-540-88873-4_37
- [4] Dziekonski A.M., Schoeneich R.O. (2016); DTN Routing Algorithm for Networks with Nodes Social Behavior, in *Proc. International Journal of Computers Communications and Control*, 11(4), 457-471, 2016. doi:10.15837/ijccc.2016.4.1454
- [5] Golanski M., Schoeneich R.O., Siwko M. (2010); The algorithm for distribution of large-size data in the Wireless Ad-hoc Sensor Network, in *proc. Concepts and Implementation for Innovative Military Communications and Information Technologies, Military University of Technology*, 577-584, 2010.
- [6] Ha P., Yamamoto H., Yamazaki K. (2013); Using Autonomous Air Vehicle in DTN Sensor Network for Environmental Observation, in *Proc. Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual. IEEE*, 447-450, 2013. Doi:10.1109/COMPSAC.2013.74
- [7] Fall K. (2003); A delay-tolerant network architecture for challenged internets, In *Proc. of The 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03)*, 27-34, 2003. doi:10.1145/863955.863960
- [8] Fan C.-S. (2016); HIGH: A Hexagon-based Intelligent Grouping Approach in Wireless Sensor Networks, *Advances in Electrical and Computer Engineering*, 16(1), 41-46, 2016. doi:10.4316/AECE.2016.01006
- [9] Fang W., Li S., Liang X., Li Z. (2012); Cluster-based Data Gathering in Long-Strip Wireless Sensor Networks, *Advances in Electrical and Computer Engineering*, 12(1), 3-8, 2012. doi:10.4316/AECE.2012.01001
- [10] Juang P., Oki H., Wang Y., Martonosi M., Peh L.S., Rubenstein D. (2002); Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet, *SIGOPS Oper. Syst. Rev.*, 36(5) (October 2002), 96-107. doi:10.1145/635508.605408
- [11] Kawecki M., Schoeneich R.O. (2016); Mobility-based routing algorithm in delay tolerant networks, in *Proc. EURASIP Journal on Wireless Communications and Networking*, 81, 1-9, 2016. doi:10.1007/978-3-642-32518-2_23
- [12] Koziuk M., Domaszewicz J., Schoeneich R.O., Jablonowski M., Boetzel P. (2008); Mobile Context-Addressable Messaging with DL-Lite Domain Model, in *Proc. Smart Sensing and Context*, 5279, 168-181, 2008. doi 10.1007/978-3-540-88793-5_13
- [13] Lindgren A., Doria A., Schelen O. (2003); Probabilistic routing in intermittently connected networks, *ACM SIGMOBILE Mobile Computing and Communications Review*, 19-20, 2003. doi:10.1145/961268.961272

-
- [14] Lu Z., Fan J. (2010); Delay/disruption tolerant network and its application in military communications, *Proc. of computer design and applications*, 2010. doi: 10.1109/IC-CDA.2010.5541302
- [15] Matsuzaki R., Ebara H., Muranaka N. (2015); Rescue Support System with DTN for Earthquake Disasters, *IEICE Trans. Commun.*, 98(9), 1832-1847, 2015, doi:10.1587/transcom.E98.B.1832
- [16] Ondrus J., Pigneur Y. (2007); An Assessment of NFC for Future Mobile Payment Systems, *Proceedings of the International Conference on the Management of Mobile Business (ICMB '07)*. *IEEE Computer Society*, 2007, doi:10.1109/ICMB.2007.9
- [17] Quwaider M., Biswas S. (2010); DTN routing in body sensor networks with dynamic postural partitioning, *Ad Hoc Networks*, 8(8), 824-841, 2010, <http://dx.doi.org/10.1016/j.adhoc.2010.03.002>
- [18] Palka P., Schoeneich R.O. (2013); Multi-commodity Trade Application to the Routing Algorithm for the Delay and Disruptive Tolerant Networks, in *Proc. New Trends in Databases and Information Systems*, 185, 241-250, 2013. doi:10.1007/978-3-642-32518-2_23
- [19] Schoeneich R.O., Domaszewicz J., Koziuk M. (2009); Concept Based Routing in Ad-Hoc Networks, *Lecture Notes In Computer Science*, 5408, 43-48, 2009. doi:10.1007/978-3-540-92295-7_8
- [20] Schoeneich R.O., Golanski M. (2007); Mesh Cluster Based Routing Protocol: Enhancing Multi-hop Internet Access using Cluster paradigm, *Proc. EUROCON, 2007. The International Conference on Computer as a Tool*, 962-965, 2007. doi:10.1109/EURCON.2007.4400318
- [21] Spyropoulos T., Psounis K., Raghavendra C.S. (2005); Spray and wait: an efficient routing scheme for intermittently connected mobile networks, *Proc. of The 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, 252-259, 2005. doi:10.1145/1080139.1080143
- [22] Vladuta A.-V., Pura M.L., Bica I. (2016); MAC Protocol for Data Gathering in Wireless Sensor Networks with the Aid of Unmanned Aerial Vehicles, *Advances in Electrical and Computer Engineering*, 16(2), 51-56, 2016. doi:10.4316/AECE.2016.02007
- [23] Want R. (2006); An introduction to RFID technology, *IEEE Pervasive Computing* 5(1), 25-33, 2006. doi:10.1109/MPRV.2006.2
- [24] Want R. (2011); Near field communication, *IEEE Pervasive Computing*, 3(10), 4-7, 2011. doi:10.1109/MPRV.2011.55
- [25] Zimmerman T.G. (1996); Personal area networks: near-field intrabody communication, *IBM Syst. J.*, 35(3-4), 609-617, 1996. doi:10.1147/sj.353.0609

Fast Motion Estimation Algorithm using Hybrid Search Patterns for Video Streaming Application

R. Vani, N. Ushabhanu, M. Sangeetha

Vani Rajamanickam*

Meenakshi College of Engineering, Chennai, India

*Corresponding author: ecehod@mce.edu.in

Ushabhanu Nageswaran

Valliammai Engineering College, SRM Nagar, Chennai, India

ushabhanun.ece@valliammai.co.in

Sangeetha Marikkannan

Bharath University, Chennai, India

Abstract: The objective of the paper is to develop new block matching Motion Estimation (ME) algorithm using hybrid search patterns along the direction of best match. The search efficiency for sequences with fast motions and high resolutions is improved by proposing New Cross Diagonal-Hexagon Search (NCDHS) algorithm which involves a novel multi half-hexagon grid global search pattern and a cross diagonal-hexagon local search pattern. The new search pattern enables the proposed algorithm to perform better search using 9.068 search points on an average, to obtain optimal motion vector with slight improvement in quality. This inturn reduces ME Time upto 50.11%, 47.12%, 32.99% and 43.28% on average when compared to the existing Diamond Search (DS), Hexagon Search (HS), New Cross Hexagon Search (NHEXS) and Enhanced Diamond Search (EDS) algorithms respectively. The novelty of the algorithm is further achieved by applying the algorithm proposed for live streaming application. The NCDHS algorithm is run on two MATLAB sessions on the same computer by establishing the connection using Transmission Control Protocol (TCP) /Internet Protocol (IP) network. The ME Time obtained is 14.5986 seconds for a block size 16×16 , is less when compared to existing algorithms and that makes the NCDHS algorithm suitable for real time streaming application.

Keywords: hybrid search patterns, motion estimation time, search points, peak signal-to-noise ratio (PSNR).

1 Introduction

The H.264/Advanced Video Coding (AVC) standard was developed by Joint Video Team (JVT) in the year 2003 [6]. The H.264 standard adopts the Variable Block Size Motion Estimation (VBSME) feature which enables the standard to find its importance in various applications such as High Definition Digital Video Disc (HD-DVD), digital video broadcasting and mobile applications as discussed by Chen et al. [2]. The Variable Block Size Motion Estimation (VBSME) feature plays a vital role in the H.264 video coding standard. The input video sequence is processed as frames and each frame is divided into macroblocks. As the Human Visual System (HVS) is more sensitive to luma than the chroma components, the H.264 standard uses 4:2:0 sampling format as mentioned by Richardson [8], in which each macroblock consists of one block of 16×16 pixels for representing the luminance component and two blocks of 8×8 pixels for representing the chroma components as shown in Fig. 1.

The 16×16 block known as MacroBlock (MB) is the basic building block of the video coding standard which is subdivided into 16×8 or 8×16 or 8×8 sub-blocks. An 8×8 sub-macroblock

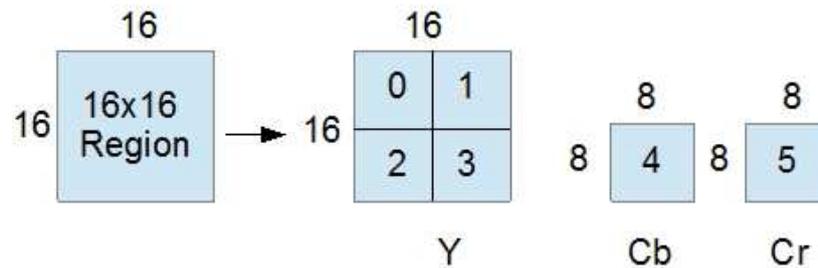
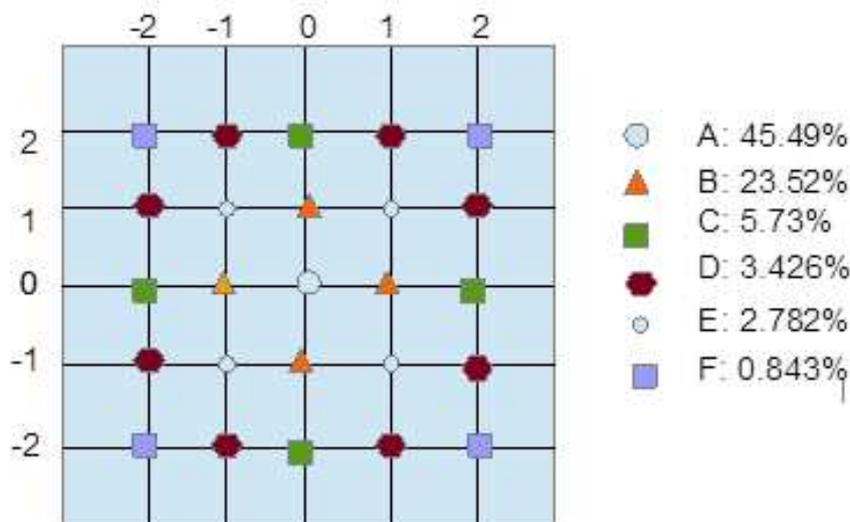


Figure 1: Macroblock (4:2:0)

can be further divided into 4×4 or 8×4 or 4×8 blocks. According to Wen et al. [9], for objects with different motions and directions, the VBSME feature has improved the coding performance in H.264 standard than the FBSME method adopted in the earlier standards. This feature has enabled the motion estimation algorithms in the H.264 standard to find its applications in video conferencing, video streaming, surveillance, video telephony, multimedia networks, etc as mentioned in Li & Yang [7].

