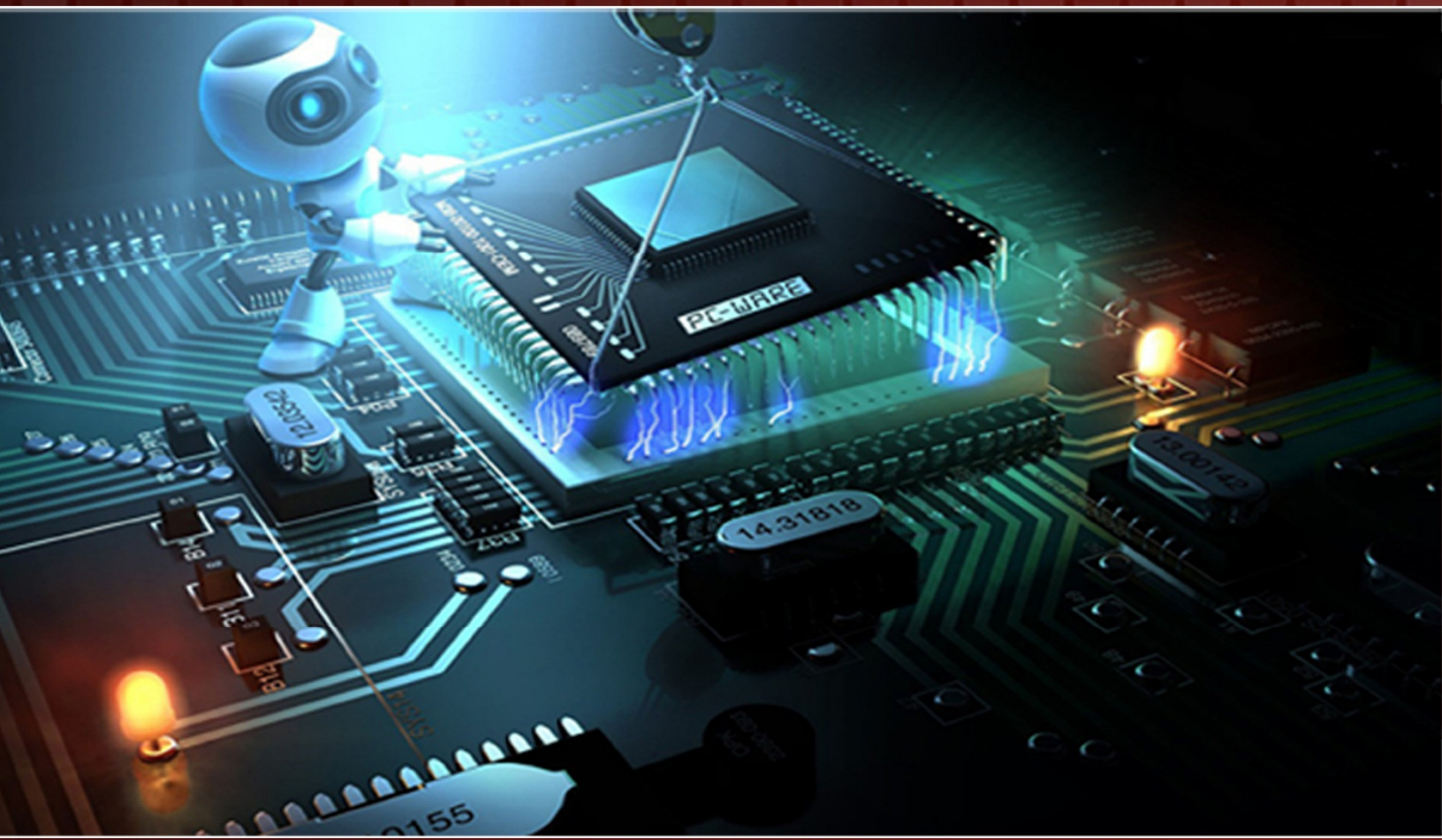
ISSN: 2054-7390

Volume 5 Issue 6, Dec 2017

Transactions on Machine Learning and Artificial Intelligence



Society for Science and Education
UNITED KINGDOM

TABLE OF CONTENTS

EDITORIAL ADVISORY BOARD

DISCLAIMER

All the contributions are published in good faith and intentions to promote and encourage research activities around the globe. The contributions are property of their respective authors/owners and the journal is not responsible for any content that hurts someone's views or feelings etc.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Volume 5 No. 6
ISSN 2054-7390

A Metaheuristic Procedure for Calculating Optimal Osmotic Dehydration Parameters: A Case Study of Mushrooms

[1] Julian Scott Yeomans
[1] OMIS Area, Schulich School of Business, York University, Toronto, ON, M3J 1P3 Canada;
syeomans@schulich.yorku.ca

ABSTRACT

The Firefly Algorithm (FA) metaheuristic is employed to determine the optimal parameter settings in a case study of the osmotic dehydration of mushrooms. In the case, the functional form of the dehydration model is established through a response surface technique and the resulting mathematical programming is formulated as a non-linear goal programming model. For optimization purposes, a computationally efficient, FA-driven method is used and the resulting optimal process parameters are shown to be superior to those from previous approaches.

Keywords: Biologically-inspired Metaheuristics, Mushrooms, Non-linear Goal Programming, Process Parameter Optimization, Osmotic Dehydration.

1 Introduction

Commercial agronomy represents a multi-billion dollar, worldwide enterprise. Within this industry, the annual global production of mushrooms currently exceeds 6 million tonnes [1]. As with many fresh fruits and vegetables, the high moisture content of mushrooms renders them highly perishable and, due to various enzymatic, microbial and chemical reactions, they commence deterioration immediately upon harvesting [2] [3]. Therefore, it becomes imperative to determine effective preservation methods that retain the overall quality and desirable features of the product. The preservation of many agricultural commodities has often been accomplished by employing various combinations of drying using heat processing and dehydration [2] [3] [4]. The dehydration of fresh produce generally extends their storage lives, decreases their shipping weights, and reduces the need for special packing requirements [2]. However, hot-air dried products conventionally processed using tray, vacuum, or cabinet dryer techniques have not received widespread acceptance due to the perceived diminished quality of the endproduct [2] [4] [5].

Recently, osmotic dehydration has been introduced as a practical alternative preservation approach that is capable of producing a higher quality final product [6]. In osmotic dehydration, fresh produce is immersed in a hypertonic solution where the water content from the cells of the produce is transferred into the solution due to the relative differences in their solute concentrations [6]. In this processing, osmotic dehydration removes a desired portion of the water from within the fresh produce resulting in a product of intermediate moisture content [7] [8]. Simultaneously, a corresponding transfer of solid materials (normally sugar and/or salt) occurs from the solution into the product [6] [9] [10]. In terms of

DOI: 10.14738/tmlai.56.3727
Publication Date: 1st October, 2017
URL: http://dx.doi.org/10.14738/tmlai.56.3727

final product quality relative to standard hot air drying methods, osmotic dehydration causes only minimal thermal degradation due to the low temperatures involved [6] [11] [12] [13].

Osmotic dehydration of fresh agricultural commodities can also be used as a pre-treatment to supplemental dry-processing because it improves many sensory, functional and nutritional properties [14]. The quality of the subsequent product is superior to one without pre-treatment due to (i) the improvements to texture of the fruits and vegetables, (ii) the stability of the colour pigmentation during storage, and (iii) increases in the solid gain transfer of sugar and salt from the hypertonic solution [2] [11]. Thus, in conjunction with other ensuing drying technologies, osmotic dehydration produces a superior quality, shelf-stable product for both local consumption and export markets.

Water removal during the dehydration process is influenced by many factors such as type and concentration of osmotic agents, temperature, circulation/agitation of solution, solution-to-sample ratio, thickness of food material, and any pre-treatments [4] [6]. While an expanding market currently exists for osmo-convective dehydrated fruits and vegetables in both domestic and world markets, only limited efforts have been undertaken to optimize the requisite osmotic process parameters [2] [11] [15] [16] [17] [18]. Specifically, an effective analysis of the mass transport occurring within the osmosis process measured in terms of water loss and solid (sugar, salt) gains is of considerable commercial and practical relevance [4] [6] [10].

In this study, the functional form of the osmotic dehydration process for mushrooms is constructed using a standard response surface technique [17] [18] [19] [20] [21]. The format of the resulting optimization model is shown to be a non-linear goal programming problem [15] [16] [17] [18]. This study employs the Firefly Algorithm (FA) [17] [22] [23] [24] to determine the optimal osmotic parameters for the mushroom dehydration case considered in [2]. It can be shown that the resulting osmotic process parameters produced by the FA are superior to those from the previous approaches.

2 Functional Form and Mathematical Model of the Osmotic Dehydration Process

The first section of the analysis examines the dehydration case of mushrooms taken from [2]. In the case, a brine solution is employed for dehydration and the solid gain corresponds to the transport of salt from the brine into the mushrooms. The first step requires the construction of an appropriate model of the responses to the three main osmotic process parameters – (i) solution temperature, (ii) hypertonic solution concentration and (iii) duration of osmosis – on the water loss and solid gain of the mushrooms. This functional representation can then be used to predict the water loss and salt gain impacts in the mushrooms over the requisite experimental ranges of the three designated parameters. Once the appropriate model has been constructed, the next step is to optimize this model in order to determine the maximum water loss and the optimum salt gain achieved during dehydration. In the subsequent formulations, let T represent the brine solution temperature in ^{o}C, C be the salt solution concentration in percent, and D be the duration of the osmosis measured in minutes. For the response variables, let WL be the percentage of water loss and SG represent the solid gain of the product during the dehydration process. In this instance, SG corresponds to the percentage of salt gain in the mushrooms.

Response surface methods are statistical techniques frequently used for optimization in empirical studies [19] [20] [21]. Response surfaces employ quantitative data in appropriately designed experiments to

simultaneously ascertain the various variable relationships within multivariate problems [21]. The equations constructed describe the effect of various test variables on responses, determine interrelationships among the test variables and represent the combined effect of all test variables in any response. Response surfaces enable an experimenter to undertake an efficient exploration of a process or system [20] [21]. These approaches have frequently been used in the optimization of food processes [2] [11] [25] [26] [27] [28] [29] and will, consequently, be employed in this study to determine the appropriate mathematical representation. The proposed model can then be used to predict the water loss and salt gain in the dehydration of mushrooms over the different experimental ranges for the process durations, brine concentrations and solution temperatures.

For the osmotic dehydration process, it should be noted that the exact mathematical representation for the relationship between the parameters remains unknown. Thus, a response surface methodology enables an empirical approximation to it using efficient experimental design techniques [20] [21]. The specific testing design actually contains the three variables (*T*, *C*, *D*) each set at three levels using the data taken from [2] in order to determine the corresponding water loss (*WL*) and salt gain (*SG*) responses. The design for the various combinations of input variables and levels requires the various experimental combinations shown in Table 1 (see [2]), while the values determined for the response variables *WL* and *SG* appear in last two columns of Table 1.

Based upon the response surface experimental design appropriately applied to the water loss and the salt gain outputs of TABLE 1 [19] [20] [21], the functional equations empirically determined for responses are:

$$WL = 19.58 - 0.13T + 1.7C + 0.98D + 0.00357TD + 0.00673CD - 343C^2 - 0.0106D^2 \quad (1)$$

$$SG = -13.87 + 0.11T + 1.09C + 0.14D - 0.000973T^2 - 0.0296C^2 - 0.00129D^2 \quad (2)$$

Mehta et al. [2] established organoleptic ranges for the osmotic dehydration parameters and restricted their search for best parameter settings to values within these ranges. Organoleptic properties refer to sensory aspects of food including taste, sight, smell, touch, dryness, moisture content, and stale-fresh factors. In order to find values for the osmotic dehydration parameters, Mehta et al. [2] constructed a number of contour plots by varying the values of the three variables and observed the effect that these had on their response functions. By superimposing these contours onto a single chart, the best settings for the temperature, concentration, and duration variables were determined to be 44.89 ^{o}C, 16.53% and 47.59 minutes, respectively. These settings generate responses of 40.55% for water loss and 2.98% for salt gain (see Table 2).

Table 1. Response Surface Experimental Design Layout for 3 Variables and 3 Levels.

Level for T	Temperature (°C)	Level for C	Concentration (%)	Level for D	Duration (Mins)	Water Loss (%)	Salt Gain (%)
1	55	1	20	0	45	44.93	3.24
1	55	-1	10	0	45	36.38	1.03
-1	35	1	20	0	45	39.70	2.56
-1	35	-1	10	0	45	29.92	0.59
1	55	0	15	1	60	43.92	2.90
1	55	0	15	-1	30	34.23	2.24
-1	35	0	15	1	60	37.09	2.34
-1	35	0	15	-1	30	29.54	1.73
0	45	1	20	1	60	45.04	3.03
0	45	1	20	-1	30	35.51	2.22
0	45	-1	10	1	60	33.69	1.06
0	45	-1	10	-1	30	26.18	0.33
0	45	0	15	0	45	38.05	2.57
0	45	0	15	0	45	38.44	2.64
0	45	0	15	0	45	38.27	2.64
0	45	0	15	0	45	38.55	2.79
0	45	0	15	0	45	38.60	2.82

Table 2. Best Osmotic Dehydration Parameters Determined by Mehta et al. [2].

Temperature (°C)	Concentration (%)	Duration (Mins)	Water Loss (%)	Salt Gain (%)
44.89	16.53	47.59	40.55	2.98

3 A Goal Programming Formulation for Setting Osmotic Dehydration Parameters

The determination of the parameters settings can be viewed as a multi-response optimization process and could, therefore, be transformed into a corresponding mathematical programming model [15] [16] [17] [18]. In this section, this formulation will be accomplished by converting the parameter setting process into an equivalent goal programming format.

Based upon the organoleptic requirements established for the parameters and response functions in [2], the technical constraints for the problem can be specified as:

$$26.18 \le WL \le 45.04 \tag{3}$$

$$0.33 \le SG \le 3.24 \tag{4}$$

$$35 \le T \le 55 \tag{5}$$

$$10 \le C \le 20 \tag{6}$$

$$30 \le D \le 60 \tag{7}$$

Additional organoleptic preferences can be applied to the responses and variables for the solution. The targets for these desired criteria are summarized in TABLE 3. From a hierarchical preference attainment perspective, several of these criteria can be recognized as more important attributes to achieve than the others. Namely, from a dehydration perspective, the water loss should be as high as possible within the indicated range, while from a taste perspective, the salt gain needs to be as close to 2.98% as possible. The relative importance for the achievement of these hierarchy targets is indicated in the last column of Table 3.

Table 3. Ranges for Process Variables and Response Goals in the Osmotic Dehydration

Parameter	Goal	Requirement	Lower Limit	Upper Limit	Relative Importance
Temperature (°C)	1	Minimize	35	55	Important
Concentration (%)	2	Minimize	10	20	Important
Duration (Mins)	3	Minimize	30	60	Important
Water Loss (%)	4	Maximize	23.02	44.05	Very Important
Salt Gain (%)	5	Target = 2.98	0.33	3.24	Very Important

Hence, from a mathematical perspective, each of these desired targets can be specified as a definitive goal and the entire formulation can then be transformed into a conventional goal programming problem. An objective function that appropriately penalizes deviations from the desired targets must be created and, in the subsequent mathematical programming formulation, a percentage deviation objective weighted by the relative importance of each goal is employed. Consequently, the problem of determining osmotic dehydration parameter values can be transformed into the following non-linear goal programming formulation.

Minimize $$W_1 * P_1 + W_2 * P_2 + W_3 * P_3 + W_4 * N_4 + W_5 * (P_5 + N_5) \quad (8)$$

subject to

$$P_1 = T - 35 \quad (9)$$

$$N_1 = 55 - T \quad (10)$$

$$P_2 = C - 10 \quad (11)$$

$$N_2 = 20 - C \quad (12)$$

$$P_3 = D - 30 \quad (13)$$

$$N_3 = 60 - D \quad (14)$$

$$P_4 = WL - 26.18 \quad (15)$$

$$N_4 = 45.04 - WL \quad (16)$$

$$P_5 = SG - 2.98 \quad (17)$$

$$N_5 = 2.98 - SG \quad (18)$$

$$P_6 = SG - 0.33 \quad (19)$$

$$N_6 = 3.24 - SG \quad (20)$$

$$P_i \geq 0, N_i \geq 0 \qquad i = 1, 2, 3, 4, 5, 6 \quad (21)$$

In order to complete the transformation of the problem into the series of defined goals, several additional deviation variables have been introduced. Namely, for the goal model, define *Pi* and *Ni*, *i* = 1 to 6, to be the positive and negative deviations, respectively, from the disparate goal targets and constraint limits shown for the variables in Table 3. Let *Wi* correspond to weighting factors applied to goal *i*, *i* = 1 to 5, to reflect the relative importance in achieving that goal's target. Each *Wi* also contains the appropriate denominator constant required to transform the deviation variables into the requisite percentage deviation value format. Thus, solving the goal programming model would be equivalent to determining optimal parameter values for the osmotic dehydration process.

4 A Goal Programming, Firefly Algorithm-driven Optimization Approach

While numerous different techniques could have been used to solve the resulting optimization problem, the method actually applied uses an FA procedure. For optimization, Yang [23] has proved that the FA is more computationally efficient than other such commonly-used metaheuristics as simulated annealing, enhanced particle swarm optimization, and genetic algorithms. Thus, the FA represents a very computationally efficient solution procedure. This section provides a brief outline of the FA procedure, while more comprehensive descriptions can be found in [22] and [23].

The FA is a population-based, nature-inspired metaheuristic in which each firefly within the population corresponds to one potential solution to the problem. All FA procedures employ three specific rules: (i) The fireflies within a population are unisex, so that one firefly will be attracted to other fireflies irrespective of their sex; (ii) Attractiveness between any two fireflies is proportional to their brightness, implying that the less bright firefly will move towards the brighter one; and (iii) The explicit brightness of any firefly is explicitly determined by the corresponding value of its objective function. For maximization problems, the brightness can be considered proportional to the value of the objective function. Yang (2010) demonstrates that the FA approaches the global optima whenever the number of fireflies $n \rightarrow \infty$ and the number of iterations t, is set so that t >>1. In reality, the FA has been shown to converge extremely quickly into both local and global optima [22] [23]. The basic operational steps of the FA are summarized in the following pseudo-code (Yang 2010).

Objective Function $F(\boldsymbol{X})$, $\boldsymbol{X} = (x_1, x_2, \ldots x_d)$

Generate the initial population of n fireflies, $\boldsymbol{X}_i$, $i = 1, 2, \ldots, n$

Light intensity I_i at $\boldsymbol{X}_i$ is determined by $F(\boldsymbol{X}_i)$

Define the light absorption coefficient γ

while (t < MaxGeneration)

for i = 1: n, all n fireflies

for j = 1: n, all n fireflies (inner loop)

if ($I_i < I_j$), Move firefly i towards j; **end if**

Vary attractiveness with distance r via $e^{-\gamma r}$

end for j

end for i

Rank the fireflies and find the current global best solution $\boldsymbol{G}^*$

end while

Postprocess the results

In the FA, there are two important issues to resolve: the formulation of attractiveness and the variation of light intensity. For simplicity, it can always be assumed that the attractiveness of a firefly is determined by its brightness which in turn is associated with its encoded objective function value. In the simplest case, the brightness of a firefly at a particular location $\boldsymbol{X}$ would be its calculated objective value $F(\boldsymbol{X})$. However, the attractiveness, β, between fireflies is relative and will vary with the distance r_{ij} between firefly i and firefly j. In addition, light intensity decreases with the distance from its source, and light is also absorbed in the media, so the attractiveness needs to vary with the degree of absorption. Consequently, the overall attractiveness of a firefly can be defined as

$$\beta = \beta_0 \exp(-\gamma r^2)$$

where β_0 is the attractiveness at distance $r = 0$ and γ is the fixed light absorption coefficient for the specific medium. If the distance r_{ij} between any two fireflies i and j located at $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, respectively, is calculated using the Euclidean norm, then the movement of a firefly i that is attracted to another more attractive (i.e. brighter) firefly j is determined by

$$\boldsymbol{X}_i = \boldsymbol{X}_i + \beta_0 \exp(-\gamma (r_{ij})^2)(\boldsymbol{X}_i - \boldsymbol{X}_j) + \alpha \boldsymbol{\varepsilon}_i.$$

In this expression of movement, the second term is due to the relative attraction and the third term is a randomization component. Yang [23] indicates that α is a randomization parameter normally selected within the range [0,1] and $\boldsymbol{\varepsilon}_i$ is a vector of random numbers drawn from either a Gaussian or uniform (generally [-0.5,0.5]) distribution. It should be explicitly noted that this expression represents a random walk biased toward brighter fireflies and if $\beta_0 = 0$, it becomes a simple random walk. The parameter γ characterizes the variation of the attractiveness and its value determines the speed of the algorithm's convergence. For most applications, γ is typically set between 0.1 to 10 [23].]. For the computational approaches for the FA considered in this study, the variation of attractiveness parameter γ is fixed at 5 while the randomization parameter α is initially set at 0.6, but is then gradually decreased to a value of 0.1 as the procedure approaches its maximum number of iterations (see [23]).

Optimizing the goal programming problem using the FA-driven procedure, the best process parameters for the osmotic dehydration of the mushrooms were calculated. The resulting values are shown in Table 4. Comparing these values to those found by Mehta et al. [2], it can be seen that the salt concentration increases by 2.5%, the required temperature increases by 9oC, while the duration of dehydration remains essentially unchanged. In terms of the two key response variables, the resulting water loss increases by 4.5%, while the salt gain remains at its desired organoleptic target of 2.98%. Consequently, since the water loss response – which is obviously the fundamental feature of the osmotic dehydration process – has been increased significantly from that determined in [2], this goal programming solution provides a significant improvement.

Table 4. Optimal Process Parameters Determined for the Osmotic Dehydration of Mushrooms

		Temperature (°C)	Concentration (%)	Duration (Mins)	Water Loss (%)	Salt Gain (%)
Mehta *et al*. [2]		44.89	16.53	47.59	40.55	2.98
FA Solution		54.043	19.031	46.777	45.04	2.98

In any given optimization problem, for a very large number of fireflies $n >> k$ where k is the number of local optima, the initial locations of the n fireflies should be distributed as uniformly as possible to ensure that a comprehensive search throughout the search domain occurs. As the FA proceeds, the fireflies should converge into all of the local optima, including the global ones. By comparing the best solutions among all these optima, the global optima can easily be determined. As noted above, the FA approaches the global optima whenever the number of fireflies $n \rightarrow \infty$ and the number of iterations t, is set so that $t >> 1$ [23]. In reality, the FA has a tendency to converge very quickly into both local and global optima [22] [23].

As can be observed in the pseudo code of the FA, the two parameters that most directly impact the solution running time of the FA are the values selected for n and t. Obviously, for practical applications, the desire is to be able to determine the best solution in the shortest period of time. This would correspond to setting n and t at the minimum possible values that produce the best solution(s). Using terminology from computational complexity, the search time for the FA is linear in t, but is a second order polynomial in n.

5 Conclusion

In this study, an empirical response surface approach was employed to provide the functional form of the osmotic dehydration responses for mushrooms. Using these estimates of the functional form, the resulting optimization model was formulated into a non-linear goal programming problem. The optimal solution to the goal programming problem was found using a computationally efficient, FA-directed procedure and the osmotic parameters determined were shown to be superior to those found in all previous instances. Since the running time complexity of the FA is linear in the number of iterations but polynomial in the number of fireflies, it would be more computationally practicable to run the FA using a relatively larger number of iterations in combination with a "reasonable" number of fireflies than vice versa. Since an FA can clearly be modified to solve a diverse spectrum of "real world" problems beyond

the context of fresh produce dehydration, the computational approach employed in this study can obviously be extended into numerous other "real world" settings. These extensions will be considered in future research.

REFERENCES

[1] Geohive World Crop Production www.geohive.com/charts/ag_crops.aspx 2014.

[2] Mehta, B.K., S.K. Jain, G.P. Sharma, V.D. Mugdal, R.C. Verma, A. Doshi and H.K. Jain, *Optimization Of Osmotic Drying Parameters For Button Mushroom (Agaricus Bisporus)*. Applied Mathematics, 2012. 3(10A): p. 1298-1305.

[3] Venturini, M.E., J.E. Reyes, C.S. Rivera, R. Oria and D. Blanco, *Microbiological Quality And Safety Of Fresh Cultivated And Wild Mushrooms Commercialized In Spain*. Food Microbiology, 2011. 28(8): p. 1492-1498.

[4] Rosa, M.D., and F. Giroux, *Osmotic Treatments And Problems Related To The Solution Management*. Journal of Food Engineering, 2001. 49(3): p. 223–236.

[5] Jain, S.K., and R.C. Verma, *Osmotic Dehydration: A New, Promising And Emerging Industry*. Beverage and Food World, 2003. 30(1): 3, p. 30–34.

[6] Rastogi, N.K., K.S.M.S. Raghavarao, K. Niranjan, and D. Knorr, *Recent Developments In Osmotic Dehydration: Method To Enhance Mass Transfer*. Food Science Technology, 2002. 13(1): p. 48–59.

[7] Hawkes, J., and J.M. Fink, *Osmotic Concentration Of Fruit Slices Prior To Dehydration*. Food Processing Preservation, 1978. 2(4): p. 265-267.

[8] Shukla, B.D., and S.P. Singh, *Osmo-Convective Drying Of Cauliflower, Mushroom And Green Pea*. Food Engineering, 2007. 80(2): p. 741-747.

[9] Tonon, R.V., A.F. Baroni, and M.D. Hubinges, *Osmotic Dehydration Of Tomato In Ternary Solutions: Influence Of Process Variables On Mass Transfer Kinetics And An Evaluation Of The Retention Of Arytenoids*. Food Engineering, 2007. 82(4): p. 509-517.

[10] Nieto, A., M.A. Castro, and A. Alzamora, *Kinetics Of Moisture Transfer During Air Drying Of Blanched And/Or Osmotically Dehydrated Mango*. Journal of Food Engineering, 2001. 50(2): p. 175–185.

[11] Jain, S.K., R.C. Verma, L.K. Murdia, H.K. Jain, and G.P. Sharma, *Optimization Of Process Parameters For Osmotic Dehydration Of Papaya Cubes*. Food Science and Technology, 2011. 48(2): p. 211-217.

[12] Kar, A., and D.K. Gupta, *Osmotic Dehydration Characteristics Of Button Mushrooms*. Journal of Food Science and Technology, 2001. 38(4): p. 352–357.

[13] Sodhi, N.S., N. Singh, and K. Komal, *Osmotic Dehydration Kinetics Of Carrots*. Journal of Food Science and Technology, 2006. 43(4): p. 374–376.

[14] Torreggiani, D., and G. Bertolo, *Osmotic Pretreatments In Fruit Processing: Chemical, Physical And Structural Effects*. Journal of Food Engineering, 2001. 49(30): p. 247–253.

[15] Imanirad, R., and J.S. Yeomans, *Fireflies In The Fruits And Vegetables: Combining The Firefly Algorithm With Goal Programming For Setting Optimal Osmotic Dehydration Parameters Of Produce*. in *Recent Advances In Swarm Intelligence And Evolutionary Computation*, X-S. Yang, Editor 2015. Springer: Heidelberg (Germany). p. 49-69.

[16] Yeomans, J.S., *Computing Optimal Food Frying Parameters Using The Firefly Algorithm*. Journal on Computing, 2014. 4(1): p. 40-44.

[17] Yeomans, J.S., *Establishing Optimal Dehydration Process Parameters For Papaya By Employing A Firefly Algorithm, Goal Programming Approach*. International Journal of Engineering Research and Applications, 2014. 4(9): p. 145-149.