1.1 Concepts involved in block matching algorithms

In the encoder design, the ME process occupies about 53% of the time when one reference frame is used and about 70.20% of the total encoding time, when four reference frames are used, as mentioned in Huayi et al. [5]. It increases the overall computational complexity in the encoder design. The probability of MV distribution has to be analyzed to obtain better match accuracy with less number of search points in the ME algorithms. In a video sequence, more than 70% of the blocks are considered as either stationary at the location (0,0) or quasi stationary at the location $(\pm 1, 0)$ or $(0, \pm 1)$. As discussed by Zhao & Xu [10], in the 5×5 search area as shown in Fig. 2, the higher possibility of MV occurs at the centre position A, which is about 45.49%.

Figure 2: Probability of MV distribution in the 5×5 search area

At a pixel distance of ± 1 from the centre, the MV is distributed for about 23.52% which is indicated as position B. At a pixel distance of ± 2 from the centre, the possibility of distribution is about 5.73% indicated by position C. At the positions D, E and F, the possibility of MV distribution is very less and the sum of probability is about 7.051%.

1.2 Limitations of the existing algorithms

The Diamond Search (DS) algorithm proposed by Zhu & Ma [12] starts searching with Large Diamond Search Pattern (LDSP) and this process is repeated a number of times until the best match is obtained. Then final refinement is done using Small Diamond Search Pattern (SDSP) which is applied only once. Zhu et al. [11] proposed Hexagon Search (HS) algorithm that achieves better speed improvement than the existing DS algorithm with similar distortion performance. It is used for implementing the video codec standards such as H.261, MPEG-1 and MPEG-2. Cheung & Po [3] proposed Cross-Diamond-Hexagonal Search (CDHS) algorithm which uses cross shaped pattern followed by hexagon search pattern. It performs faster searching than DS algorithm. The New Cross Hexagon Search (NCHEXS) hybrid algorithm proposed by Belloulta et al. [1] is developed for H.264 standard. It utilizes hybrid search patterns such as the cross search and the hexagon search patterns for search path analysis.

The performance of Motion Estimation algorithms are influenced by the number of search points for many video based applications. The traditional ME block matching algorithms used in the earlier standards are based on fixed block size and are not able to incorporate different motions in the video frames. For real time video sequences with different motions, the H.264 standard adopts variable block size for video processing to address the complexity problems. The algorithms mentioned in the literature adopt fixed and hybrid search patterns which have reduced the complexity in ME Time. It is necessary to develop the motion estimation algorithm based on motion vector characteristics with quality improvement. The objective of the paper is to propose new efficient ME algorithm for improving the search efficiency using hybrid search patterns and to have less time consumption.

The rest of the section is organised as follows: Section 2 explains about the proposed algorithm and its search path analysis. In section 3, the performance analysis of the proposed and existing algorithm is done using MATLAB software. Section 5 illustrates the live streaming of the proposed algorithm. Finally, section 6 concludes the future possibilities of the work.

2 Proposed NCDHS algorithm

The NCDHS algorithm involves both coarse grain and fine grain search to find the best matching between the successive frames. Initially, we consider the Motion Vector Point (MVP) to be at the position $(0,0)$. The first step is designed using unsymmetrical Cross pattern with step size ± 1 , and followed by modified asymmetric Hexagon with step size ± 1 and ± 2 . The last step in which the fine grain is done which utilises either a cross or diagonal Hexagon depends on the minimum cost with step size ± 1 from centre point. The steps for proposed algorithm are given below:

Step 1: The Coarse grain search starts with the unsymmetrical Cross search pattern with pixel distance ± 1 as shown in Fig. 3. The minimum cost point is obtained using SAD computation. If the minimum cost is found to be at centre point, then the search stops else the search continues to Step 2.

Step 2: The half hexagon search is performed with pixel distance ± 1 and ± 2 as shown in Fig. 4 as a second search with minimum cost point from Step 1 acting as a centre point. This reduces the number of search points as the grid chooses the minimum cost either in the upper half or lower half of the hexagon pattern.

Step 3: Switch the search from the coarse to the fine resolution inner search to find the best motion vector.

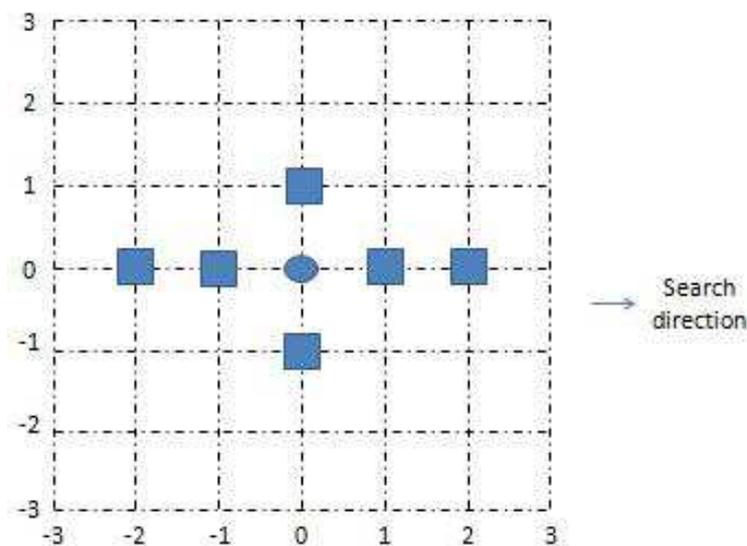


Figure 3: First search

- (i) A small cross pattern with best search point obtained from global search is formed as a centre. If the current minimum SAD is at centre, the search stops. Else it moves to next step 2.
- (ii) The same cross pattern is repeated based on the minimum SAD position. If the current and the previous minimum SAD positions are same, then the selected search point is the best match point else move to step 3.
- (iii) Diagonal-Hexagon search pattern is formed with previous minimum SAD point as centre.

The algorithm steps are iterated until the final best vector is obtained as shown in Fig. 5.

The proposed NCDHS algorithm adopts the Pseudo code as shown in Algorithm 1 to find the minimum cost.

2.1 Theoretical search point analysis

Based on the assumption, let the best match block occur at the position (3,0) as shown in Fig. 6. Initially the search starts with unsymmetrical cross search pattern with seven check points. If the MBD point is at the location (1, 0), the algorithm searches for minimum motion vector using half hexagon grid with two check points (2,-1), (3,0) in addition with the pixels at the location (1,0), (0,-1) and (-1,0) that is reused from the previous search.

With the minimum MBD point found at location (3,0) from second search, the NCDHS algorithm finally searches for four check points around it, to form a small cross pattern. The search point located at (3,0) is found by using 12 search points by adopting NCDHS algorithm.

3 Performance analysis

To evaluate the performance of the NCDHS algorithm in terms of ME Time, search points and PSNR, simulation is done using MATLAB for the proposed and the existing algorithms at a frame rate of 30 fps. The test sequences used are "Foreman" and "Mobile" with a resolution

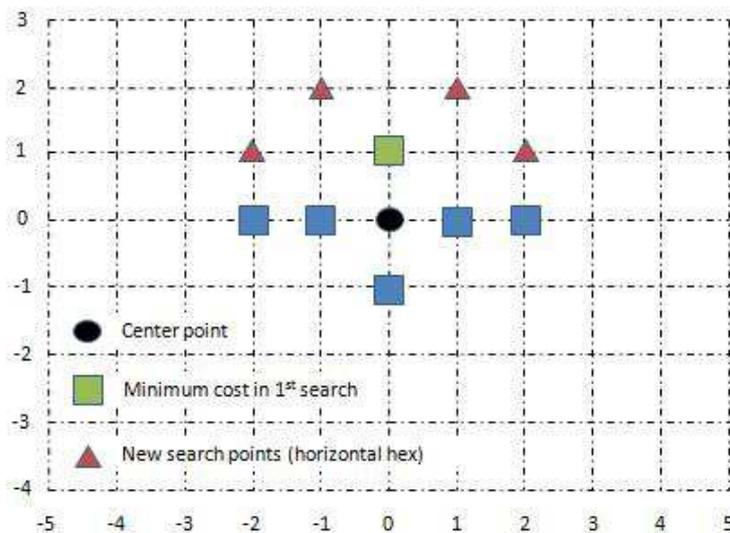


Figure 4: Second search

of 352×288 , "Bqmall" and "Mobisode" with a resolution of 832×480 , "People on street" and "Traffic" with a resolution of 2560×1600 . The block size is chosen as 16×16 with Sum of Absolute Differences (SAD) as the distortion metric. The SAD metric at the search location (m, n) is calculated as given in Eq. (1).

$$SAD(m, n) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |I_1(x, y) - I_2(x + m, y + n)| \quad (1)$$

where $N \times N$ denotes block size, $I_1(x, y)$ indicates the pixel intensity at the location (x, y) in the current frame, $I_2(x, y)$ indicates the pixel intensity at the location (x, y) in the reference frame and (m, n) denotes the displacement vector. The lowest SAD estimates the best position of prediction occurred within the search window.

3.1 Motion estimation time

Table 1 summarises the simulation results obtained for the ME Time taken by the proposed NCDHS and the existing algorithms. For the mobile sequence, the ME Time obtained for proposed NCDHS algorithm requires 10.161 seconds and the ME Time obtained for the existing DS algorithm proposed by Zhu & Ma [12], HS algorithm proposed by Zhu et al. [11] and NCHEXS algorithm proposed by Belloulata et al. [1] are 19.881, 19.448 and 13.321 seconds respectively. The results show that the proposed NCDHS algorithm has reduced the ME Time when compared to the other existing algorithms.

3.2 Search points

The average search points for each block in a frame are calculated for different sequence in encoding 30 frames. Table 2 shows that the proposed NCDHS algorithm consumes the smallest number of search points compared to other algorithms. The results show that the average number of search points in NCDHS algorithm is 9.068. In order to demonstrate the performance of the proposed algorithm, the Fig. 7 gives the result of the average number of computations per block for different video sequences as the frame number increases. For Mobisode sequence, the search

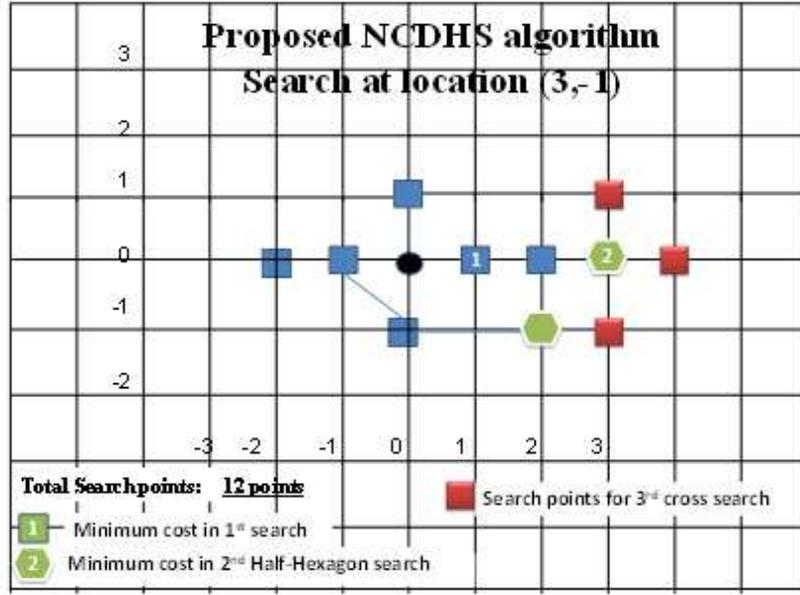


Figure 5: Search path analysis of NCDHS algorithm

Table 1: Simulation results showing ME Time of the NCDHS and the existing algorithms

Test sequences	ME Time obtained for existing and proposed algorithms (in seconds)			
	DS [12]	HS [11]	NCHEXS [1]	Proposed NCDHS
Foreman	19.387	19.86	15.233	9.48
Mobile	19.881	19.448	13.321	10.161
BQmall	61.207	54.765	48.776	31.506
Mobisode	77.945	69.422	57.707	37.461
Video codec Standards	H.261, H.263	H.261, MPEG-1,2	H.264	

points obtained for NCDHS algorithm is 8.174 which is less when compared to the existing algorithms. It clearly manifests the superiority of the proposed NCDHS algorithm to the other methods in terms of number of search points used.

4 ME Time reduction and PSNR improvement

The Δ ME Time and Δ PSNR denotes the CPU time reduction in ME process and improvement of Peak Signal to Noise Ratio (PSNR) of the encoded image respectively. They are defined as in Eqs. (2)–(4):

$$\Delta \text{ME Time} = \text{MET}_{\text{proposed}} - \text{MET}_{\text{original}} \quad (2)$$

$$\% \Delta \text{ME Time} = \frac{(\text{MET}_{\text{original}} - \text{MET}_{\text{proposed}})}{\text{MET}_{\text{original}}} \times 100 \quad (3)$$

$$\Delta \text{PSNR} = \text{PSNR}_{\text{proposed}} - \text{PSNR}_{\text{original}} \quad (4)$$

Algorithm 1 Pseudocode of NCDHS algorithm

Input: Initialize a large cross pattern such that $SP_H = 2 * (SP_V)$
Output To find minimum cost

- 1: **Begin**
- 2: **if** ($MVP = \min SAD$) **then**
- 3: $MV = MVP$
- 4: **else**
- 5: with $p = \min SAD$, select large half hexagon with range $\{p+2, p-2\}$ calculate the overall $\min SAD$ ($\min SAD_{cg}$)
- 6: **end if**
- 7: with $\min SAD_{cg}$, initialize fine-grain search with small cross with 5 points
- 8: **if** ($\min SAD_{current} = \min SAD_{prev}$) **then**
- 9: $MV = \min SAD(x,y)$
- 10: **else if** ($\min SAD$ is in the same row or column) **then**
- 11: cross slides to next position
- 12: **else**
- 13: with $\min SAD(x,y)$, a diagonal-hexagon pattern is formed
- 14: **if** ($\min SAD_{prev} = \min SAD_{current}$) **then**
- 15: $MV = \min SAD(x,y)$
- 16: **end if**
- 17: **end if**
- 18: **End**

Table 2: Average number of search points per block

Sequence	Average number of search points (computations) for each block				
	DS [12]	HS [11]	NHEXS [1]	EDS [4]	Proposed
Foreman	16.839	12.716	12.084	12.757	10.48
Mobile	14.323	12.73	11.341	11.634	9.396
BQMall	15.519	11.439	11.204	11.307	8.225
Mobisode	18.793	13.31	12.982	13.86	8.174
Average computations	16.368	12.548	11.902	12.3895	9.068

where $MET_{proposed}$ and $PSNR_{proposed}$ denotes the ME Time and PSNR of the proposed algorithm. The $MET_{original}$ and $PSNR_{original}$ represents the ME Time and PSNR of DS, NHEXS, HS and EDS algorithms.

Table 3 shows the improvement in PSNR and the average reduction of ME Time for proposed algorithm when compared to existing DS, HS and NHEXS search patterns. It is clear that the NCDHS algorithm reduces an average of 50.11% of time for motion estimation compared to Diamond Search (DS) consequently 47.12% of Motion Estimation time compared to Hexagon Search (HS), 32.99% of Motion Estimation time compared to New Hexagon Search (NHEXS) and 43.28% of Motion Estimation time compared to Enhanced Diamond Search (EDS). This indicates that the average reduction in ME Time is greatly achieved by the proposed algorithm with slight improvement in PSNR, when compared to the existing algorithms. The comparison of the existing and the proposed algorithms is shown in Fig. 8 highlighting 30 frames of different video sequences. When compared to the existing algorithms, the proposed scheme can reduce the ME Time of the encoded image, as the frame number increases.

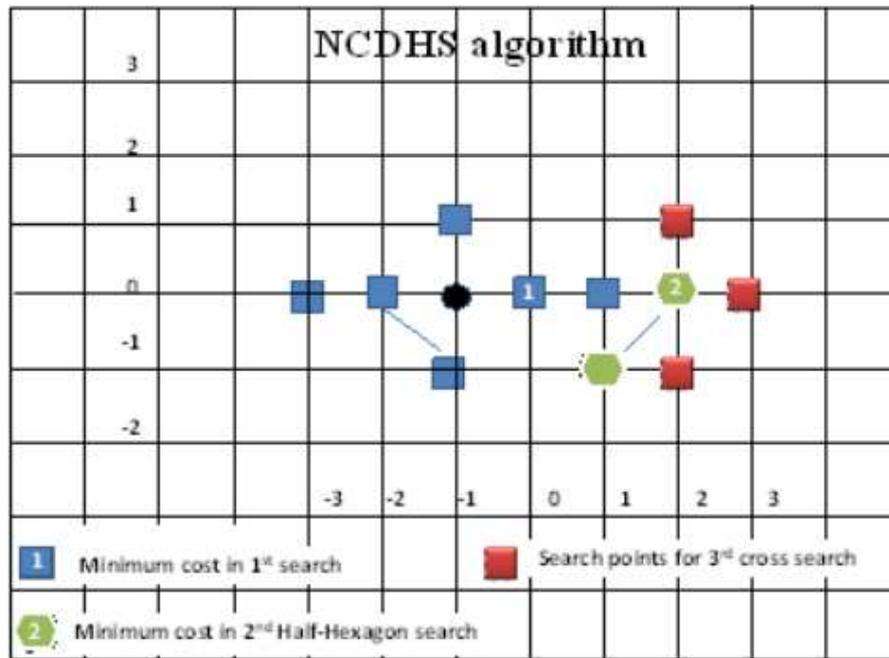


Figure 6: NCDHS algorithm search point analysis

Table 3: Simulation results showing PSNR improvement and average reduction of ME time

Sequence	PSNR improvement (dB) (Δ PSNR)				Δ ME Time (in%)			
	Proposed vs DS	Proposed vs HS	Proposed vs NHEXS	Proposed vs EDS	Proposed vs DS	Proposed vs HS	Proposed vs NHEXS	Proposed vs EDS
Foreman	0.57	0.11	0.06	0.15	-51.10	-52.26	-37.76	-51.83
Mobile	0.64	0.34	0.1	0.33	-48.89	-47.75	-23.72	-46.99
BQMall	0.62	0.24	0.02	0.05	-48.52	-42.47	-35.40	-39.07
Mobisode	0.92	0.84	0.24	0.59	-51.93	-46.03	-35.08	-35.23
Average % time savings of the NCDHS algorithm					-50.11	-47.12	-32.99	-43.28

5 Video streaming application based on NCDHS algorithm

The block diagram of video streaming system for the proposed NCDHS algorithm is shown in Fig. 9. The NCDHS server and the NCDHS client run on two MATLAB sessions on the same computer as shown in Fig. 10. After establishing the connection using Transmission Control Protocol (TCP) /Internet Protocol (IP) network, the server and client becomes ready for streaming. At the server end, the input video frames of frame size 352×288 are captured by the camera and is displayed on the PC screen. The proposed NCDHS algorithm starts estimating the motion between the frames by computing SAD values. The motion vectors are transmitted to the client over the network. The reconstruction of the video frames is done using NCDHS algorithm and the video frames are displayed on the client screen.

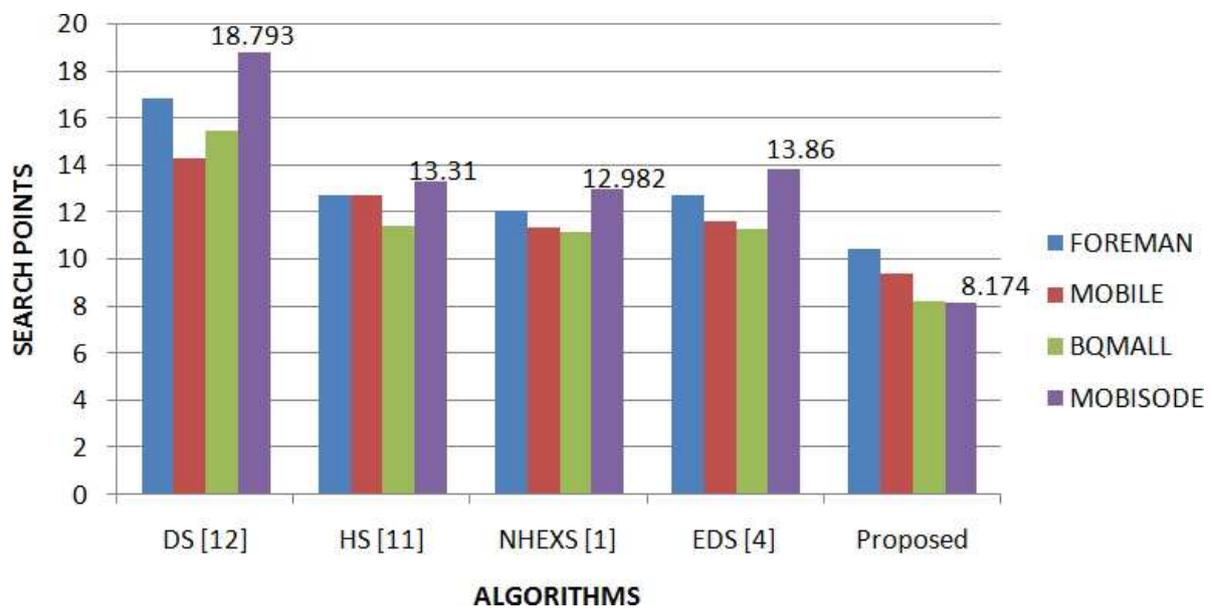


Figure 7: Performance comparison of DS, HS, NHEXS, EDS and the proposed NCDHS algorithm for (a) Foreman Sequence (b) Mobile sequence (c) Mobisode sequence (d) BQMALL sequence in terms of average number of search points