[18] Yeomans, J.S., and X.S. Yang, *Determining Optimal Osmotic Drying Parameters Using The Firefly Algorithm*. International Conference on Applied Operational Research (ICAOR), Vancouver, Canada, July 29-31, 2014.

[19] Box, G.E., and D.W. Behnken, *Some New Three Level Designs For The Study Of Quantitative Three Variables*. Technometrics, 1960. 2(4): p. 455-475.

[20] Myers, R.H., and D.C. Montgomery, Response Surface Methodology: Process And Product Optimization Using Designed Experiments, 1995: John Wiley and Sons, New York (USA).

[21] Montgomery, D.C., Design, and Analysis of Experiments 4th Edition, 1997: John Wiley and Sons, New York (USA).

[22] Imanirad, R., X.S. Yang, and J.S. Yeomans, *Modelling-To-Generate-Alternatives Via The Firefly Algorithm*. Journal of Applied Operational Research, 2013. 5(1): p. 14-21.

[23] Yang, X.S., Nature-Inspired Metaheuristic Algorithms 2nd Edition 2010: Luniver Press, Frome(United Kingdom).

[24] Yeomans, J.S., and X.S. Yang, *Municipal Waste Management Optimization Using A Firefly Algorithm-Driven Simulation-Optimization Approach*. International Journal of Process Management and Benchmarking, 2014. 4(4): p. 363-375.

[25] Alam, M.S., A. Singh, and B.K. Sawhney, *Response Surface Optimization Of Osmotic Dehydration Process For Aonla Slices*. Food Science Technology, 2010. 47(1): p. 47-54.

[26] Mudhar, G.S., R.T. Toledo, J.D. Floros and J.J. Jen, *Optimization Of Carrot Dehydration Process Using Response Surface Methodology*. Journal of Food Science, 1989. 54(11): p. 714–719.

[27] Shi, L., C.H. Xue, Y. Zhao, Z.J. Li, X.Y. Wang and D.L. Luan, *Optimization Of Processing Parameters Of Horse Mackerel (Trachurus Japonicus) Dried In A Heat Pump Dehumidifier Using Response Surface Methodology*. Food Engineering, 2008. 87(1): p. 74-81.

[28] Uddin, M.B., P. Amsworth, and S. Ibanoglu, *Evaluation Of Mass Exchange During Osmotic Dehydration Of Carrots Using Response Surface Methodology*. Food Engineering, 2004. 65(4): p. 473-477.

[29] Ranganna, S., Handbook of analysis and quality control for fruits and vegetable products, 1986, Tata McGraw Hill Publishing, New Delhi (India).

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Volume 5 No. 6
ISSN 2054-7390

Disassembly Modeling of an of End-Of-Life (EOL) Mechanical Damper for Recycling

[1]Akintunde, Olalekan. Akintunde, [2]Simolowo, Oluwafunbi. Emmanuel
[1, 2] Department of Mechanical Engineering, University of Ibadan. Nigeria.
esimmar@yahoo.com; oe.simolowo@mail.ui.edu.ng

ABSTRACT

Today's rapidly developing technologies and product designs have enabled manufacturers to deliver new products to consumers at a dramatic rate. This has in turn resulted in shorter lifespan for products, because, more often than not, they are discarded even though they are still in excellent working conditions. An overflowing stream of used scrapped products has become an alarming problem for waste management, as it is quickly elevating the level of environmental detriments. Countries around the world have also observed an explosive growth in the waste stream that is filling up municipal landfills and clogging up incinerators. These problems, in addition to the limited material resources on Earth have made the recycling of End-Of-Life (EOL) products a rapidly expanding research area. However, before recycling of EOL products can be done, they need to be disassembled. Therefore, this paper focuses on the automation of disassembly of EOL car dampers that are found wasting away in many auto repair workshops. It discusses the application of the component-mating graph and newer CAD methods in the disassembly modeling of an EOL mechanical damper.

Keywords: Disassembly, Component-Mating Graph, Modularity Analysis, Sequencing

1 Introduction

In the recent years, the knowledge of conserving energy, material resources, and landfill (dump sites) capacity, and recycling regulations, has put pressure on many manufacturers to produce and dispose of products, in an environmentally friendly manner. It has also aroused consumer interest in "green products", and has promoted environmentally responsible use, consumption, and disposal of products. Many governments worldwide are also stiffening legislation, thereby requiring manufacturers to use as much recycled material as possible, and to play a key role in recycling products at their end of life (EOL) [1] – [6].

Throughout the world, many reclamation facilities have been established by product manufacturers, for study and disassembly of their products. Sony Corporation (a Japanese Electronics Manufacturer), has built the Sony Disassembly Evaluation Workshop, at Stuttgart, Germany, to assess the reuse and recycling qualities of their electronic products. The International Business Machines (IBM) Corporation, which is one of the world's largest manufacturers of computers, has also established the Reutilization Centre at Endicott, New York, to disassemble and recover reusable components from their personal and notebook computer products. Even though a significant amount of research has focused on the design of products

DOI: 10.14738/tmlai.56.2530
Publication Date: 1ST Oct, 2017
URL: http://dx.doi.org/10.14738/tmlai.56.2530

from an environmental and disassembly perspective [9] – [10], disassembly still plays an important role in material and product recovery. [11].

1.1 Objective of Study

The objective of this work is to apply the component-mating graph and modern CAD methods to a newly-analysed EOL mechanical device, a car damper that is commonly disposed. This entails the use of: Autodesk Inventor, Computer Aided Design (CAD) software; MATLAB; SimMechanics Link Utility software for linking the CAD software with MATLAB for disassembly modeling; graph representation of the car damper and modularity analysis.

2 Methodology

One of the modeling methods is the Component-Mating Graph, through which the Modularity Analysis of the Damper is performed, in this paper. This is also referred to as Component-Fastener Graph. It is an undirected graph which can be constructed using data from Computer Aided Design (CAD) software that was used to design the product. For this paper, Autodesk Inventor Professional 2013 software was used. In a Component Mating Graph [12] – [13], $G = (V, E)$, where vertices $V = \{v_1, v_2, \ldots, v_n\}$ represent components; and edges$_n$ denote geometrical relationships among components, where m is the number of edges. It is clear that the upper limit for $|V|$ is n. Figure 1 shows the CAD model of the Damper, with its four component parts. The labels (shown) in parenthesis are chosen arbitrarily to make the components identifiable during the disassembly modeling process. The components of the Damper model were mated/ constrained, and their motion was simulated within Autodesk Inventor. An XML file was generated from the CAD model within Autodesk Inventor, and this new file was exported to, and saved in MATLAB's current directory

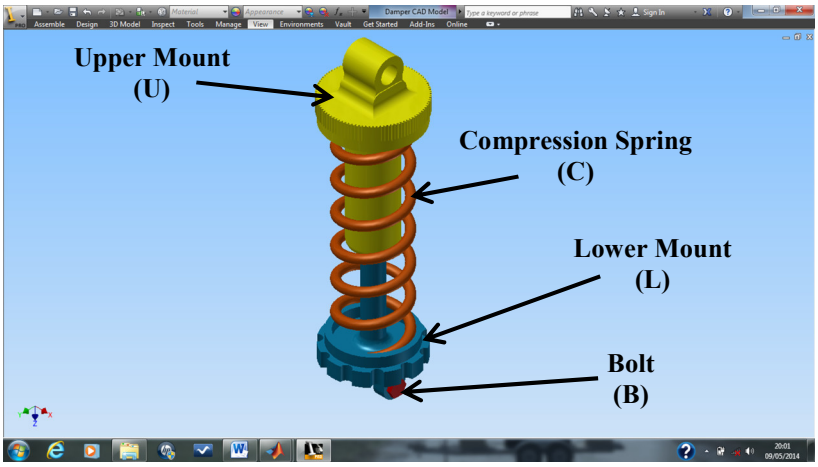


Figure 1: Damper CAD Model

At the command prompt in MATLAB, the **mech_import** command was executed. This brought up the **Import Physical Modelling XML** dialog box, and the previous XML file was selected. A SimMechanics model of the damper was then generated in MATLAB. This model showed the connection between the

parts of the imported 3D CAD model, as well as their various properties, such as; mass, principal moments of inertia, volume and surface area. The SimMechanics model is shown in Figure 2.

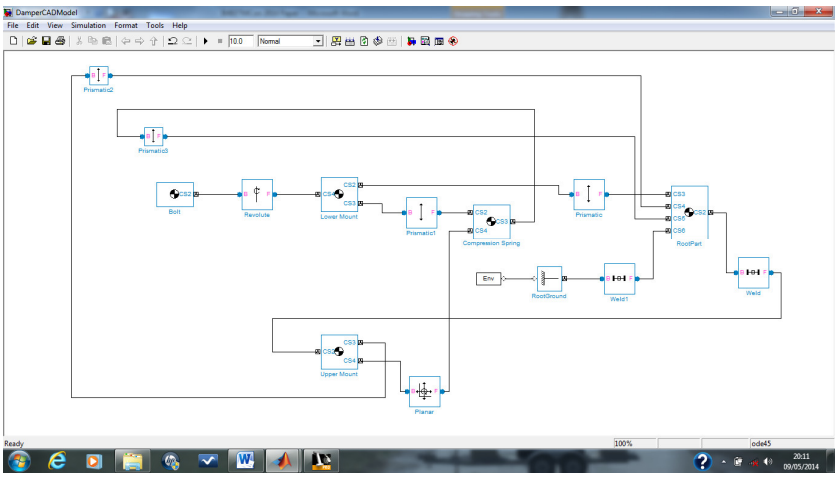


Figure 2: SimMechanics Model of the Damper

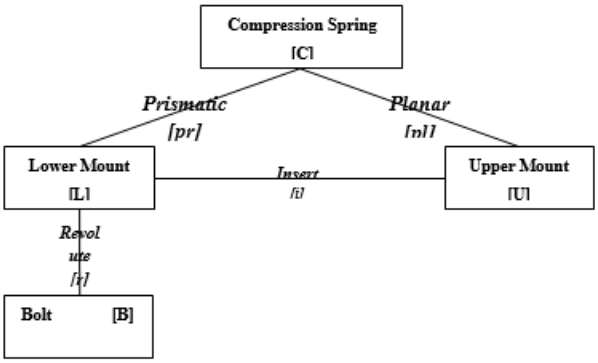


Figure 3: Component-Mating Graph of the Damper

3 Application of Component-Mating Graph to Damper

The Connection Graph (shown in Figure 3) was then constructed from the SimMechanics model by excluding the additional feature blocks created by the software The Connection Graph of Figure 3 can be written mathematically as: $G_C = (V,\ E)$,

Where: $V = \{U,\ C,\ L,\ B\}$, i.e. (set of vertices), $E = \{pr,\ pl,\ i,\ r\}$, i.e. (set of edges)

Number of vertices, or components $(n) = 4$. Let $E_C = [E_{ij}]$ be $G_C's$ adjacency matrix.

$$\therefore E_C = \begin{array}{c} \\ \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \text{ Were } E_{ij} = \begin{cases} 1, if\ component\ i\ is\ connected\ to\ component\ j \\ 0,\ otherwise \end{cases}$$

3.1 Modularity Analysis

To reduce the complexity of an EOL product, an assembly (module) can be decomposed into several subassemblies (sub-modules). The subassemblies can be further decomposed into simple subassemblies. Before describing the decomposition methodology, the concepts of cut-vertices, pendant vertices, and bi-connected graphs must be defined. Cut-Vertex (CV): a vertex whose removal disconnects the graph. In a module, cut- vertices refer to the connection component between two other components. Pendant Vertex: only one edge incident on the vertex is a pendant vertex. In a module, it represents a single component. Bi-connected graph: a connected graph with no cut-vertices. In a module, a bi-connected graph means a sub-module.

3.2 Subassembly

There are three types of geometric assembly methods: Type I: An assembly which has no main component (or CV). In this type of assembly method, all the components are assembled with others. Type II: An assembly which has a main component (or CV). In this type of assembly method, other components or subassemblies are directly assembled or indirectly assembled with the main component. Type III: A combination of Type I and Type II. From the Component Mating Graph of Figure 3, it is obvious that the Damper contains the Type II subassembly category. When considering the disassembly processes, the following disassembly rules are adopted: Rule 1: If a type I subassembly is found, the subassembly can be further disassembled as sub subassemblies, or single components. Rule 2: If a type II subassembly is found, the subassembly can only be further disassembled as single components. Rule 3: If a type III subassembly is found, the subassembly can be further disassembled into more subassemblies and/ or single components.

3.3 Depth First Search (dfs)

The dfs algorithm used in this paper was adopted from the algorithm shown in the book *of Algorithmic Theory and Practice* (Gilles and Paul, 1988). The search was done by following a path from the top to the bottom component of the Damper.

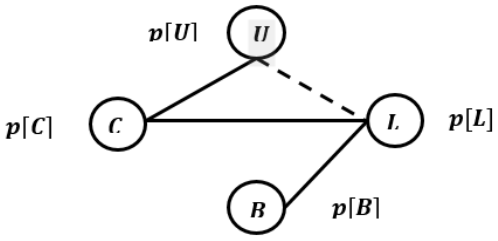


Figure 4: Depth First Tree (G_d) for the Component Mating Graph (G_C) of the Damper

The label: $prenum\ [v_i]$, which appears on each node is the algorithm for the dfs procedure, and is illustrated as follows:

$prenum\ [1]$	$dfs(U)$	Initial call
$prenum\ [2]$	$dfs(C)$	Recursive call
$prenum\ [3]$	$dfs(L)$	Recursive call

$prenum\ [4]$ $dfs(B)$ Recursive call

The broken line represents an edge that was present in graph G_C, but which was not present in graph G_d.