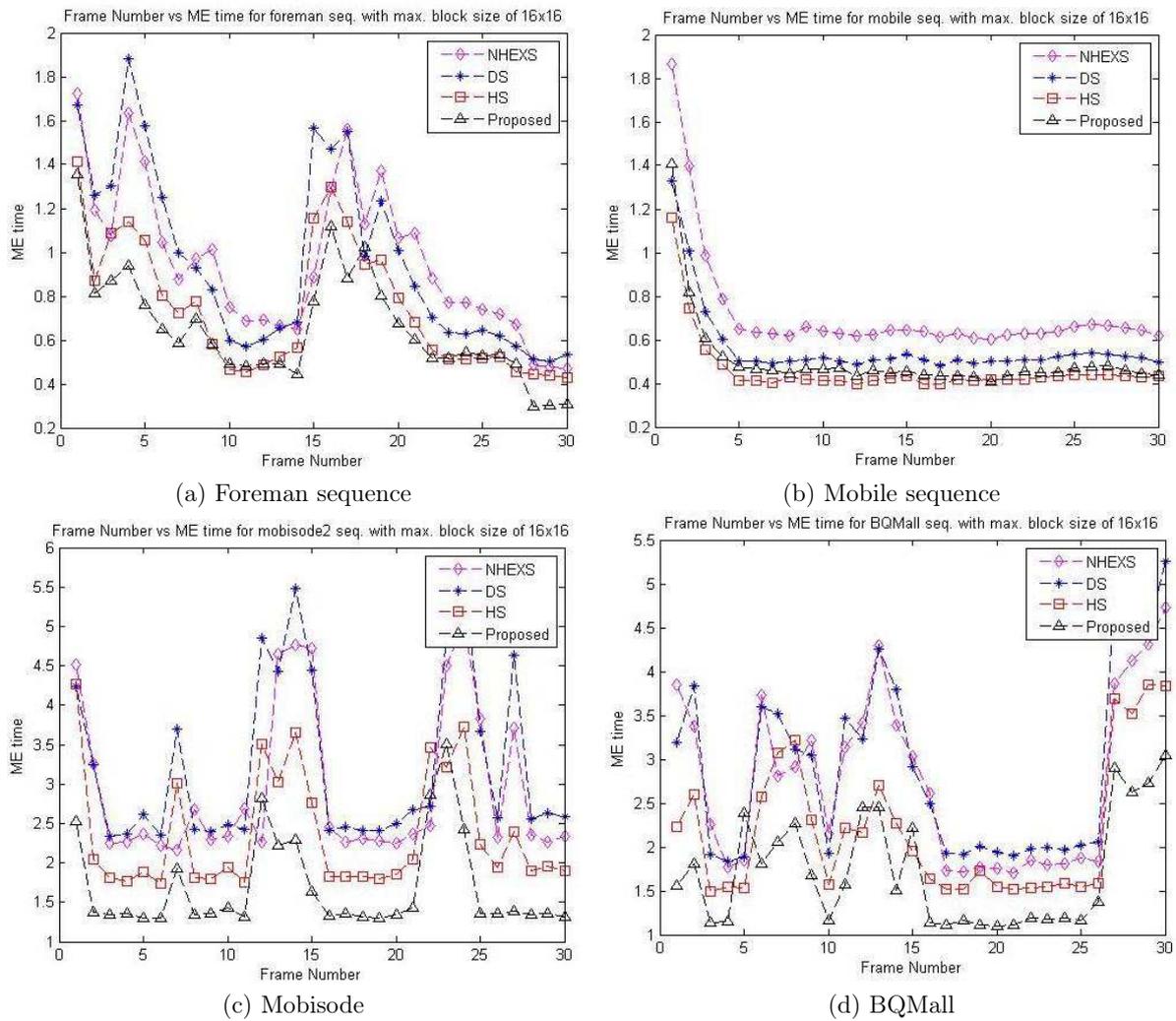


Figure 8: Performance comparison of DS, HS, NHEXS and the proposed NCDHS for (a) Foreman sequence (b) Mobile sequence (c) Mobisode sequence (d) BQMALL sequence in terms of time reduction in ME process

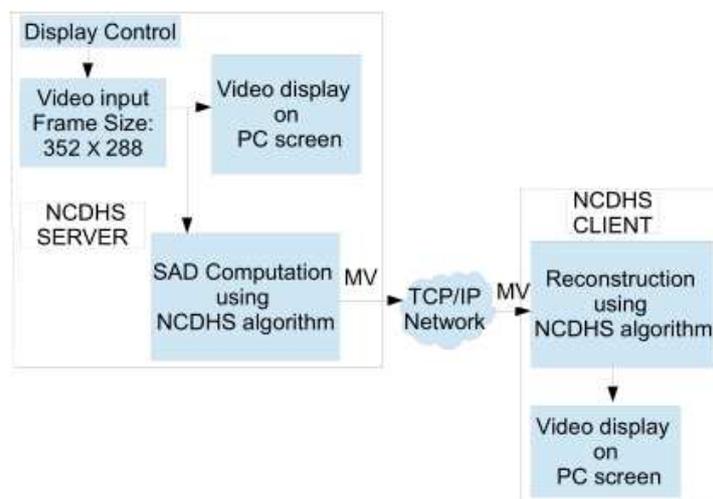


Figure 9: Video streaming using NCDHS algorithm

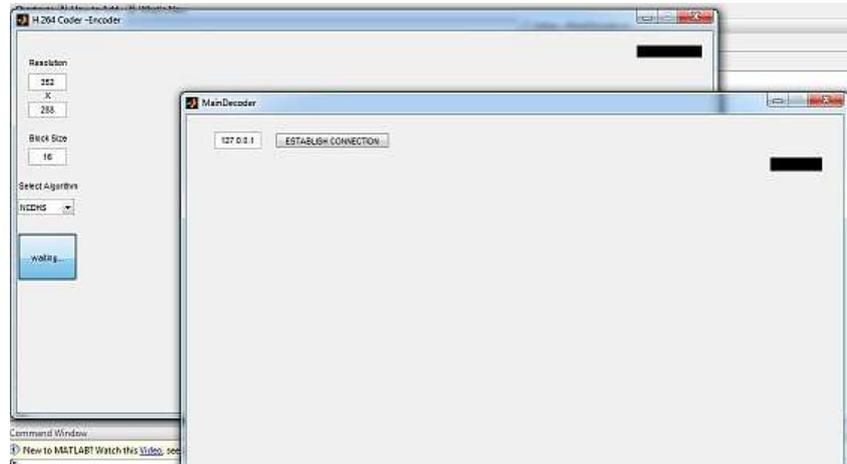


Figure 10: Snapshot of NCDHS server and client sessions using NCDHS

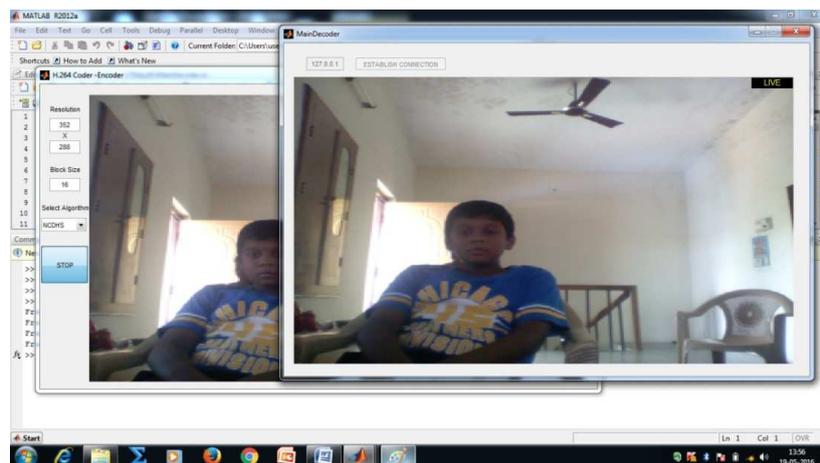


Figure 11: Sample of live input and the reconstructed output in the server and client using NCDHS algorithm

Table 4: Simulation results showing ME Time of the NCDHS and the existing algorithms

Video Codec Standard	H.264		
Algorithms	NCHEXS [1]	EDS [4]	NCDHS
ME Time (in Seconds)	17.18488	19.1599	14.5986

The sample video input and the reconstructed output is as shown in Fig. 11. For a frame size 352×288 and block size 16×16 , the live video frames are captured using laptop camera. For the first thirty frames, the motion estimation time taken for the proposed NCDHS and the existing algorithms for block size 16×16 is as shown in Table 4. For the live video sequence, the ME time obtained for NCDHS algorithm is 14.5986 seconds, for the NCHEXS algorithm introduced by Belloulata et al. [1] is 19.1599 seconds and for the EDS algorithm discussed by Devi et al. [4] is 17.18488 seconds. The simulation results show that the ME time obtained by the NCDHS algorithm is less when compared to the existing algorithms. The speed of the algorithm can be further improved by reducing the motion estimation time when implemented on dedicated hardware as suitable for real time streaming applications.

6 Conclusion

The proposed New Cross Diagonal-Hexagon Search (NCDHS) algorithm has adopted novel half-hexagon grid global search pattern and a cross diagonal-hexagon local search pattern. The hybrid global and local search pattern utilises less search points to obtain optimal motion vector and this inturn reduces the motion estimation time, when tested on live sequences. The search points reduction has reduced the computational complexity when compared to the existing algorithms with little improvement in quality. The new block matching motion estimation algorithm using hybrid search patterns achieves less ME time when compared to the existing algorithms, which makes it suitable for live streaming application. The algorithm proposed can be extended to High Efficiency Video Coding (HEVC)/H.265 Standard by considering the reduction in motion estimation time and by extending the block size to meet the requirements of the standard.

Bibliography

- [1] Belloulata K., Zhu S., Tian J., Shen X. (2011); A novel cross hexagon search algorithm for fast block motion estimation, *International Workshop on Systems, Signal Processing and their Applications*, 1-4, 2011.
- [2] Chen T.C., Lian C., Chen L.G. (2006); Hardware architecture design of an H.264/AVC video codec, *IEEE Asia and South Pacific Conference on Design Automation*, 8, 2006.
- [3] Cheung C.H., Po L.M. (2005); Novel cross-diamond-hexagonal search algorithms for fast block motion estimation, *IEEE Transactions on Multimedia*, 7(1), 16-22, 2005.
- [4] Devi S.R., Rangarajan P., Perinbam J.R.P., Paul R. (2013); VLSI Implementation of High Performance Optimized Architecture for Video Coding Standards, *Acta Polytechnica Hungarica*, 10(6), 237-249, 2013.
- [5] Huayi L., Lini M., Hai L. (2010); Analysis and optimization of the UMHexagons algorithm in H. 264 based on SIMD, *Second International Conference on Communication Systems, Networks and Applications*, 1, 239-244, 2010.

- [6] ITU-T Rec.H.264, ISO/IEC 14496-10 AVC. (2003); Joint Video Team Draft ITU-T recommendation and final draft international standard of joint video specification, 2003.
- [7] Li Z., Yang Q. (2012); A fast adaptive motion estimation algorithm, *International Conference on Computer Science and Electronics Engineering*, 3, 656-660, 2012.
- [8] Richardson I.E. (2011); The H. 264 advanced video compression standard, John Wiley & Sons, 2011.
- [9] Wen X., Au O.C., Xu J., Fang L., Cha R., Li J. (2011); Novel RD optimized VBSME with Matching Highly Data Re-usable Hardware Architecture, *IEEE Transactions on Circuits and Systems for Video Technology*, 21(2), 206-219, 2011.
- [10] Zhao W., Xu S. (2012); Research and optimization of UMHexagons algorithm based on H.264, *IEEE 4th International Conference on Multimedia Information Networking and Security*, 600-603, 2012.
- [11] Zhu C., Lin X., Chau L.P., Lim K.P., Ang H.A., Ong C.Y. (2001); A novel hexagon-based search algorithm for fast block motion estimation, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 3, 1593-1596, 2001.
- [12] Zhu S., Ma K.K. (2000); A new diamond search algorithm for fast block-matching motion estimation, *IEEE Transactions on Image Processing*, 9(2), 287-290, 2000.

A Hybrid Social Network-based Collaborative Filtering Method for Personalized Manufacturing Service Recommendation

S. Zhang, W.T. Yang, S. Xu, W.Y. Zhang

Shuai Zhang*, Wenting Yang, Wenyu Zhang

School of Information

Zhejiang University of Finance and Economics

No.18 Xueyuan Street, Xiasha, Hangzhou 310018, China

*Corresponding author: zhangshuai@zufe.edu.cn

1123797538@qq.com, wyzhang@e.ntu.edu.sg

Song Xu

Business School

NanKai University

No.94 Weijin Road, Tianjin 300071, China

xs_xsong@zufe.edu.cn

Abstract: Nowadays, social network-based collaborative filtering (CF) methods are widely applied to recommend suitable products to consumers by combining trust relationships and similarities in the preference ratings among past users. However, these types of methods are rarely used for recommending manufacturing services. Hence, this study has developed a hybrid social network-based CF method for recommending personalized manufacturing services. The trustworthy enterprises and three types of similar enterprises with different features were considered as the four influential components for calculating predicted ratings of candidate services. The stochastic approach for link structure analysis (SALSA) was adopted to select top K trustworthy enterprises while also considering their reputation propagation on enterprise social network. The predicted ratings of candidate services were computed by using an extended user-based CF method where the particle swarm optimization (PSO) algorithm was leveraged to optimize the weights of the four components, thus making service recommendation more objective. Finally, an evaluation experiment illustrated that the proposed method is more accurate than the traditional user-based CF method.

Keywords: manufacturing service recommendation, social network, collaborative filtering, SALSA, PSO.

1 Introduction

Given the parallel and almost simultaneous rapid development of both Web 2.0 and global manufacturing resources, a large number of manufacturing enterprises have turned to social network-based resources to improve their core competitiveness. For example, service-oriented architectures (SOAs) [7] and their corresponding web services offer dynamic methods that allow manufacturing enterprises to communicate with their suppliers and customers [4]. However, selecting the most appropriate recommendation technique is still a common challenge among numerous manufacturing enterprises.

Recommender methods have been widely applied to address the information overload problem often encountered in recommending personalized products or services by using the collaborative filtering (CF) [1] method, mostly in the commercial domain, including Amazon and Netflix. The traditional CF-based recommender systems usually face serious drawbacks, such as data sparsity and cold-start, which seriously affect the effectiveness of the service recommendation. To solve data sparsity and cold-start problems, social network information can be explored to leverage

service preferences of users, trust relationships among users, and influence of users on others, thus making personalized service recommendations more accurate and objective [6].

Through comparison with and analysis of previous works, this study proposed that the social network-based CF method can also be applied in the area of manufacturing services not only as a solution to the industry's information overload problem but as an alternative to its need for a more accurate and objective personalized recommendation technique. It is assumed that the opinions of trustworthy enterprises have significant influence on the consumers who use social network-based information to access information on required services or products. Apart from trust relationships of enterprises, similarity of consumer enterprises is another important factor for calculating predicted ratings to recommend services, since two enterprises with similar features may have higher similarity of preference ratings on required services.

However, conventional social network-based CF methods still have some serious drawbacks to be explored. First, the traditional trust evaluation methods usually ignore the reputation propagation in social networks. Second, traditional CF recommendation methods basically use single similarity between consumer enterprises (i.e., rating similarity), making the result of service recommendation ineffective. Furthermore, the corresponding weights of influential attributes for calculating predicted ratings are subjectively or averagely determined, resulting in inaccurate manufacturing service recommendation.

To overcome the above barriers, this study proposed a hybrid social network-based CF method to make manufacturing service recommendation more effective and accurate. First, stochastic approach for link structure analysis (SALSA) [15] was leveraged to compute the global reputation values of enterprises in selecting top $K1$ trustworthy enterprises by considering reputation propagation in enterprise social network. In addition to the selection of trustworthy enterprises, three types of similar enterprises were selected as influential components for predicting ratings. The types of enterprise similarity include rating similarity, development stage similarity, and category similarity. After the four influential components were selected, the predicted ratings of candidate services were calculated by using an extended user-based CF method where particle swarm optimization (PSO) [12] was leveraged to automatically and objectively obtain the weights of the four influential components. Finally, the manufacturing services with higher predicted ratings can be recommended to the consumer enterprise, by using user-based CF method.

The rest of this paper is organized as follows: Section 2 summarizes several related works. Section 3 further elaborates on the proposed hybrid method. Section 4 presents the experimental evaluation to verify the effectiveness of the proposed method. Lastly, Section 5 gives the conclusion of this paper, as well as suggestions for further research.

2 Related works

2.1 Social network-based recommender method

Social networks derive their name from social associations among people, and model social connections among individuals or objects [20]. In recent years, large amounts of useful information from social networks have been mined and integrated with recommender systems to enhance the quality of recommendation, and consequently, social network-based recommender methods have been broadly applied and extended to various domains.

For instance, Eirinaki et al. [8] presented a trust-aware system for personalized recommendation, employing both implicit trust and explicit trust between users in the social network. Meanwhile, the Liu et al. [17] study proposed a novel recommendation system that considered social relations and item contents into the Bayesian Probabilistic Matrix Factorization (BPMF) [23] to improve the accuracy of recommendation. Additionally, Sun et al. [25] proposed a dynamic

competitive approach to overcome the problem of environmental change for social network service recommendation. A previous work [29]-also done by the present writers-presented Quality of Service (QoS)-aware service recommendation by combining social network and CF technology to predict the missing QoS values for manufacturing service recommendation.

However, the previous social network-based recommender methods [8, 17, 23, 25, 29], have not considered reputation propagation between consumer enterprises, which makes the result of manufacturing service recommendation ineffective. Therefore, this study employed SALSA, which is an effective link analysis method to compute global and accurate reputation value of enterprises in a social network.

2.2 Stochastic approach for link structure analysis (SALSA)

SALSA is an effective link analysis method that combines the idea of PageRank with the hub and authority idea of HITS [21] and it is usually used in ranking of web pages.

White et al. [28], for instance, proposed a general framework to estimate the relative importance of nodes in a graph by using SALSA algorithm. Najork et al. [18] proposed some definitions of neighborhood graph to enhance the effectiveness and the efficiency of query-dependent link-based ranking algorithms including SALSA algorithm. Furthermore, Borodin et al. [2] presented an extended SALSA-called popularity SALSA (pSALSA)-to extract useful information about the relative ranking of the web pages. Subsequently, Langville et al. [14] proved that SALSA is the best ranking algorithm for Web information retrieval by comparing HITS and PageRank. We find that the link structure of SALSA is similar with transactions among consumer enterprises in a social network. Therefore, this study adopted SALSA to calculate the global reputation values of trustworthy enterprises in a social network.

2.3 Particle swarm optimization (PSO)

PSO [12] was initially proposed by Kennedy and Eberhart in 1995, and the concept was derived from the foraging behaviors of a swarm of birds to address optimization problems. In the past two decades, PSO has been continually extended to facilitate its application in various domains, including education, economics, and engineering.

For instance, Sobecki [24] proposed five swarm intelligence algorithms in the field of student course recommendation, including ant colony optimization (ACO) [5] and PSO. Specifically, PSO was employed to find the set of the optimal neighborhood of students for further grade prediction. The study of Park et al. [19] presented an improved PSO that combined chaotic sequences with the linearly decreasing inertia weights to enhance the global searching capability to overcome nonconvex economic dispatch (ED) problems. Esfahani et al. [9] provided an optimal control system to enhance power quality-where the fuzzy PSO method was proposed for optimization of proportional-integral-derivative (PID) controller. The Tyagi et al. [26] research utilized the multi-objective PSO-based association rule mining model to objectively obtain minimum support and minimum confidence to extract the useful association rules, so that the collaborative filtering recommendations are obviously enhanced.

According to these aforementioned studies, PSO is a promising stochastic optimization technique for handling multi-objective problems; however, it is rarely employed as an optimal service recommendation in the domain of manufacturing services. In the present study, we takes advantage of the PSO algorithm to automatically and objectively obtain corresponding weights of influential components to recommend the most effective and efficient manufacturing services.

3 The proposed hybrid method

To enhance the quality of manufacturing service recommendation, this paper proposes a hybrid social network-based CF method to modify the traditional user-based CF method. Trustworthy enterprises and three types of similar enterprises with different features can be regarded as influential components for calculating predicted rating. Fig. 1 demonstrates the overview of the proposed method, and it is divided into four parts: (a) selection of top $K1$ trustworthy enterprises-considering global reputation propagation in enterprise social network; (b) selection of three types of top K similar enterprises-with different features, including rating similarity, development stage similarity, and category similarity; (c) optimization of four corresponding weights-by means of PSO algorithm; and (d) manufacturing service recommendation-by calculating rating value of required service through extended user-based CF method.