3.4 Cut Vertex Search

From the tree G_d of Figure 4, the $lowest[v_i]$ for each node is calculated in post order (i.e. from last node to first node of the dfs procedure). This is given by the relation:

$$lowest[v_i]\ or\ L[v_i] = min\left\{prenum[v_i],\ min\{prenum[w_i]\},\ min\{prenum[c_{ij}]\}\right\}.$$

Where: $prenum[v_i]$: the search order for each node v_i (from dfs procedure)

$prenum[w_i]$: if a node v_i has a back edge (shown with dashed line) to one or more node(s) (say v_k) higher up the tree in G_d, then $prenum[w_i] = min\{prenum[v_k]\}$ for the node(s) adjacent to only the back edge; otherwise, $prenum[w_i] = \infty$.

$prenum[c_{ij}]$: for each child j of node v_i in G_d, $prenum[c_{ij}] = lowest[v_j]$ for all the children. If node v_i has no child, then $prenum[c_{ij}] = \infty$.

The results are shown below:

$$lowest[B] = min\left\{prenum[B],\ min\{prenum[w_B]\},\ min\{prenum[c_{Bj}]\}\right\}$$
$$= min\{4, \infty, \infty\} = 4.$$
$$lowest[L] = min\left\{prenum[L],\ min\{prenum[w_L]\},\ min\{prenum[c_{Lj}]\}\right\}$$
$$= min\{3, 1, 4\} = 1.$$
$$lowest[C] = min\left\{prenum[C],\ min\{prenum[w_C]\},\ min\{prenum[c_{Cj}]\}\right\}$$
$$= min\{2, \infty, 3\} = 2.$$
$$lowest[U] = min\left\{prenum[U],\ min\{prenum[w_U]\},\ min\{prenum[c_{Uj}]\}\right\}$$
$$= min\{1, 3, 2\} = 1.$$

Figure 5 shows the comparison between $prenum[v_i]$ and $lowest[v_i]$ for all the nodes of G_d.

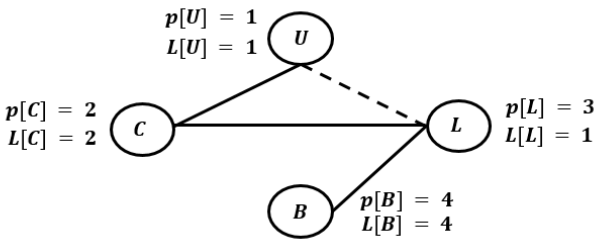


Figure 5: Comparison of $prenum[v_i]$ and $lowest[v_i]$ for Nodes of Graph G_d of the Damper

The cut vertex is determined from the following rules:

Rule 1: The root (first vertex) of graph G_d is a cut vertex of G_C if and only if it has more than one child. Since the root vertex $[U]$ has only one child $[C]$, it is not a cut vertex.

Rule 2: A vertex v_i other than the root of G_d is a cut vertex of G_C if and only if v_i has

a child j such that $lowest[v_j] \geq prenum[v_i]$. From Figure 4.8, the only vertex that satisfies this condition is $[L]$.

Therefore, the Cut Vertex (CV) of the graph G_d is vertex $[L]$.

3.5 Pendant Vertex and Sub-Graph Classification

To find the pendant vertex, it can be assumed that $CV's$ row and column vectors are zero in the adjacency matrix E_{ij}. If the total edge number of a vertex (except for the edge that is connected with CV) is 0, then that vertex is a pendant vertex based on the CV.

Recall:

$$\therefore E_C = \begin{array}{c} \\ \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

For any vertex i, if $EN_i = \sum_{j=1}^{n} E_{ij} = 0$, where $j \neq CV$, then i is a pendant vertex.

From the matrix E_C above, only vertex ID 4 satisfies this condition.

$$EN_4 = \sum_{j=1}^{4} E_{4j} = E_{41} + E_{42} + E_{43} + E_{44} = 0 + 0 + 0 + 0 \quad = 0.$$

Therefore, the Pendant (p) Vertex is vertex $[B]$. To find the sub-graphs in the graph G_C, it can be assumed the $CV's$ row and column vectors are zero in the adjacency matrix E_{ij}. This means that the CV and p vertices are completely removed from the graph G_C. To achieve this, the dfs procedure will be carried out on the new graph $G'_C = G_c - CV - p$, as shown in Figure 6. The algorithm is as shown:

$prenum\ [1]$	$dfs(U)$	initial call
$prenum\ [2]$	$dfs(C)$	recursive call

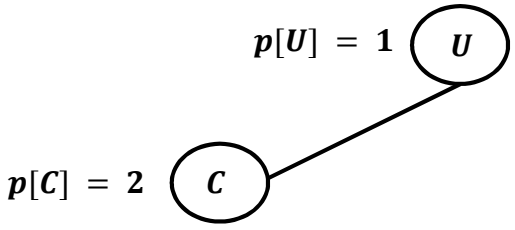


Figure 6: Depth First Tree $(G'_C = G_c - CV - p)$ of the Car Damper

From the dfs procedure of Figure 6, there is one sub-graph (subassembly) within the graph G'_C, and this is denoted by: $S = \{U, C\}$.

3.6 Decomposition and Modularity Analysis

Introducing CV and p back into the graph G'_C, and doing some re-arranging, produces the modularity graph shown in Figure 7.

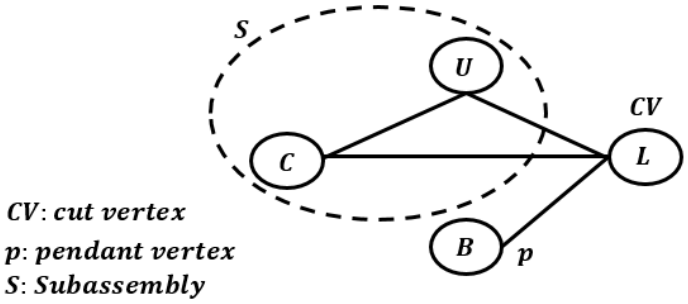


Figure 7: Modularity Analysis I of the Damper.

The next step is to determine whether $CV \subset S$. When examining the total number of edges for the subassembly (S) that are connected with CV, the CV will be grouped with the subassembly that has the highest number of edges connected with it. From Figure 7, subassembly S has two edges connected with the CV. Therefore $S = \{U,\ C,\ L\}$, as shown in Figure 8. Since the Damper module has now been decomposed into sub-modules, the disassembly process is analyzed by the DPMs. Disassembly precedence means a component i cannot be removed until component j is removed. The precedence relation is local to the parts concerned, and signifies the partial order of disassembly. In this paper, six disassembly directions are adopted (Eq. (1) – (6)): $\pm x$, $\pm y$, and $\pm z$ (from the CAD model of Figure 1).

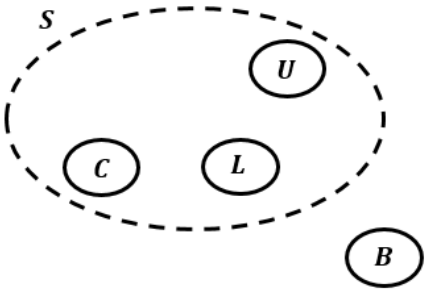


Figure 8: Modularity Analysis II of the Damper.

3.7 Disassembly Precedence Matrices (DPMs) Analysis

$$DP_{\pm d} = [DP_{ij}]\begin{cases}1, component\ j\ needs\ to\ be\ removed\ before\ component\ i, along\ \pm d\\ 0, otherwise\end{cases}$$

(1), (2)

$$DP_{+x} = \begin{array}{c} \\ \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad DP_{-x} = \begin{array}{c} \\ \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \quad (3), (4)$$

$$DP_{+y} = \begin{array}{c} \\ \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad DP_{-y} = \begin{array}{c} \\ \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad (5), (6)$$

$$DP_{+z} = \begin{array}{c} \\ \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad DP_{-z} = \begin{array}{c} \\ \mathbf{U} \\ \mathbf{C} \\ \mathbf{L} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{U} & \mathbf{C} & \mathbf{L} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

3.8 Merging of the DPM's

The six DPM's can be written as (Eq. (7) – (12)):

$$DP_{+x} = \begin{array}{c} \\ \mathbf{S} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \boldsymbol{D_{+x}} \\ 1 \\ 0 \end{array} \quad DP_{-x} = \begin{array}{c} \\ \mathbf{S} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} \boldsymbol{D_{-x}} \\ 0 \\ 1 \end{array} \quad (7), (8)$$

$$DP_{+y} = \begin{array}{c} \\ \mathbf{S} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} \boldsymbol{D_{+y}} \\ 1 \\ 1 \end{array} \quad DP_{-y} = \begin{array}{c} \\ \mathbf{S} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} \boldsymbol{D_{-y}} \\ 1 \\ 1 \end{array} \quad (9), (10)$$

$$DP_{+z} = \begin{array}{c} \\ \mathbf{S} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} \boldsymbol{D_{+z}} \\ 1 \\ 1 \end{array} \quad DP_{-z} = \begin{array}{c} \\ \mathbf{S} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{S} & \mathbf{B} \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} \boldsymbol{D_{-z}} \\ 1 \\ 1 \end{array} \quad (11), (12)$$

Therefore, the six DPM's can be represented as one DPM D_{ij} as shown in Eq. (13)

$$DP = \begin{array}{c} \\ \mathbf{B} \end{array} \begin{array}{c} \begin{array}{cccccc} \mathbf{S} & \boldsymbol{D_{+x}} & \boldsymbol{D_{+y}} & \boldsymbol{D_{+z}} & \boldsymbol{D_{-x}} & \boldsymbol{D_{-y}} & \boldsymbol{D_{-z}} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array} \quad (13)$$

From the matrix "DP", the subassembly or component can be disassembled if $DP_{ij} = 0$. Therefore, subassembly $S = \{U,\ C,\ L\}$ can only be disassembled from the $-x$ direction, and component $\{B\}$ can only be disassembled from the $+x$ direction.

4 Result of Disassembly Analyses

The disassembly tree and sequence are obtained as a final outcome of the disassembly analyses of the car damper as shown in Figure 9. From the disassembly tree, in level 0, the Damper assembly is represented as a parent vertex $\{U,\ C,\ L,\ B\}$ which is in the level 0 (L_0). In level 1 (L_1), the assembly is decomposed into one subassembly $S = \{U,\ C,\ L\}$, and component $[B]$. The Modularity Analysis and Disassembly Precedence Matrix (DPM) test are processed recursively until all the components are disassembled from the whole assembly. In level 2, the subassembly S is decomposed into components $\{U\},\ \{C\},$ and $\{L\}$. Thus, the disassembly process is complete.

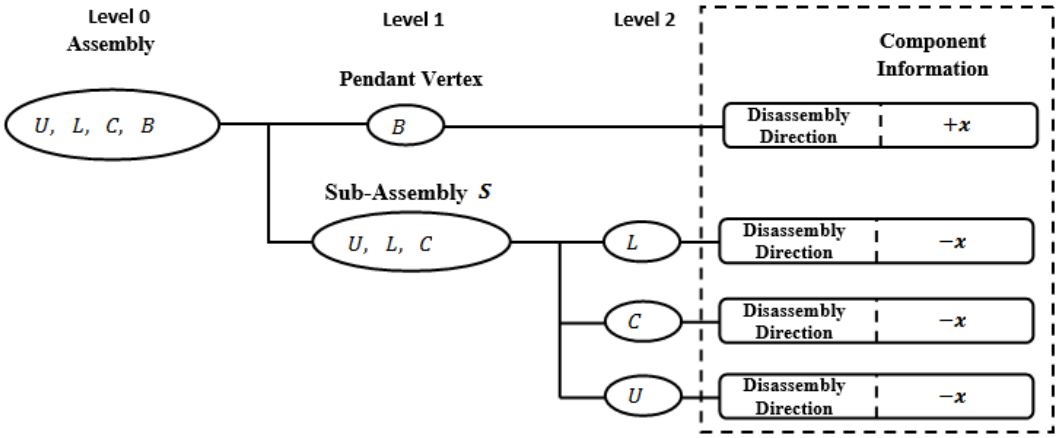


Figure 9: Disassembly Tree Representation of the Car Damper

5 Conclusion

A graph-based product design representation is presented for generating disassembly sequences of an Automobile Damper assembly. A Modularity Analysis was also used to generate feasible disassembly sequences. The process involved the use of the Component-Fastener Graph and Disassembly Precedence Matrix (DPM) that were developed for the Damper. The Component-Fastener Graph represents the hierarchy of the product structure, while the DPM represents the local and partial order of disassembly. The final result of the analysis was the disassembly tree, which shows complete disassembly sequences of the Damper. This sequence of disassembly is to incorporated in the automation of the recycling process of the damper.

REFERENCES

[1]. Chynoweth E. and Rotman D, *Catalyst: Cover story*, June 1993. chemical week,

[2]. Williams, J. A. S., *A Review of Research towards Computer Integrated Demanufacturing for Materials Recovery.* International Journal of Computer Integrated Manufacturing, 2007. 20(8): p. 773-780.

Simolowo, O. E.; Mousavi, A.; Adjapong, P. O., *A Computer-Based Product Classification and Component Detection for Demanufacturing Processes*. International Journal of Computer Intergrated Manufacturing Tailor and Francis, 2011. 24(10): p. 900 – 914.

[3]. Simolowo O. E. and Mousavi A., *Developing Imaging Software for Automated Recycling Processes (Phase 1).* Journal of Advanced Materials Research, 2012. 628: p 298 -303.

[4]. Simolowo O. E. and Onovugbe E. C., *Automation of Generation of Models for Disassembly Process Planning for Recycling* Book Of Proceeding Conference of Mechanical Engineering ICME_2013 at Imperial College London, 3, p. 1964 – 1968.

[5]. Onovughe, E. C. I., *Disassembly Modelling Process for Recycling of EOL Products*". Unpublished. Project, University of Ibadan, Ibadan, Nigeria, 2012 p. 9-31.

[6]. Li, W.; Zhang, H. P.; Wang, B.; and Awoniyi, S. A., *Design for Disassembly Analysis for Environmentally Conscious.* Design and Manufacturing. Manufacturing Science and Engineering, ASME, 1995; 2-2, 3(2): p. 969-976.