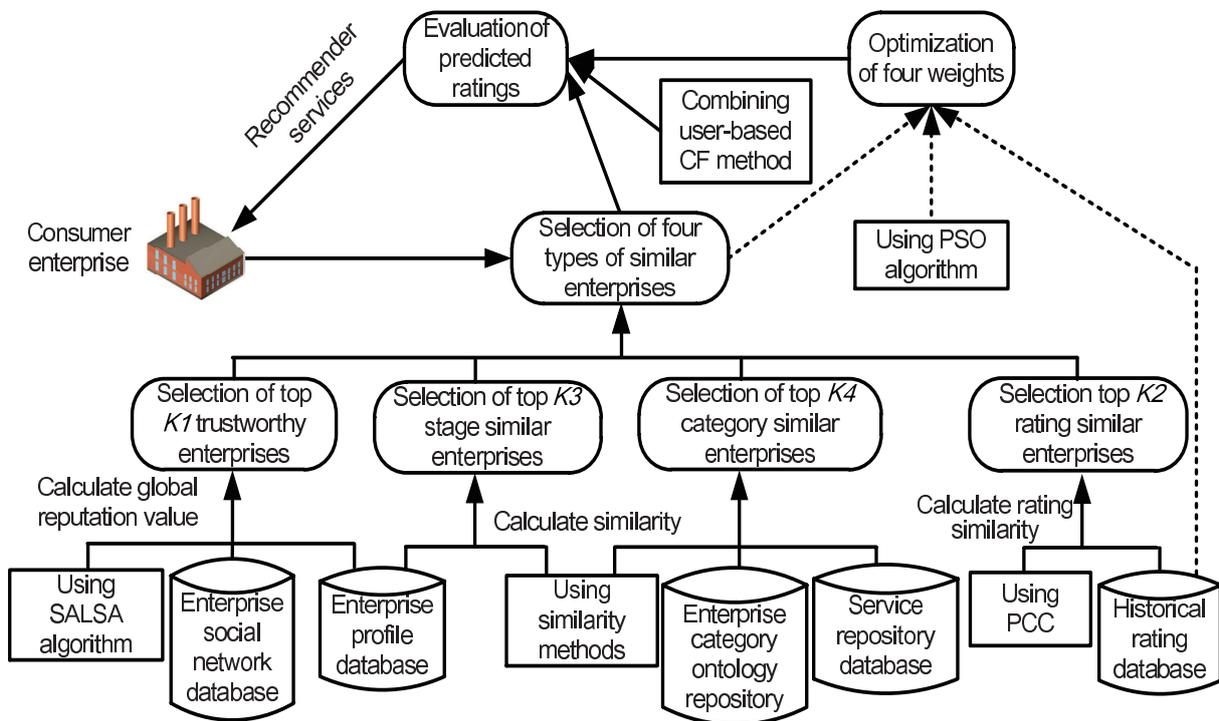


Figure 1: Overview of our proposed method

3.1 Selection of top $K1$ trustworthy enterprises by using SALSA

To select the trustworthy enterprises more accurately, first, the relative trust weight based on social network was identified. Then, SALSA was employed to compute the global reputation value by considering reputation propagation in social networks.

Trust weight identification in a social network

Each enterprise was linked with one or several friend enterprises in social networks and each link carried a corresponding trust weight that was first identified. Inspired by a previous work-also conducted by the present writers-in the domain of attention (DOA) [30], it was established that the more number of times that a consumer enterprise has connected with friend enterprises such as visiting home page and inquiring services, the more trust the consumer enterprise has on a friend enterprise. Therefore, this study considered average connection times as trust weight

and the formulation is represented as follows:

$$tw_{ij} = \frac{ConnTimes(E_i, E_j)}{\sum_{k=1}^K ConnTimes(E_i, E_k)}, \quad (1)$$

where $ConnTimes(E_i, E_j)$ denotes the number of times that enterprise i has connected to enterprise j , and denominator means the total times that enterprise i has connected with all enterprises in social network. The number of connection times can be extracted from the enterprise profile.

Calculation of global reputation value by using SALSA

A large amount of reputation evaluation methods have been widely used to compute reputation value. However, the conventional reputation evaluation methods usually ignore the reputation propagation in enterprise social networks. Largely, SALSA [15] is known as a promising link analysis algorithm, combining advantages of both PageRank [3] and HITS [13]. This study found that the link structure of SALSA is similar to reputation propagation in an enterprise social network, such that it is appropriately adopted to calculate the global reputation values of enterprises based on the trust weight.

In an enterprise social network, the relationships among consumer enterprises can be regarded as a bipartite undirected graph $G = (V_h, V_a, E)$, where V_h and V_a denote a set of hub enterprises (all the enterprises in a social network with out-degree) and a set of authority enterprises (all the enterprises in a social network with in-degree), respectively. E is the set of directed interactions edge between the enterprises. Each edge is arranged with a corresponding trust weight calculated by equation (1). An adjacency matrix \mathbf{M} can be built based on the link structure of the social network. The initial $m_{ij} = 1$ if the enterprise i points to enterprise j , otherwise, $m_{ij} = 0$. Each edge is assigned a trust weight, so that $m_{ij} = 1 \times tw_{ij}$ or 0. The equations of hub matrix \mathbf{H} and authority matrix \mathbf{A} are represented as follows [16]:

$$\mathbf{H} = \mathbf{M}_r \mathbf{M}_c^T, \quad \mathbf{A} = \mathbf{M}_c^T \mathbf{M}_r, \quad (2)$$

where \mathbf{M}_r and \mathbf{M}_c denote each nonzero row divided by row sum of matrix \mathbf{M} and each nonzero column divided by its column sum, respectively.

This research considered the authority value of SALSA as the reputation value of an enterprise. When G is connected, SALSA assigns each page an authority weight, which is proportional to the sum of weights of its incoming edges [16]. However, if G is not connected, the authority value of enterprise i can be represented as follows [16]:

$$a_i = \frac{|A_j|}{|A|} \times \frac{\sum_{j=1}^{|B(i)|} tw_{ij}}{\sum_{j=1}^{|E_j|} tw_{ij}}, \quad (3)$$

where $|A_j|$ and $|A|$ denote the number of enterprises in j th connected component and total number of enterprises in the set of V_a , respectively; $|B(i)|$ and $|E_j|$, respectively, denote the number of in-degree enterprise i and total number of in-degree enterprises in j th connected component; enterprise i belongs to j th connected component.

An example of trust relationship among enterprises in social networks is shown in Fig. 2.

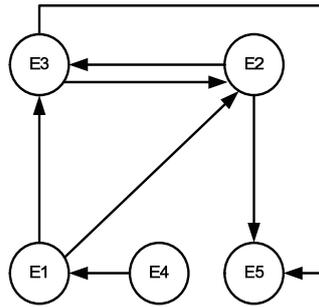


Figure 2: An example of trust relationship in enterprise social network

The connection times among enterprises can be extracted from enterprise profile in table 1.

Table 1: Connection times among enterprises

	E1	E2	E3	E4	E5
E1	0	30	50	0	0
E2	0	0	40	0	10
E3	0	10	0	0	30
E4	10	0	0	0	0
E5	0	0	0	0	0

The trust weights between the enterprises can be computed by using formulation 1, and adjectve matrix \mathbf{M} can be built. The trust relationship in enterprise social network in Fig. 2 can be transformed as bipartite graph in Fig. 3.

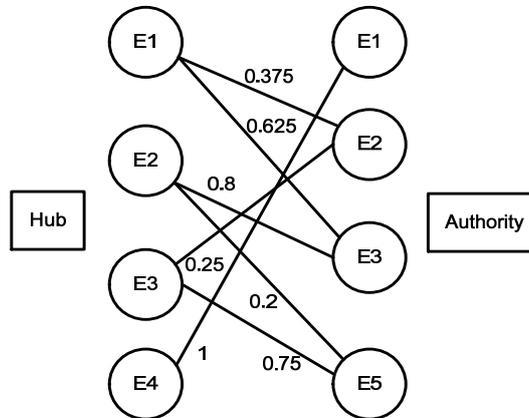


Figure 3: G: bipartite graph of the example

According to the aforementioned formulation 2, we can calculate the hub matrix \mathbf{H} and authority matrix \mathbf{A} in the enterprise social network.

$$\mathbf{M} = \begin{bmatrix} E1 & E2 & E3 & E4 & E5 \\ E1 & 0.000 & 0.375 & 0.625 & 0.000 & 0.000 \\ E2 & 0.000 & 0.000 & 0.800 & 0.000 & 0.200 \\ E3 & 0.000 & 0.250 & 0.000 & 0.000 & 0.750 \\ E4 & 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ E5 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} E1 & E2 & E3 & E4 & E5 \\ E1 & 0.629 & 0.278 & 0.094 & 0.000 & 0.000 \\ E2 & 0.445 & 0.405 & 0.150 & 0.000 & 0.000 \\ E3 & 0.188 & 0.185 & 0.625 & 0.000 & 0.000 \\ E4 & 0.000 & 0.000 & 0.000 & 1.000 & 0.000 \\ E5 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} E1 & E2 & E3 & E4 & E5 \\ E1 & 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ E2 & 0.000 & 0.344 & 0.469 & 0.000 & 0.188 \\ E3 & 0.000 & 0.209 & 0.703 & 0.000 & 0.089 \\ E4 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ E5 & 0.000 & 0.188 & 0.200 & 0.000 & 0.613 \end{bmatrix}$$

According to the structure of matrix \mathbf{H} and matrix \mathbf{A} , G is not connected and the authority value of enterprise i can be calculated by using formulation 3. The result of global reputation values of enterprises can be represented as $[a_1, a_2, a_3, a_4, a_5]^T = [0.250, 0.156, 0.356, 0, 0.238]^T$.

Therefore, this study selected top $K1$ trustworthy enterprises in social networks as an influential component of manufacturing service recommendation.

3.2 Selection of three types of similar enterprises

In addition to trustworthy enterprises, this paper considered the similarity of past consumer enterprises to recommend suitable manufacturing service. Conventional user-based CF method usually only considers rating similarity of past consumer enterprises to predict ratings of candidate manufacturing services.

However, other similar features of enterprises also have significant influence on predicting ratings of candidate manufacturing services. This section discusses the three types of top K similar enterprises-chosen for this study-as the additional three influential components of manufacturing service recommendation, including rating similarity, stage similarity, and category similarity.