[7]. Capelli et al. *Design for disassembly: a methodology for identifying the optimal disassembly sequence.* Journal of Engineering Design 2007, 18(6): p. 563–575.

[8]. Gu, P. and Yan, X. (1996). *CAD-Directed Automatic Assembly Sequence Planning*. Int. J. PROD. RES., 1996, 33(11): p.3069-3100.

[9]. Mazouz, A. K.; Souilah, A.; and Talbi, M. *Design of an Expert System for Generating Optimal Assembly Sequence*. Computer-Aided Engineering Journal, December, 1991: p. 225-259.

[10]. Tang, Y.; Zhou, M. C.; Zussman, E.; and Caudill, R. (2002). *Disassembly Modeling, Planning, and Application.* Journal of Manufacturing Systems. 2002, 21(3): p. 200-217.

[11]. Gilles Brassard and Paul Bratley. *Algorithmic Theory and Practice.* 1988. Prentice hall publishers, Michigan

[12]. Johnson Baugh, R., *Discrete Mathematics Textbook.* 2005. .6th Edition

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Volume 5 No. 6
ISSN 2054-7390

Computational Intelligence for Congestion Control and Quality of Service Improvement in Wireless Sensor Networks

[1*]Mukhdeep Singh Manshahia, [2]Mayank Dave, [3]S.B. Singh
[1,3]Department of Mathematics, Punjabi University, Patiala, Punjab, India- 147002
[2]Department of Computer Engineering, NIT, Kurukshetra, Haryana, India- 136119
mukhdeep@gmail.com

ABSTRACT

Congestion and quality of service are widely researched topics in Wireless Sensor Networks in recent years. Many researchers proposed and compared the merits and demerits of various algorithms with the existing algorithms. The major challenge lies in developing an algorithm which optimizes the various performance parameters like packet drop ratio, residual energy and throughput of the network. Focus of the present work is to reduce congestion and improve quality of service by applying various metaheuristic or computational intelligence techniques which can optimize performance parameters. An objective function is formulated on the basis of factors like residual energy, throughput, distance between nodes and the number of retransmissions and its value is optimized by using various nature inspired computational intelligence techniques and their results are compared. Simulation results have shown that water wave algorithm outperforms all the other algorithms on the basis of packet drop ratio and throughput of wireless sensor network.

Keywords: Computational Intelligence, Congestion Control, Wireless Sensor Networks

1 Introduction

Congestion control in WSNs (Wireless Sensor Networks) means to improve the performance when demand for the finite transmission capacity exceeds the supply. There are mainly two causes for congestion in WSNs. First, packet arrival rate exceeds the packet service rate. Second is link level performance degradation due to contention, interference and bit-error rate. Increase in loss rate and starvation of most of the network due to traffic from nodes one hope away from sink are the basic symptoms of congestion in WSNs.

Congestion control is a core concept over the network for each and every type of media traffic, which has an active space of the analysis within the last decade. It can enhance the audio visual traffic of digital convergence [1]. Demand of the user for network applications is increasing boundlessly. Multiple users for network applications is endlessly growing therefore leading to congestion [2]. The major problem in congestion network is packet loss which gives transmission errors. TCP's congestion control mechanism reacts to packet loss to drop the amount of unacknowledged information segments allowed within the network [3]. All network applications do not used the TCP protocol. Therefore, the results of non- TCP applications are failed to have an abundant impact of most traffic within the network that uses TCP-based

DOI: 10.14738/tmlai.56.3689
Publication Date: 1ST Oct, 2017
URL: http://dx.doi.org/10.14738/tmlai.56.3689

protocols [4]. As the video and audio streaming applications like video players and web audio, video conferencing applications gives arise to non-TCP traffic. All the TCP-flows scale back their information rates to interrupt the congestion, wherever non-TCP flows maintains to transfer the actual rate [5]. It is extremely unfair that the condition can result in the starvation of TCP-traffic or congestion collapse. Congestion collapse can be described a state of the network when accessible data can be discarded the results of congestion before reaching the destination [6].

For this reason, it can be desirable to draw a fixed congestion management mechanism for the non-TCP traffic, which is compatible for the rate-adaptation mechanism for different kind of traffic. These mechanisms can be used to create the non-TCP application which is also TCP-friendly, and so result in distribution of information measure [7].

1.1 General Mechanisms for Congestion Control

Congestion should be controlled with efficiency. There are two basic approaches to control the congestion in WSNs. First approach of capacity provisioning means how to make the links wide enough to cope with traffic demand. The second approach is load control. WSNs nodes have very limited power due to hardware constraints. Packet losses and retransmissions resulting from congestion cost precious energy and shorten the lifetime of sensor nodes. This problem motivates the need for congestion control mechanisms in WSN. Congestion management protocol potency may depend on proportion it can do the subsequent objectives [8]:

i. Energy-efficiency requires to improved system life. Therefore, congestion management protocols have been compelled to avoid or cut back packet loss to buffer overflow, and stay lower to the management overhead that consumes less energy;
ii. Its conjointly necessary to support the ancient QoS metrics like packet delay, packet ratio, and throughput.
iii. Most of prevailing work guarantees straightforward fairness in each sensing node obtains an equivalent out turn to the sink. In fact, sensing nodes may be either outfitted with completely different sensing nodes or geographically deployed in several places and so they will have different importance or priority and want realize different out turn. Thus weighted fairness is needed.

In order to control the congestion, Network resource management approach is used to increase the network resource to mitigate congestion. In wireless network, power management and multiple radio interfaces will be used to increase the information measure and congestion avoidance. Virtual sinks in radio interfaces can be used. One primary low-power speck radio with the smaller information measure and another long-rage radio with the larger information measure [9]. When congestion occurs, long-rage radio can be used as cut off to mitigate the congestion. With this method, it is vital to ensure the precise network resource adjustment to avoid congestion. Other approach tries to control and manage congestion through adjusting the traffic rate at supply nodes or intermediates nodes [10]. This approach can be useful to save several network resources and economical precise adjustment of the network resource becomes tough. Most existing congestion management protocols belong to present sort. To control the management behaviour, two ways are used end-to-end and hop-by hop. The end-to-end management will execute precise rate adjustment at every node and change will look at intermediate

nodes. In distinction, the hop-by-hop congestion management has quicker response. However, it is sometimes tough to regulate packet forwarding rate at the intermediate nodes as a result of packet forwarding rate relies on MAC protocol and can be variable [11].

In Wireless Sensor Networks, each packet can possibly include helpful information, which may be utilised through packet-based computation, and utilised to reinforce the congestion management [12]. The packet-based computation may well be sensible for the Wireless Sensor Network considering:

1) Wireless Sensing Network typically has little packet forwarding rate and so unit time to forward one packet may well long enough for sensing node to perform the bound computation even supposing its computation capability is limited [13].
2) Though Wireless Sensing Network at most time are used with the restricted energy and lacks recharging approach. The packet-based computation continues to the most accepted for sensing node since typically acknowledges the computation consumes less energy than communication. Moreover packet-based computation may offer helpful information to the scale back or avoid useless communication and successively compensate the energy it consumes [14].

1.2 Focus of the Paper

The major focus area in the work is to reduce congestion and improve quality of service by applying various metaheuristic or machine learning algorithms which can optimize performance parameters. These algorithms then compared with each other on parameters like network lifetime, normalized system throughput and packet drop ratio. These algorithms work iteratively and try to optimize the result after each iteration. For detecting the congestion in the network queue length, window size, channel occupancy etc must be monitored periodically.

1.3 Paper outline

The whole work is divided into various sections which describes various aspects and phases of research. Section 2 provides the literature survey which is considered for the research. Section 3 describes the methodology used for the work which includes the derivation of objective function and the implementation of various proposed algorithms. The implementation of improved bat algorithm, firefly, water wave algorithm, PSO and ACO are also presented in this section. In Section 4, results of all the above implemented algorithms along with CODA are discussed. Conclusion and findings are given in Section 5. Finally, the limitation of the work, Suggestions and Future and acknowledgments are discussed in section 6, 7 and 8 respectively.

2 Literature Review

Chandrashekar et al. [15] proposed a transport protocol to control congestion. A fuzzy logic based congestion estimation and a congestion mitigation technique which decreases frame quality at an acceptable level is suggested. Congestion factor is calculated using fuzzy logic on the basis of frame size and buffer size. Frame rate based congestion controller has been used to maintain the rate of flow.

Cheng et al. [16] proposed a traffic regulation based approach to control congestion in wireless sensor networks. They have proposed to control congestion by forwarding the traffic regulating factors to children nodes. Traffic regulating factors are calculated from packet round trip delay and minimum event detection degree. Generation rate of source nodes are adjusted based on traffic regulations. Simulation

results have shown that network throughput has increased with reduced packet loss rate and stabilize node queue length.

Meshram et al. [17] proposed an agent based upstream congestion control scheme for wireless sensor networks. Priority index and congestion degree is used to analyze the traffic rate. Upstream traffic of network is controlled by a new multi-agent system. Latency and throughput parameters are investigated during simulation.

Aghaei et al. [18] proposed an ant colony based routing technique for Wireless Sensor Networks. They have proposed a shortest path finding techniques based on ant colony intelligence and these intelligent paths avoids the probable collision with help of congestion control mechanism. They have mapped the shortest path with neighborhood information and a data ant can guess collision probability with help of this mapping. Simulation results have shown that this protocol performs better in finding the shortest path.

Saleem et al. [19] gave a cross layer design based self-optimized routing protocol for wireless sensor networks by using ACO. Optimal path finding mechanism from source to destination is proposed by using link quality, energy level and velocity as parameters. Signal strength, remaining power and timestamp metrics are used for cross layer communication. The proposed algorithm is capable of avoiding loops and deadlocks. Simulation results has shown that proposed protocol provides better delivery ratio over WSN.

Raha et al. [20] proposed a genetic algorithm based load balancing protocol for congestion control in wireless sensor networks. They have represented the fitness function as a function of the representative genes of chromosomes. A balanced distribution of traffic among alternate paths from source to destination has been proposed. Protocol targets the selection of reliable and trusted routes more frequently than unreliable routes. Simulation results have shown improvement in data delivery ration and network lifetime.

Verma et al. [21] gave an optimal path finding approach from source to destination with different source and sink node mobility scenario based on genetic algorithm with crossover and mutation. Connection value and localization region has been used to find an optimal path every time before sending data packet which reduces chances of congestion in network. Sensor nodes in localization region are supposed to be localized. Simulation results have indicated that genetic algorithm provides better results in case of complex and large networks.

Naveena et al. [22] gave a cross layer dynamic adaptation mechanism for wireless ad hoc networks. A joint congestion control scheme with scheduling algorithm is proposed for dynamic networks by changing scheduling scheme with adaptation model. They have generalized the channel access management and routing process with management of traffic, connection maintenance and distributed scheduling for concurrent transmission. Simulation results have proved that the proposed protocol is highly stable and robust for unicast data and it improves the packet relay in all mobility situations.

3 Proposed Methodology

Nature inspired computational intelligence techniques are implemented in the work to control congestion in the Wireless Sensor Networks. The clustering operation is performed using firefly algorithm [23]. A fitness function is derived for clustering which is based on the values of residual energy and the distance

between the nodes. Each node in the cluster shares its information like number of packets lost, its residual energy, queue size etc to every other node in the cluster. Number of retransmissions of every node and the distance value between any two nodes in the cluster is calculated. This data is used by the proposed approaches to route the packets by calculating the optimum route.

An objective function based on various parameters like residual energy, packet loss ratio, throughput, distance has been proposed in the present work to control congestion in wireless sensor networks. The Improved Bat Algorithm which is based on the echolocation of bats, Firefly algorithm that relies on the attractiveness issue of the firefly, water-wave algorithm which works on the propagation, refraction and breaking of the waves, Particle swarm optimisation (PSO), and Ant Colony Optimization (ACO) are applied on the objective function to find the optimum solution. NS-2.35 based simulation results of implementation of all five algorithms are presented and compared with the Collision Detection and Avoidance algorithm (CODA). The analysis of the simulation results have shown that the computational intelligence techniques perform better on many parameters.

Figure 3.1 shows the flow diagram of proposed methodology. The proposed methodology is for all the five algorithms i.e. Improved Bat Algorithm, Firefly Algorithm, Water wave Algorithm, PSO and ACO. The result of the global best will be calculated after every iteration and compared with the previous global best solution.

3.1 Derivation of Fitness Function

The problem of congestion control in WSN is a very wide area of research. From the past few years many researchers use various algorithms to solve the same problem. Among the many reasons of congestion, Buffer overflow, Channel contention and packet collision are considered as the major focus areas. A fitness function is considered based on various parameters like throughput of the network, packet lost rate and residual energy. These parameters are:

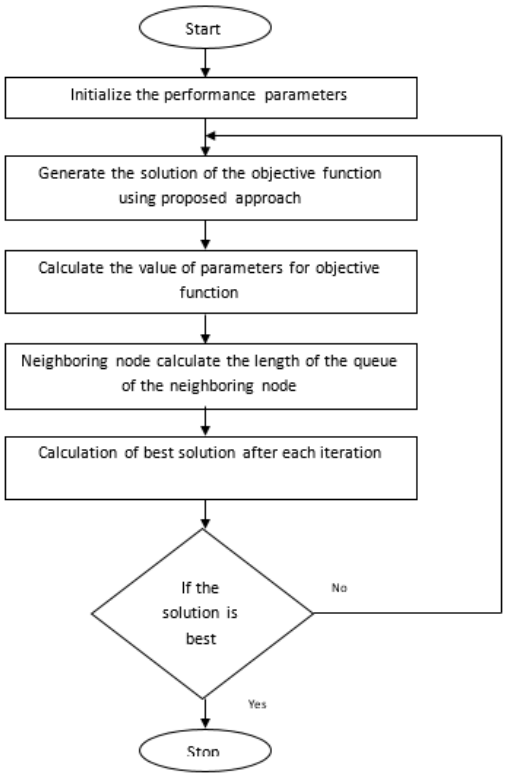


Figure 1: Flow Diagram of Proposed Methodology

The fitness function considered in the research work is a combination of the above written parameters along with the numerical weight given to each parameter. The numerical values will be given according to the contribution of each parameter in the optimum result of the algorithm. To improve the performance of the network and to reduce the congestion we need to optimize the fitness function.

$$\boldsymbol{Fitness\ Function}\left(F_j\right) = \sum_{i=1}^{n}\left(w_1 * \tau_i + w_2 * \left(1 - P_{L_i}\right) + w_3 * E_i + w_4 * d_{i,j} \right) \quad (1)$$

Where, i is the iteration which ranges from 1 to N (total number of nodes),

w_1, w_2, w_3 and w_4 are the weights supplied to the algorithm,

τ is the throughput of the network,

P_L is the normalized Packet lost rate in the network

$d_{j,i}$ is the distance between node i and j and

E is the residual energy of each node in the network.

3.2 Routing Technique

Reactive routing technique is used in the proposed methodology [25]. In reactive routing technique the route is determined in the real time. RREQ message is generated when a node wants to transfer the information or data which simply defines as the route request message. In reactive routing single hop or multi hop route is considered on the basis of RREQ message. In these types of protocol there is a provision

of route recovery also and many routing schemes consider the following three route recovery mechanisms:

RREQ - Source node initiates the route request message to the destination node. These requests generally have a time to live (TTL) parameter which is initially set as a predefined value and can be increased when there is a loss of packets in the network.

RREP- It is unidirectional message targeted towards the source node from the destination node. Route Reply message is used to inform the source node about the route to the destination node.

RERR- RERR message notifies the other nodes for the loss of the link. Nodes in the network monitor the link status and update the next hop information in the network.

3.3 Performance Metric

There are many factors responsible for the congestion in WSN. Congestion can be detected in WSN by tracking various parameters which gives a clear insight view of the congestion in WSN [26]. These factors are:

Channel Occupancy: It is defined as the number of active packets in the network which will occupy the network or channel resources. It is an important parameter as it describes the channel usage by the nodes transferring the packets in the network.

Reporting Rate: Reporting rate is defined as the number of packets reaching the destination over a specified period of time. It depends on the sub tree value of each node.

Queue Length: The length of the queue of each node also plays an important role in congestion detection in the network. It is the parameter associated with each node and the number of packets that a node can store is termed as its queue length. A node can store only a fixed number of packets and if the queue is full then packets started to overflow and thus lead to extra packets in the network which can cause congestion.

Packet Loss and Window Size: Window size is associated with the network and is defined as the number of packets that can flow in the network on a particular moment of time. This parameter is also related to the channel capacity of the network and only a fixed number of packets can be handled by the network at a particular time. If the number of packets increases in the network at that time then it lead to the loss of the packets and these packets also cause congestion in the network. So a network is designed to handle the packets in an optimum way by distributing the load to all the nodes in the network.

The congestion is detected by monitoring the queue length and the window size of each node in the proposed work.

3.4 Proposed Congestion Control Approaches

The objective of the congestion control is to increase the throughput of the network. So the problem of congestion becomes a multi objective constraint optimization problem. Nature-inspired algorithms Improved Bat Algorithm, Firefly Algorithm and Water wave Algorithm, PSO, ACO are implemented to optimize the objective function. These algorithms are based on iterations and an optimal route is calculated based on iterative analysis.

Improved Bat Algorithm: The algorithm is based on the concept of echolocation of bats [27, 28, 29]. For obstacle detection and avoidance, bats use sonar echoes which reflect from obstacle and are transformed to frequency. They use time delay between the emission and reflection and use that delay for navigation. This can be implemented in the proposed work by using simple rules proposed in the algorithm. Packets will go randomly in any direction towards their destination node and use the value of packet lost rate and residual energy of the target node to decide which node they will choose to go further. This selection is based on the fitness function value.

Firefly Algorithm: Firefly is an insect that mostly produces short and rhythmic flashes that produced by a process of bioluminescence [24, 30, 31]. The function of the flashing light is to attract partners (communication) or attract potential prey and as a protective warning toward the predator. Thus, the intensity of light is the factor of the other fireflies to move toward the other firefly. This algorithm can be implemented in this work by taking advantage of the attractiveness factor of the fireflies. Packets will follow the path through those nodes which have high residual energy value. The node can broadcast the value of the fitness function along with the residual energy to their neighbouring nodes in the network. Packets will follow the path which has the most optimized set of fitness values.

Water Wave Algorithm: Water Wave algorithm is based on the shallow water wave models for solving optimization problems [32,33]. When a wave travels from deep water to shallow water, its wave height increases and its wavelength decreases and vice versa. In Water Wave Optimization, the solution space and the fitness of a point is inversely proportion to its seabed depth. It means the shorter the distance to the still water level, the higher will be the fitness function $f(x)$. This algorithm is implemented in this work by considering the fitness function and performing all the three operations i.e. propagation, refraction and breaking.

Particle Swarm Optimization (PSO): Particle Swarm Optimization (PSO) is a metaheuristic algorithm which computes the mathematical solution depicting the particles in the search space [34, 35, 36]. PSO calculates the solution of the moving particles in the search space in terms of velocity and position. It iteratively computes the best solution of the problem. Swarm and data packets will follow search space path computed by the algorithm.

Ant Colony Optimization (ACO): Ant Colony Optimization (ACO) is a method for finding the solution in terms of optimal routes based on the behavior of ants searching for food [37, 38]. Firstly ants wander randomly and when they find the food they walks back to the colony leaving the traces called the pheromones for other ants to get the path to the food. This model is used by packets to select path for transmission.

3.5 Congestion Detection and Avoidance (CODA)

We have compared our approach with Collision Detection and Avoidance (CODA).Collision Detection and Avoidance (CODA) is a congestion control mechanism which works on the energy efficiency parameter [39, 40]:

- Closed-loop multisource regulation
- Open loop hop-by-hop backpressure
- Receiver based congestion detection

MAC layer is used in the performance of the algorithm and management of data in the network. CSMA is used for detection and avoidance of congestion.

3.6 Simulation Environment

For creating the simulation environment, a 1000*1000 grid is considered in ns- 2.35. 50 nodes are placed in the network grid. Wireless parameters like antenna type, channel type and propagation model are defined for each node. MAC layer is used for the extraction of energy from each node with standard IEEE 802.11 and the radio model has been adopted [41,42,43,44].

Table 1. Simulation Parameters

Parameter	Value
No. of Nodes	50
X dimension	1000
Y dimension	1000
Grid Area	1000*1000
Mac Protocol	IEEE 802.11
Propagation Model	Two-Ray Ground Model
Transmission Range	200m (approx.)
Antenna type	Omni-Antenna
Channel type	Wireless Channel
Routing Protocol	AODV
Simulation time	60
Interface Queue type	Prequeue

4 Results and Discussions

A comparative analysis is performed between Improved Bat Algorithm, Firefly Algorithm, Water Wave Algorithm, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and CODA. The factors of comparison are the throughput of the network, network lifetime and the packet loss ratio. The proposed approaches which are implemented on an objective function to resolve the problem of congestion follow different approaches to solve the problem. While CODA (Congestion detection and Avoidance) is also implemented on similar environment and compared to our proposed approaches.

Figure 7.7 shows the graph of Normalized system throughput. Throughput of a network is defined as the ratio of packet size and the time required transferring the packet from source to destination. System throughput as shown in the graph is best for Water wave Algorithm while performance of firefly algorithm is also better than CODA, Improved Bat algorithm, PSO and ACO. Improved Bat algorithm, PSO and ACO performs almost same in this case. The performance of CODA is worst because there is a decrease in value for CODA at around 30 seconds of simulation

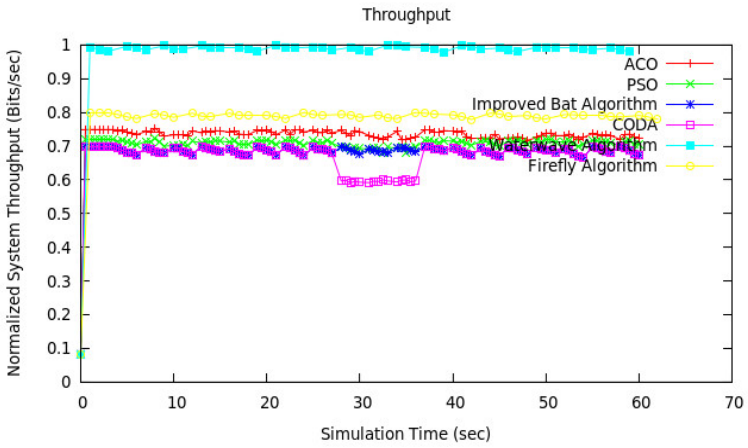


Figure 2: Graph between Simulation Time and Normalized System Throughput

Figure 7.8 shows the graph of Network lifetime plotted against the simulation time. In this graph the network lifetime in case of Improved Bat Algorithm is better as compared to the other algorithms. CODA has the worst performance as the residual energy is less and exhausted very quickly. While the performance of Firefly and Water wave algorithm, PSO is almost same and better than ACO

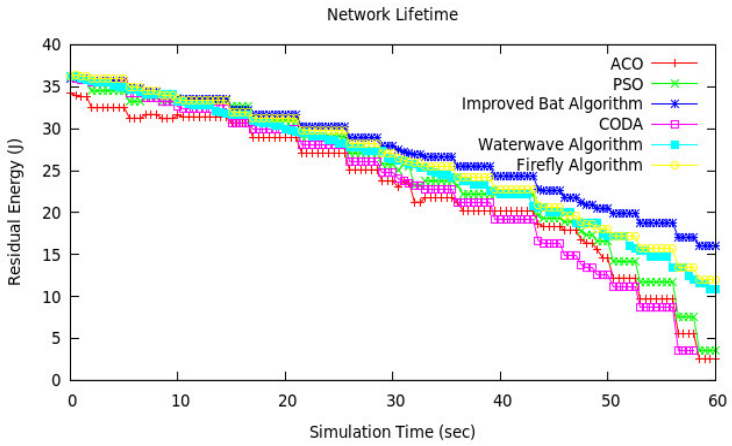


Figure 3: Graph between Simulation Time and Network Lifetime

Figure 7.9 shows the graph of packet loss (drop) ratio. In this graph packet loss ratio of all four algorithms is considered and compared with each other. , Water wave algorithm performs better than other algorithms. Packet loss rate of CODA and PSO is almost equal while it is on higher side in case of Improved Bat algorithm, Firefly algorithm and ACO.

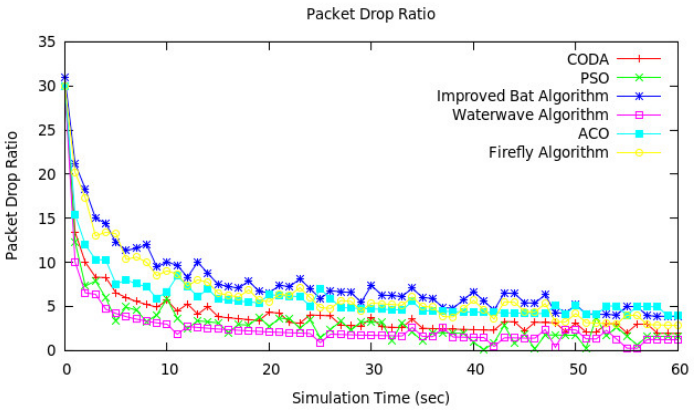


Figure 4: Graph between Simulation Time and Packet Drop Ratio

Table 2. Comparison of Various Techniques

Algorithm	Throughput	Network Lifetime	Packet Drop Ratio
BAT	M	H	H
Firefly	H	M	M
Water Wave	VH	M	L
PSO	M	M	M
CODA	L	L	L
ACO	M	L	M

The values for input linguistic variables are Low (L), Medium (M), High (H) and Very High (VH).

5 Conclusion and Findings

To solve the problem of congestion various machine learning algorithms have been proposed in recent past. In this work we have implemented five of the recognized machine learning algorithms. The results of various performance parameters are compared with each other and their approximation for the best value is carried out. Few important findings are drawn from this work which is well suited with their theoretical explanation. Number of packets stored by the sensor node directly depends on the hop count. It is also evident from the graphs of all the proposed approaches that the number of packets in the queue decreases with the increase in the number of hops. It means that while transferring the packets from source to destination the number of hops encountered in the route depicts the congestion in the network.

All the proposed algorithms are also compared on the basis of three other performance parameters which are:

Normalized System Throughput: Throughput value is normalized to 1 and Water wave algorithm outperforms all the other algorithms and maintains its value close to 1. In case of firefly throughput is better as compared to the PSO, ACO, CODA and Improved Bat Algorithm. The performance of CODA is worst due fluctuation at 30 sec.

Network Lifetime: Network lifetime of nodes in the network is an important parameter of comparison as it decides the active period of network. It directly depends on the residual energy of all the nodes in the network. The network lifetime in case of Improved Bat Algorithm is better as compared to the other

algorithms. In case of CODA the energy of all the nodes exhausted very rapidly and the network which works on CODA algorithm becomes inactive in a short period of time. While in case of Water Wave, Firefly and PSO the network lifetime is better as compared ACO.

Packet Drop Ratio: While transferring the packets from source to destination, Packets can be lost due collisions and network congestion. Water wave algorithm performs better than other algorithms on packet loss ratio. Packet loss rate of CODA and PSO is almost equal while it is on higher side in case of Improved Bat algorithm, Firefly algorithm and ACO.

6 Limitations

Major limitation of this work is that all the algorithms used in optimizing the fitness function are not showing the better results in every performance parameter considered. Bat algorithm performs better in case network lifetime but performs badly in case of packet drop ratio. The performance of Firefly algorithm is average. Water wave algorithm shows best performance in case of normalized system throughput and packet drop ratio parameter while in case of remaining parameter other algorithms perform slightly better.