Selection of top $K2$ enterprises with similar ratings

Pearson Correlation Coefficient (PCC) [22] is a popular and effective technology to calculate similarity among users based on the historical ratings. In this section, PCC is also leveraged to calculate the rating similarity of consumer enterprises according to their historical ratings to services, and the equation is presented as follows [29]:

$$Rate_Sim(E_a, E_b) = \frac{\sum_{i=1}^I (r_{a,i} - \bar{r}_a) \times (r_{b,i} - \bar{r}_b)}{\sqrt{\sum_{i=1}^I (r_{a,i} - \bar{r}_a)^2} \times \sqrt{\sum_{i=1}^I (r_{b,i} - \bar{r}_b)^2}}, \quad (4)$$

where I means the total number of manufacturing services co-rated by consumer enterprise E_a and E_b ; $r_{a,i}$ and $r_{b,i}$ denote the rating value of manufacturing service i provided by E_a and E_b , respectively; \bar{r}_a and \bar{r}_b mean the average ratings of E_a and E_b , respectively. The value of $Rate_Sim(E_a, E_b)$ will be within the interval $[-1, 1]$, and the larger it is, the higher is the similarity between E_a and E_b . We can select top $K2$ enterprises with similar ratings.

Selection of top $K3$ enterprises with similar development stages

For a manufacturing enterprise, the stage of enterprise development also plays an important role in service selection since the different stages of development have a different focus on manufacturing services. For example, start-up stage enterprises usually pay more attention on price of manufacturing service to reduce costs, while mature enterprise may rather put more emphasis on the quality of manufacturing service to attract more consumers. However, the development stages similarity is usually ignored in the process of manufacturing service recommendation. Therefore, this study considered the development stages of enterprise as an influential factor of manufacturing service recommendation.

According to the lifecycle of enterprise development, each enterprise can be divided into four stages: start-up, growth, maturity, and decline. The stage of each enterprise can be assessed by the Hwang method [11] and we will not repeat here. The equation of stage similarity is shown as follows:

$$Stage_Sim(E_a, E_b) = 1 - \frac{|E_{a,s} - E_{b,s}|}{dec - start}, \quad (5)$$

where $E_{a,s}$ and $E_{b,s}$ denote the stages of enterprises E_a and enterprise E_b , respectively; dec and $start$ mean the start-up stage and declining stage, respectively. We define the four stages as real numbers: 1, 2, 3, and 4. The top $K3$ enterprises with similar development stage were selected.

Selection of top $K4$ enterprises with similar enterprise category

The category of manufacturing enterprise can be divided into three levels, namely broad heading, medium-class, and subclass. The broad heading can be further segmented into 25

types, including food manufacturing, tobacco manufacturing, and medical manufacturing. It can be further decomposed into medium-class and subclass. The semantic similarity of enterprise category is an indispensable factor for recommending manufacturing service. Food manufacturing enterprise, for instance, and medical manufacturing enterprise have completely different demands in terms of manufacturing services.

A previous work [4]-by these writers-explored a rich body of OWL-based manufacturing service ontologies, and it was referenced to develop the ontology of enterprise category to calculate the semantic similarity of this category. The particle of the ontology is illustrated in Fig. 4. The formulation of the semantic similarity of enterprise category can be represented as follows [4]:

$$\begin{aligned}
 \text{Category_Sim} = & \mu \frac{n(C_{\text{super}}(E_a, O) \cap C_{\text{super}}(E_b, O))}{n(C_{\text{super}}(E_a, O) \cup C_{\text{super}}(E_b, O))} \\
 & + \nu \frac{n(C_{\text{midd}}(E_a, O) \cap C_{\text{midd}}(E_b, O))}{n(C_{\text{midd}}(E_a, O) \cup C_{\text{midd}}(E_b, O))} + \lambda \frac{n(C_{\text{sub}}(E_a, O) \cap C_{\text{sub}}(E_b, O))}{n(C_{\text{sub}}(E_a, O) \cup C_{\text{sub}}(E_b, O))},
 \end{aligned} \tag{6}$$

where $n(C_{\text{super}}(E_a, O) \cap N_{\text{super}}(E_b, O))$ means the number of the broad heading to which both enterprise a and enterprise b belong within ontology. $n(C_{\text{super}}(E_a, O) \cup N_{\text{super}}(E_b, O))$ denotes the number of the broad heading to which either enterprise a or enterprise b belongs within ontology. By that analogy, the similarity of the medium-class and subclass of enterprise category can be computed, and $\mu + \nu + \lambda = 1$. According to the result of semantic similarity of manufacturing enterprise category, the top K_4 enterprises with similar location were selected.

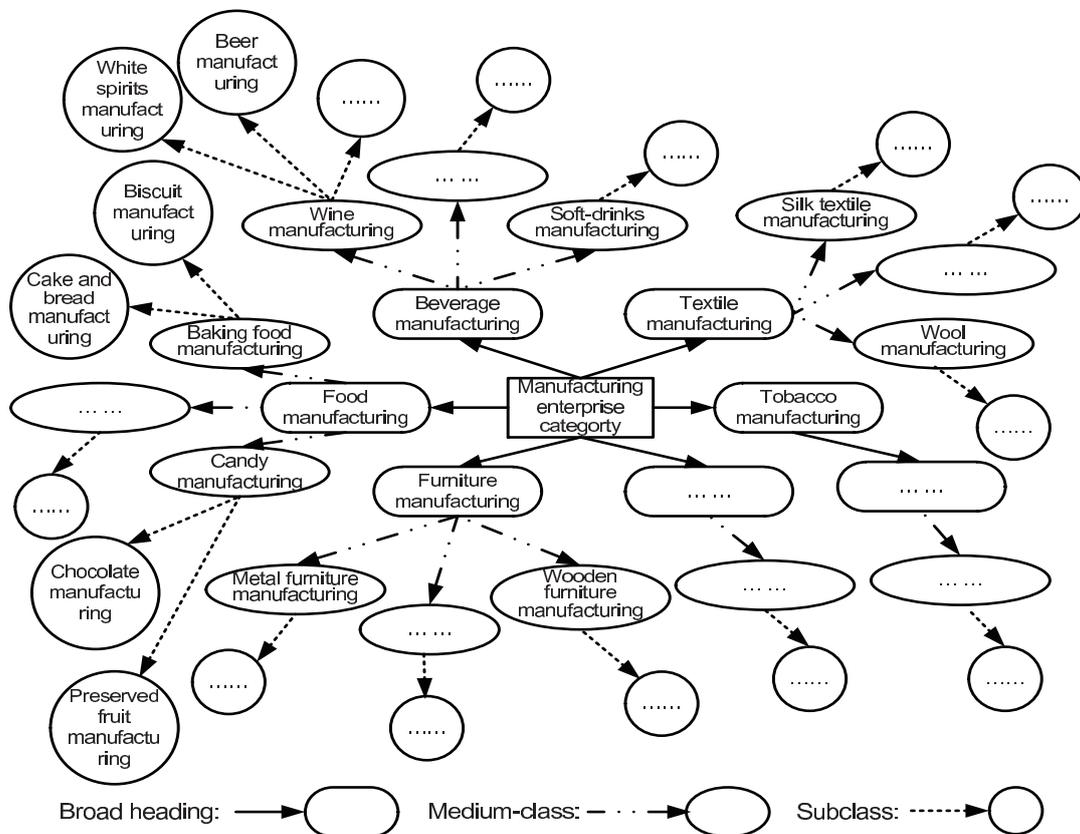


Figure 4: Particle of the ontology of manufacturing enterprise category

3.3 Four weights optimization by using PSO

The weight optimization of the four influential components is inspired by genetic algorithm (GA)-based learning method [10], which was employed to determine the degree of importance of corresponding criteria of electronic service of Internet banking.

However, PSO algorithm is more efficient than GA, since PSO has no complicated operators, including selection, crossover, and mutation. Therefore, the study adopted the PSO algorithm to determine the objective weights of the four influential components to calculate the predicted ratings of target manufacturing services.

Fitness function

Based on the principle of PSO algorithm, the study first constructed the fitness function to judge the performance of positions of particles. The main ideology behind constructing a fitness function is to compare actual average rating of services, which the query enterprise has consumed and rated, with predicted rating of past consumer enterprises. The actual average rating of J services that was rated by consumer enterprise is denoted as \overline{R}_a . The equation for predicted average rating of J services rated by consumer enterprise is represented as follows:

$$PAR_J = \frac{1}{J} \times \sum_{j=1}^J PR(E_a, S_j), \quad (7)$$

where J means total number of services rated by enterprise E_a . $PR(E_a, S_j)$ denotes the predicted rating of the j th service using user-based CF, and it can be calculated as following formulation:

$$PR(E_a, S_j) = \sum_{i=1}^4 \alpha_i \times \frac{\sum_{ki=1}^{K_i} R(E_{ki}, S_j) \times comp_i(E_a, E_{ki})}{\sum_{ki=1}^{K_i} comp_i(E_a, E_{ki})}, \quad (8)$$

where K_i means i th set of top K enterprises (i.e., K_1 , K_2 , K_3 and K_4); $comp_i(E_a, E_{ki})$ is expressed as i th component calculated by the four equations (3-6); $R(E_a, S_j)$ is the rating provided by k th enterprise E_{ki} in top K_i enterprises; and α_i denotes corresponding weight which will be optimized.

Through comparison of actual average rating and predicted average rating, the distance between them seems closer; hence, the performance of the position of the particle is better. The final fitness function was constructed as follows:

$$fv_a = \frac{1}{1 + |\overline{R}_a - PAR_J|}, \quad (9)$$

where fv_a denotes the fitness value of the four weights, and \overline{R}_a means average rating of services that the consumer enterprise has rated.

Identification of optimal weights of each influential component

The four influential components are regarded as four particles in corresponding four-dimensional space and the range of each axis was set on interval $[0, 1]$. Each particle has a velocity to update its position in its corresponding dimensional space. The position of the particle with highest fitness value was set as the global best position (gd) in each iteration. The i th particle kept its best personal position (p_{id}) which was visited in the corresponding d th dimension of search space. The velocity and position of each particle were updated as follows [27]:

$$v_{id}^{t+1} = \omega v_{id}^{t+1} + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t), \quad (10)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \quad (11)$$

where ω means a random inertia weight between 0 and 1; v_{id}^t denotes the velocity of i th particle in generation t ; c_1 and c_2 are two positive constants; r_1 and r_2 are two random numbers among 0 and 1; and x_{id}^t means the current position of i th particle in generation t .

Through multiple iterations, the best personal positions of the four particles in their corresponding dimensional spaces were obtained. The weights of four influential components were normalized as follows:

$$\alpha_i = \frac{x_{id}}{\sum_{i=1}^4 x_{id}}, \quad (12)$$

where x_{id} denotes the best personal position of i th particle in corresponding d th space.

3.4 Manufacturing service recommendation

The missing rating of manufacturing service Sr which consumer enterprise Ea needs can be predicted using a user-based CF method. The final equation of manufacturing service recommendation can be referred to equation (8), and the corresponding weight α_i was objectively obtained through the optimization process that is illustrated in sub-section 3.3 in detail. Accordingly, the predicted missing ratings of various manufacturing services, the top k manufacturing services with higher predicted ratings, can be recommended to the consumer enterprise.