7 Suggestions and Future Scope

Congestion in the Wireless Sensor Network is dealt by implementing various machine learning algorithms and their results are compared. In future the problem may be dealt by implementing various other algorithms which can perform best in almost all the considered performance parameters. Also performance of algorithms present in this work may be tested on various other parameters by varying the size of the network, as the performance of the given parameters can be changed by increasing the number of nodes in the network.

ACKNOWLEDGEMENTS

Authors are grateful to Punjabi University Patiala and NIT, Kurukshetra for providing adequate library and internet facility

REFERENCES

[1] J. Long, H. Zhang, O. W.W. Yang, *A Queueing performance comparison of Extended proportional controller in TCP Traffic Control*, Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering, IEEE, vol.2, Montréal , 4-7 May 2003. p. 973 – 976.

[2] Zhai. H, Chen. X, Fang. Y, *Rate-Based Transport control for Mobile Ad Hoc Networks*, Proceedings of 2005 IEEE Wireless Communications and Networking Conference, WCNC 2005. 4:p. 2264 - 2269

[3] G. Anastasi, E. Anallatti, M. Conti, A. Passarella, *TPA: a Transpart Protocol for AD hoc Networks*, Proceedings of 10th IEEE symposium on computers and communications, IEEE, 27-30 June 2005. p. 51-56.

[4] M. Elshakankiri, Y. Damroury, *A quality of service protocol for Mobile Ad hoc Networks*, Proceedings of IEEE 46th Midwest Symposium on Circuits and Systems, IEEE, Vol. 1, 27-30 Dec. 2003. p. 490-495.

[5] K. Chen, Y. Xue, K. Nahrstedt, *On setting TCP's congestion window limit in Mobile Ad Hoc Networks*, Proceedings of IEEE International Conference on Communications, Vol. 2 , IEEE, 11-15 May 2003. p. 1080 - 1084.

[6] H. Xu. H, X. Wang, C.K. Toh, *Comparative analysis of scheduling algorithms in Ad Hoc Mobile Networking*, Proceedings of Sixth international conference on parallel and distributed computing, Applications and Technologies, IEEE, Dec. 2005. p.639 - 643.

[7] A. Legout, E.W. Biersack, *Revisiting the fair queueing paradigm for End to End congestion control*, IEEE Network, 2002. 16(5): p. 38 - 46.

[8] D. Kim, C. K. Toh, H. J. Jeong, *An early Retransmission Technique to improve TCP performance for Mobile Ad Hoc Networks*, Proceedings of 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2004. 4: p. 2695 – 2699.

[9] A. Kumar, L. Jacold, A. L. Ananda, *SCTP Vs TCP: Performance comparison in MANETs*, Proceedings of 29th Annual IEEE International Conference on Local Computer Networks, IEEE, 16-18 Nov. 2004. p. 431 – 432.

[10] D. Kim, H. Bae, C.K. Toh*, Improving TCP-vegas performance over MANET Routing protocols*, IEEE Transactions on Vehicular Technology, 2007. 56(1): p. 372 – 377

[11] A.A. Abouzeid, M. Azizoglu , *Comprehensive performance analysis of a TCP session over a Wireless Fading Link with Queueing*, IEEE Transactions on Wireless Communication, 2003. 2(2): p. 344 - 356

[12] V.S. Mansouri, B. Afsari, H. Shahmansouri, *A simple Transport protocol for wireless sensor Networks*, Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference, IEEE, 5-8 Dec. 2005. p. 127 - 131.

[13] S. Hashmi, H. T. Mouftah, N. D. Georganas, *A New Transport layer Sensor network protocol*, Proceedings of IEEE Canadian Conference on electrical and Computer Engineering, Ottawa, May 2006. p. 1116 - 1119.

[14] I. Stoica, S. Shenker, H. Zhang, *Core-Stateless Fair Queueing : Achieving approximately fair bandwidth allocations in high speed Networks*, IEEE/ACM Transactions on Networking , 2003. 11(1): p. 33 – 46

[15] G. Chandrashekar, S. S. Manvi, *Fuzzy Logic based Congestion control in Wireless Multimedia Sensor Networks*, International Journal of Latest Trends in Engineering and Technology, 2014. 4(2):p.216-221

[16] L. Cheng, Z. Yong, X. Wei-xin, Traffic Regulation based Congestion Control Algorithm in Sensor Networks, Journal of Information Hiding and Multimedia Signal Processing, 2014. 5(2): p. 187-198

[17] S. Meshram, R. V. Bobate, S. Chaturvedi, *Congestion Control by Multiagent in Wireless Sensor Network*, International Journal of Software and Web Sciences, 2013. 5(1): p. 37-41

[18] R.G. Aghaei, A.M. Rahman, M.A. Rahman, W. Gueaieb, *Ant Colony-Based Many-to-One Sensory Data Routing in Wireless Sensor Networks*, Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, IEEE, Doha , March 31- April 4 2008. p. 1005 - 1010.

[19] K. Saleem, N. Fisal, M.A. Baharudin, A.A. Ahmed, S. Hafizah, S. Kamilah, *Ant Colony inspired Self-Optimized Routing Protocol based on Cross Layer Architecture for Wireless Sensor Networks*, WSEAS transactions on communications, 2010. 9(10): p.669-678

[20] A. Raha, M.K. Naskar, A. Paul, A. Chakraborty, A. Karmakar, *A Genetic Algorithm Inspired Load Balancing Protocol for Congestion Control in Wireless Sensor Networks using Trust Based Routing Framework*, I.J. Computer Network and Information Security, 2013. 9: p. 9-20

[21] A. Verma, N. Mittal, *Congestion Controlled WSN using Genetic Algorithm with different Source and Sink Mobility Scenarios*, International Journal of Computer Applications, 2014. 101(13):p. 8-15

[22] G. Naveena, H. H. Babu, *Dynamic Congestion Control with Multihop Scheduling Algorithm For Mobile Ad-Hoc Network*, International Journal of Innovative Research in Computer and Communication Engineering, 2014. 2(1): p. 2011-2015

[23] Manshahia, M.S., Dave, M. and Singh, S.B., *Firefly algorithm based clustering technique for Wireless Sensor Networks*, in: Proceedings of International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 23-25 March 2016.

[24] Manshahia, M.S., Dave, M. and Singh, S.B., *Congestion Control in Wireless Sensor Networks Based on Bioluminescent Firefly Behavior*, Wireless Sensor Network , 2015. 7: p. 149-156

[25] R. Ding, L. Yang, *A Reactive Geographic Routing Protocol for Wireless Sensor Networks*, Proceeding of 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Brisbane, 7-10 December 2010. p. 31-36.

[26] L. Alazzawi, A. Elkateeb, *Performance Evaluation of the WSN Routing Protocols Scalability*, Journal of Computer Systems, Networks, and Communications, Vol. 2008 , 2008.

[27] X..S. Yang, A. H. Gandomi, *Bat algorithm: a novel approach for global engineering optimization*, Engineering Computations, 2012. 29(5): p. 464-483

[28] A. H. Gandomi , X.S. Yang, *Chaotic Bat Algorithm*, Journal of Computational Science, 2014: 5(2): p. 224–232

[29] Manshahia, M.S., Dave, M. and Singh, S.B., *Bio Inspired Congestion Control Mechanism for Wireless Sensor Networks*, in: Proceedings. of 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, India , December, 2015.

[30] X..S. Yang, Xingshi He, *Firefly Algorithm: Recent Advances and Applications*, Int. J. of Swarm Intelligence, 2013. 1(1): p. 36-50

[31] X..S. Yang, *Firefly algorithms for multimodal optimization*, Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computer Sciences, 2009. 5792: p. 169–178

[32] Y.J. Zheng, Water wave optimization: A New Nature-inspired Metaheuristic, Computers & Operations Research, 2015. 55: p. 1–11

[33] Manshahia, M.S., *Water Wave Optimization Algorithm based Congestion Control and Quality of Service Improvement in Wireless Sensor Networks*, Transactions on Networks and Communications, 2017. 5(4): p .31-39

[34] P. Antoniou, A. Pitsillides, T. Blackwell, A. Engelbrecht and L. Michael, *Congestion Control in Wireless Sensor Networks based on Bird Flocking Behavior*, Computer Networks, 2013. 57(5): p. 1167–1191

[35] R. C. Eberhart, J. A Kennedy, *New optimizer using particle swarm theory*, Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, 1995. p. 39-43

[36] J. Kennedy, R.C. Eberhart, *Particle swarm optimization*, Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995. p. 1942-1948

[37] Marco Dorigo, Thomas Stützle, *The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances*, Handbook of Metaheuristics, Volume 57 of the series International Series in Operations Research & Management Science, 2003. p. 250-285, (ISBN: 978-0-306-48056-0)

[38] Marco Dorigo, Gianni Di Caro, *The ant colony optimization meta-heuristic, New ideas in optimization*, McGraw-Hill Ltd., UK Maidenhead, UK, England, 1999. p.11-32 (ISBN: 0-07-709506-5)

[39] C.Y. Wan, S.B. Eisenman and A.T. Campbell, *CODA: Congestion Detection and Avoidance in Sensor Networks*, Proceedings of the 1st international conference on Embedded networked sensor systems, sensys 03, Los Angels, USA, Nov 5-7, 2003. p. 266-279

[40] Mukhdeep Singh, S.B. Singh, Mayank Dave, *Congestion Control in Computer Networks*, Ultrascientist: International Journal of Physical Sciences, 2008. 20(2m)

[41] F. Paganini, Z. Wang, J.C. Doyle,S.H. Low, *Congestion control for high performance, stability and fairness in general networks*, IEEE/ACM Transactions on Networking Los Angeles, 2005. 13: p. 43-56

[42] Manshahia, M.S., *Wireless Sensor Networks: A Survey*, International Journal of Scientific & Engineering Research, 2016. 7(4): p. 710-716

[43] Manshahia, M.S., *A Firefly Based Energy Efficient Routing in Wireless Sensor Networks*, African Journal of Computing & ICT, 2015. 8(4): p. 27-32

[44] Manshahia, M.S., Dave, M. and Singh, S.B, *Improved Bat Algorithm Based Energy Efficient Congestion Control Scheme for Wireless Sensor Networks*, Wireless Sensor Network, 2016. 8: p. 229-241

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Volume 5 No. 6
ISSN 2054-7390

Twitter Sports: Real Time Detection of Key Events from Sports Tweets

[1]Jeyakumar Kannan, [2]AR. Mohamed Shanavas, [3]Sridhar Swaminathan
[1,2]Department of Computer Science, Jamal Mohamed College, Tiruchirappalli, India;
[3]Department of Computer Science and Engineering, Bennett University, Greater Noida, India;
meetjey@gmail.com; arms3375@gmail.com; sridhar.swaminathan@bennett.edu

ABSTRACT

Twitter users play a role of human sensors and update information about real-life events by posting their tweets about them. Event detection in Twitter is the process of detecting an event which is an occurrence causing change in the volume of tweets that discuss the associated topic at a specific time and a location by Twitter users. Twitter has been extensively used to detect major social and physical events such as *earthquakes*, *celebrity deaths*, *presidential elections*, *traffic jam* and others. Real time event detection in Twitter is detecting real-life events from live tweets instantly as soon as the event has occurred. Real time event detection from Cricket sports using Twitter media is an interesting, yet a complex problem. Because, event detection algorithm needs live tweets streamed at real-time about the game and should detect events such as *boundary* and *sixer*, at near real-time within few seconds from their occurrences. In this paper, a novel real-time event detection approach is proposed for the Cricket sports domain. The proposed approach first computes the post rate of an adaptive window, which is the ratio between the volumes of tweets in the second half window and the volume of tweets in the first half. An event has occurred if the post rate is above the pre-defined threshold, otherwise the algorithm selects the next big window in an adaptive manner. The predefined threshold helps to filter out the small spikes in the streaming tweets volume. Once an event is detected in a time window along the tweet stream, the event represented inside the window is recognized using the event lexicon representing different events of a cricket game. The proposed real-time event detection algorithm is extensively evaluated on 2017 IPL T20 Cricket sports dataset using ROC and AUC evaluation measures. The experimental results on the performance of the proposed approach show that the adaptive sliding window detects sports events with over 80% true positives and around 15% false positive rates.

Keywords: Social media; Microblogs; Twitter; Event detection; Sports events; Adaptive sliding windows.

1 Introduction

Modern day internet, web and mobile technologies have now enabled vast majority of people around the world to communicate with each other through social media services such as Facebook, Twitter etc. Online social media services have changed the way communication happened between people, groups, and communities [1]. These social media has become a platform for many of its users to express their ideas, opinions and share information to the rest of the world. Microblogging is one such form of social media broadcasting medium where users share small digital content such as short texts, links, images, or

DOI: 10.14738/tmlai.56.3729
Publication Date: 2nd Oct, 2017
URL: http://dx.doi.org/10.14738/tmlai.56.3729

videos [2]. Microblogging has become famous and highly used by numerous people, organizations and researchers from different fields around the entire world, despite it is a totally new medium of communication when comparing to the traditional social media. It allows users to share information and respond to different opinions quickly. Despite the constraints on size of the information being shared in the microblogging, it has gained wide spread usage due to its features such as easier portability, unrestricted content, quickness and ease of usage in communication.

Among the popular social media services in microblogging, Twitter is one of the most widely used and fast-growing microblogging social networking service which has more than 500 million users around the world [3]. Twitter allows its users to share a short text called *Tweet*, which is no longer than 140 characters, by using different communication services such as smartphones, web interfaces and social media apps. Twitter differs from other social networks by being a micro-blogging service that limits the size of messages. This feature allows Twitter users to publish short messages, in a faster and summarized way, makes it the preferred tool for the quick dissemination of information over the web. The content of a tweet highly varies based on individual user's interests and behaviors [4]. These tweets contain wide range of information such as advice, opinions, moods, concerns, facts, rumors, world news, and general information, report of important events [5]. In the context of online social networking, social media users can be regarded as sensors reporting important information.