4 An evaluation experiment

In this section, a comparison experiment to evaluate the quality of manufacturing service recommendation was conducted through the use of the study's proposed hybrid method. Evaluation metric of mean absolute error (MAE) was employed to evaluate the accuracy of our proposed hybrid method. The formulation is expressed as follows [29]:

$$MAE = \frac{\sum_{t=1}^T |p_t - a_t|}{T}, \quad (13)$$

where T means the total number of predictions, and p_t and a_t denote the t th predicted rating and t th actual rating, respectively.

4.1 Comparative methods

Two recommender methods were selected to compare with the proposed hybrid method of the study to demonstrate the accuracy of the method. These are:

- (1) Traditional user-based CF method, which only considers rating similarity between past consumer enterprises to predict ratings of candidate services.
- (2) Combined social network and CF method [29], which is an effective method for manufacturing service recommendation, while it has not considered reputation propagation in enterprise social network, and only subjectively assigned relative weights of the influential components.

4.2 Comparison results

In the historical rating registry, there are 8796 ratings of 372 manufacturing services provided by 156 consumer enterprises with their relative trust information and profiles. The ratings varied from 1 to 5, and 80% of the rating data were randomly selected to act as training set and the rest were represented as testing set.

Fig. 5 demonstrates the three sets of MAE values calculated by using three recommender methods with different sizes of neighbors from 5 to 17. According to the results of experiments, the MAE values calculated by our proposed method are lower than the other two methods. Therefore, our proposed method in this study is more accurate than traditional user-based CF method and the combination of the social network and CF method. The results also reveal that the performance of our proposed method is the best at the point of 11 in neighbor size.

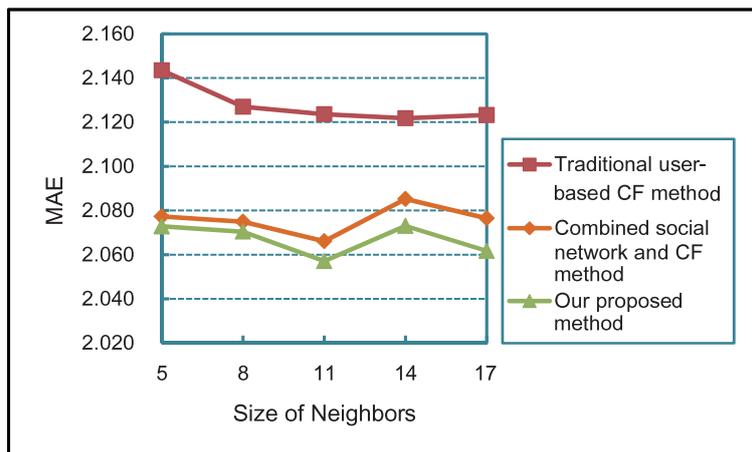


Figure 5: Comparison of MAE values of three methods

5 Conclusion

In this paper, a hybrid social network-based CF method is proposed to make manufacturing service recommendation more accurate and objective. The result of the evaluation experiment illustrates that the proposed hybrid method is more accurate than the other traditional recommender methods.

To sum up, the major contributions of this paper are as follows: first, trustworthy enterprises are obtained by identifying trust weight and leveraging SALSA that considers reputation propagation based on social network in the process of calculating reputation value; second, the trustworthy enterprises and three types of similar enterprises with different features are considered as four influential components for calculating predicted ratings of candidate services to enhance the quality of manufacturing service recommendation; third, personalized manufacturing service can be recommended by using an extended user-based CF method where PSO algorithm is employed to automatically and objectively obtain the weights of four influential components.

Still, this paper has some limitations, which can be further explored in future research. For instance, more useful information on enterprise social networks can be mined to enhance the accuracy of manufacturing service recommendation.

Acknowledgement

The work has been supported by National Natural Science Foundation of China (No. 51475410, No. 51375429) and Zhejiang Natural Science Foundation of China (No. LY17E050010).

Bibliography

- [1] Adomavicius G., Tuzhilin A.(2005); Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749, 2005.
- [2] Borodin A., Roberts G.O., Rosenthal J.S. et al. (2001); Finding authorities and hubs from link structures on the world wide web, *In Proceedings of the 10th International Conference on World Wide Web*, ACM, Hong Kong, China, 415–429, 2001.
- [3] Brin S., Page L. (1998); The anatomy of a large-scale hypertextual web search engine, *Computer Networks and ISDN Systems*, 30(1-7), 107–117, 1998.
- [4] Cai M., Zhang W.Y., Zhang K.(2011); ManuHub: A semantic web system for ontology-based service management in distributed manufacturing environments, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(3), 574–582, 2011.
- [5] Colorni A., Dorigo M., Maffioli F., et al. (1986); Heuristics from nature for hard combinatorial optimization problems, *International Transactions in Operational Research*, 3(1), 1-21, 1986.
- [6] Deng S. G., Huang L. T., Xu G. D.(2014); Social network-based service recommendation with trust enhancement, *Expert Systems with Applications*, 41(18), 8075–8084, 2014.
- [7] Duke A., Davies J., Richardson M. (2005); Enabling a scalable service-oriented architecture with semantic Web Services, *BT Technology Journal*, 23(3), 191–201, 2005.
- [8] Eirinaki M., Louta M. D., Varlamis I. (2014); A trust-aware system for personalized user recommendations in social networks, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(4), 409-421, 2014.
- [9] Esfahani M. T., Torabia S. H., Vahidi B. (2015); A new optimal approach for improvement of active power filter using FPSO for enhancing power quality, *International Journal of Electrical Power & Energy Systems*, 69, 188-199, 2015.
- [10] Hu Y. C., Liao P. C. (2011); Finding critical criteria of evaluating electronic service quality of Internet banking using fuzzy multiple-criteria decision making, *Applied Soft Computing*, 11(4), 3764-3770, 2011.
- [11] Hwang Y. S.(2004); *The evolution of alliance formation: an organizational life cycle framework*, Diss. Rutgers University, 2004.
- [12] Kennedy J., Eberhart R. (1995); Particle swarm optimization, *In Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, Australia, 1942–1948, 1995.
- [13] Kleinberg J. M. (1999); Authoritative sources in a hyperlinked environment, *Journal of the ACM (JACM)*, 46(5), 604–632, 1999.
- [14] Langville A. N., Meyer C. D. (2005); A survey of eigenvector methods for web information retrieval, *Society for Industrial and Applied Mathematics (SIAM) Review*, 47(1), 135–161, 2005.
- [15] Lempel R., Moran S. (2000); The stochastic approach for link-structure analysis (SALSA) and the TKC effect, *Computer Networks*, 33(1-6), 387-401, 2000.

-
- [16] Lempel R., Moran S. (2001); SALSA: the stochastic approach for link-structure analysis, *ACM Transactions on Information Systems (TOIS)*, 19(2), 131–160, 2001.
- [17] Liu J. T., Wu C. H., Liu W. Y. (2013); Bayesian probabilistic matrix factorization with social relations and item contents for recommendation, *Decision Support Systems*, 55(3), 838–850, 2013.
- [18] Najork M., Gollapudi S., Panigrahy R. (2009); Less is more: sampling the neighborhood graph makes salsa better and faster, *Proceedings of the 2th ACM International Conference on Web Search and Data Mining*, ACM, Barcelona, Spain, 242–251, 2009.
- [19] Park J. B., Jeong Y. W., Shin J. R., et al. (2010); An improved particle swarm optimization for nonconvex economic dispatch problems, *IEEE Transactions on Power Systems*, 25(1), 156–166, 2010.
- [20] Perugini S., Goncalves M. A., Fox E. A. (2004); Recommender systems research: A connection-centric survey, *Journal of Intelligent Information Systems*, 23(2), 107–143, 2004.
- [21] Rahuman M. S. (2012); Improved web link analysis using community based popularity approach, *Proc. of the 2th Intl. Conf. on Computing, Communication and Information Technology*, Hammamet, Tunisia, 41-44, 2012.
- [22] Rodgers J. L., Nicewander W. A. (1988); Thirteen ways to look at the correlation coefficient, *The American Statistician*, 42(1), 59-66, 1988.
- [23] Salakhutdinov R., Mnih A. (2008); Bayesian probabilistic matrix factorization using Markov Chain Monte Carlo, *Proc. of the 25th Intl. Conf. on Machine Learning*, Helsinki, Finland, 880-887, 2008.
- [24] Sobiecki J. (2014); Comparison of selected swarm intelligence algorithms in student courses recommendation application, *International Journal of Software Engineering and Knowledge Engineering*, 24(1), 91-109, 2014.
- [25] Sun Z. B., Han L. X., Huang W. L., et al. (2015); Recommender systems based on social networks, *Journal of Systems and Software*, 99, 109-119, 2015.
- [26] Tyagi S., Bharadwaj K. K. (2013); Enhancing collaborative filtering recommendations by utilizing multi-objective particle swarm optimization embedded association rule mining, *Swarm and Evolutionary Computation*, 13, 1-12, 2013.
- [27] Wang Y. J., Yang Y. P. (2009); Particle swarm optimization with preference order ranking for multi-objective optimization, *Information Sciences*, 179(12), 1944-1959, 2009.
- [28] White S., Smyth P. (2003); Algorithms for estimating relative importance in networks, *Proc. of the 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, Washington D.C., USA, 266-275, 2003.
- [29] Zhang W. Y., Zhang S., Chen Y. G., et al. (2013); Combining social network and collaborative filtering for personalised manufacturing service recommendation, *International Journal of Production Research*, 51(22), 6702-6719, 2013.
- [30] Zhu Y., Zhang S., Wang Y., et al. (2013); A social network-based expertise-enhanced collaborative filtering method for e-government service recommendation, *Advances in Information Sciences and Service Sciences*, 5(10), 724-735, 2013.

Author index

Bojan-Dragos C.-A., 599

El-Sayed H., 612

Hergane S., 599

Jiang W., 631

Jin H.-W., 645

Kim K., 645

Lee S., 645

Lee S.-H., 645

Li P., 661

Liu Y., 689

Mohammadi S., 677

Namadchian A., 677

Peng L., 689

Precup R.-E., 599

Preitl S., 599

Sadło P., 704

Sangeetha N., 715

Sankar S., 612

Schoeneich R.O., 704

Tanasoiu O.-M., 599

Thandavarayan G., 612

Tomescu L.M., 599

Ushabhanu N., 715

Vani R., 715

Wang S., 631

Wu C.X., 661

Xu S., 728

Yang W.T., 728

Yu H., 612

Yu X.W., 661

Zhang S., 728

Zhang S.Z., 661

Zhang W.Y., 728

Zhong H.D., 661