People, community and organizations can be well informed of live happenings around the world from the dynamic source of information in Tweets. Twitter users are also interested in receiving different tips, opinions, live updates on news from the other Twitter users [4]. Corporations use Twitter to make announcements of products, services, events, and news media companies use Twitter to publish near real-time information about breaking news. Several organizations have started utilizing Twitter as a platform for advertisements, product, and service recommendations. They also started to exploit reports from sentiment analysis to build and maintain reputations of their products by responding to the complaints from customers and dynamically improving their decisions [6]. Twitter has become a quick communication medium for obtaining and sharing of viral news [7], election results prediction [8], and crime prediction [9]. Invaluable information can be gained by the continuous monitoring and analysis of rich user-generated content. For example, it is easy to understand the top trending topics in sports such as cricket by analyzing the most frequent terms from tweets. Figure 1 to 4 depict the top 25 terms and bigrams of two of the games we crawled during IPL T20 season 2017. From these figures, it is evident that Twitter users generally discuss their favorite players and teams throughout the game, besides Cricket events. This information would not have been obtained from any other traditional media outlets.

Events can be generally defined as real-world occurrences that unfold over space and time [10]. Research on social media analytics has shown that important real-life events can be detected from the information provided by the human sensors where the credibility of the information being shared by the users were always high most of the time. Automatic event detection from microblogging social media becomes necessary, due to a large volume of data, redundancy in information reporting events and presence of noises or false information about the events. Recent research on event detection has shown that Twitter can provide insights into a detection of major social and physical events such as earthquakes, celebrity deaths, and presidential elections [11]. Event detection from Twitter is considered difficult since the detection algorithm should cope with challenging factors such as limited text length, structural and grammatical errors, rumors and misinformation. In Twitter event detection, the underlying assumption

that some related words would show an increase in the usage when an event is happening is not a viable method. In comparison to traditional event detection from news wire, Twitter stream include a much higher volume of data flooded by high amounts of meaningless messages.

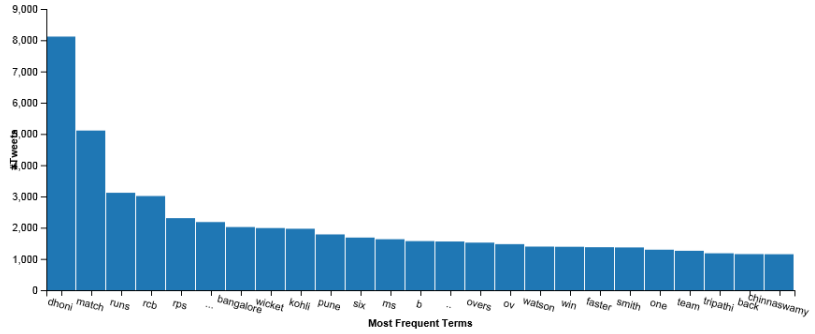


Figure.1. Most frequent terms for RCBvRPS on 16 April 2017

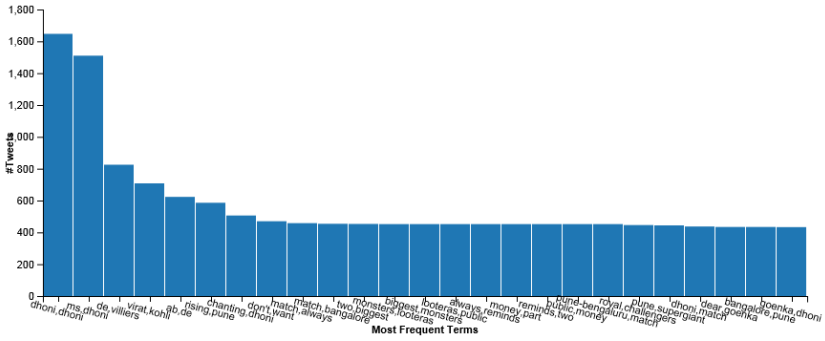


Figure 2. Most frequent bigrams for RCBvRPS on 16 April 2017

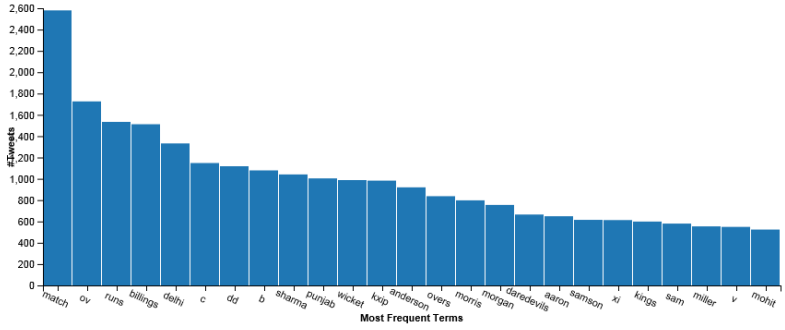


Figure 3. Most frequent terms for DDvKXIP on 15 Apr 2017

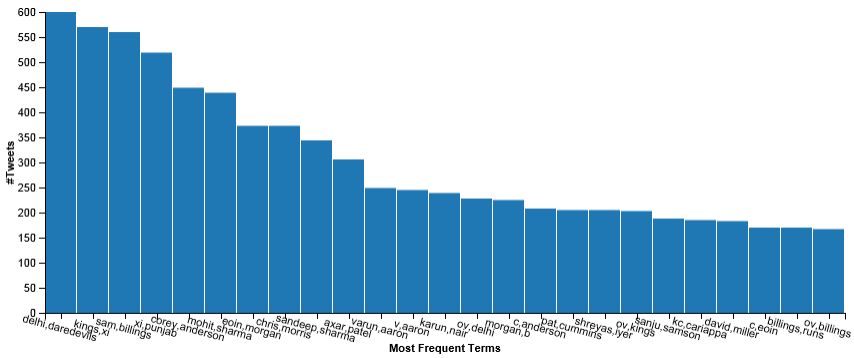


Figure 4. Most frequent bigrams for DDvKXIP on 15 Apr 2017

In this digital era, news about an event instantly reaches the other side of the world within seconds as an event happens. Millions of people around the world have started using Twitter for reporting significant events in real-time. Real-time detection and recognition of events from Twitter is yet another challenging study in recent times. There are lot of advantages in detecting events from real-life using real-time event detection in Twitter in situations such as catastrophic events, important deaths, political events, elections and campaigns. Real-time event detection is considered highly challenging due to various factors such as gathering and processing data at real-time for societal and business applications.

Although event detection has been a well-studied problem for over a decade, limited number of research work have been carried out in sports domain. Even in domain of sports, most of research work focused on NFL soccer games and detected events from offline datasets. Recently, few researchers have considered detecting key events from NFL games at real-time. Nevertheless, there are no studies that investigated *Cricket* sports as a domain for the detection of events from live tweets at real-time.

To address this demand, this paper proposes a novel event detection approach, *TwitterSports*, based on two complementary ideas namely event lexicon and adaptive sliding windows. The event lexicon is a dictionary of terms representing each event and populated with cricket terminologies gathered from the popular website for sports. The event detection and recognition algorithm recognizes events by examining the post rate of tweets when a game is ongoing. It helps to reduce the computational load significantly because detection can be achieved without analyzing the content of tweets, utilizing the event lexicon. The major contributions of this paper are summarized as follows:

1. Unlike previous approaches which used offline datasets and training data for event detection, we present a novel approach which detects events at real-time based on event lexicon and adaptive sliding windows. Our event detection algorithm recognizes all key events on the fly and does not require training data. This is particularly important as some applications such as *terrorism* and *presidential elections*, do not have any training data. Also, it does not require any natural language preprocessing steps as the event lexicon can handle all possible name variations for a possible key event and greatly reduces the computation overhead for the real-time detector.
2. For sports event detection and reporting, many studies have been made in NFL soccer sports domain. Similar to soccer sports, cricket has been one of the popular sports and attract a lot of viewers during the game. Since all viewers post tweets about the happenings of a game, a widely agreed event lexicon for Cricket sports will help researchers for detecting events. Therefore, this

paper proposes an event lexicon that has not been reported before in the literature. The event lexicon represents 37 key events for Cricket sports.

Our work is unique in a way that we study Twitter as the real-time output of the human sensors to infer the physical world. We investigate how Twitter users, as human sensors, report key events of sports games such as cricket at real-time. To the best of our knowledge, ours is a first step towards proposing an event lexicon for cricket terminologies and our algorithm is the first of its kind that detects and recognizes cricket events at real-time utilizing live tweets, when a game is ongoing.

The rest of the paper is organized as follows. In section 2, we survey the related work on Twitter event detection approaches. In section 3, we introduce Twitter APIs and our data collection method. We describe the adaptive window based real-time event detection method in section 4 and examine the performance of the proposed approach in section 5. Finally, we conclude the paper and describe our future work in section 6.

2 Related Work

A wide number of research contributions can be found in the recent literature on Twitter event detection. Generally Twitter event detection approaches can be classified into different types such as environment, social and sports events detection based on the domain in which events are detected. Based on the class of solutions, they can be categorized into term-interestingness based, incremental clustering based, topic modeling based and frequency based approaches [12]. Among them frequency based event detection is considered as a simple yet effective class of approach where no training data is needed for the detection process. For more detailed study on Twitter event detection, the readers are recommended to refer to some recent surveys [2, 12]. However in this section, recent research belong to different domains and solution classes will be discussed.

Some of the earlier work involved detection of social and physical events from Twitter. Earthquakes detection using Twitter was explored by Sakaki et al [13] and Qu et al [14]. Environment related events such as grassfire and floods in microblogs were studied by Vieweg et al [15]. In TwitterStand [16], news topics were discovered from the Twitter data where news stories among them were found using clustering the related tweets. A major drawback of the aforementioned approaches is that they could detect the event only several minutes after the actual event happened.

In a similar work by Hannon et al [17], tweet post rate is exploited to generate World Cup game highlights in a form of video. However, the approach generates the highlights in offline mode where the specific game events were not recognized. Some previous work in the sports domain have not applied event detection at real-time. Chakrabarti and Punera [18] trained Hidden Markov Models on training data of tweets of previous events to describe the events with the assumption that a game event is recognized already.

Detection and recognition of events in the sports domains has been vastly studied by the Computer Vision community earlier. Visual features has been analyzed to summarize the important events in the videos of soccer games [19]. Due to a low correlation between the visual features and the sports events, detection of events using visual features is not a highly reliable strategy. Some work have exploited the textual information associated with the videos. Rui et al [20] exploited speech detection to detect important events in baseball game videos. In another work by Zhang and Chang [21], closed captions were utilized

to detect and summarize key events in the baseball videos. Petridis et al [22] and Xu et al [23] utilized textual features such as MPEG-7 and webcast text to detect events in sports. The practicality of utilizing the aforementioned textual features highly depend upon their availability. It can be seen that these textual features do not always come in handy to support detection of events at real-time. However, Twitter is one such source which is both instant and easily available for processing as long as the game events are witnessed by huge audience.

System developed by Mathioudakis and Koudas [24] detects events based on high-rate and unusual appearances of keywords. Another event identification system named EDCoW by Weng and Lee [25] identifies events by exploiting wavelet analysis on frequencies of words by measuring new features of words. Further, words with low signal auto-correlations are filtered out, where modularity-based graph partitioning is applied for clustering the remaining words. To improve the scalability, each new document is compared against the previous document using locality-sensitive hashing [26, 27]. Twitter stream is first clustered where a classifier is trained on Twitter data, annotated using temporal, social, topical and Twitter-specific distinguishable features, in an event detection system proposed by Becker et al. [28].

Emerging topics in Twitter are detected [29] by locating strongly connected components of a directed graph containing emerging topic terms which are identified by comparing frequency of current terms with previous terms for a given period of time. An event visualization and summarization system, TwitInfo, developed by Marcus et al. [30] detects events by analyzing temporal peaks. The event detection results are presented to the users using a timeline-based visualization display where the temporal peaks are highlighted in the timeline. In an event detection system by Valkanas and Gunopoulos [31], users are clustered based on their geographical locations and the emotional states of the group of users are monitored where events are detected if there is a sudden change in a group's emotional state.

Controversial events involving celebrities are identified by analyzing the public discussions in the Twitter in response to those events by Popescu and Pennacchiotti [32]. Factor graph model is used to analyze individual Twitter messages and to cluster them to detect concert events [33]. In his work, clusters are automatically formed according to the type of events and a canonical value is generated for each property of event. Geo-social event detection system developed by Lee and Sumiya [34] identifies local festivals by modeling and monitoring different behaviors of the crowd in Twitter. Their approach detects events by analyzing the geographical regularity found from the usual behavior patterns by using geo-tags.

Sakaki et al [35] exploited tweets to detect specific types of events such as earthquakes and typhoons. They formulated event detection as a classification problem and trained an SVM on a manually labeled Twitter data set that comprises of a sample of positive and negative tweets. By combining simple rules and strategies for query building, Becker et al [36] successfully augmented information about planned events in Twitter data where events are identified by applying precise query strategies that are derived from description of events. High-quality, useful and relevant Tweets related to an event are extracted using centrality-based approach by Becker et al. [37].

Generative language modeling approach [38] based on microblog's quality indicators and query expansion is used to retrieve microblog messages. Weerkamp and de Rijke [39] proposed quality indicators such as emoticons, length of tweet post, angry expressions, capitalization of words, and hyperlinks along with other characteristics such as recency of message, number of shares of messages and the number of

followers of the user. N-grams based event modelling approach called ETree [40] groups large volume of tweets into relevant information blocks using content analysis approaches.

In summary, some research work have investigated the network properties of Twitter such as *geo-locations*, some user properties such as *influence* and *emotion*, while others used Twitter users as human sensors for detecting physical and social events such as *earth quakes*, *festivals* and *presidential elections*. However, all these research work performed their processing offline. Notably, few research work have been carried out in the domain of sports, but these studies focused on NFL soccer games. No other previous approaches have considered a Cricket domain and solved the problem of real-time event detection for Cricket sports.

In this paper, we explore how good human sensors are for the real-time detection of events in *Cricket* sports. We demonstrate the feasibility of using Twitter for real-time event detection for Cricket sports which has frequent and rapid key events[1] (aka, *key moments*) like *boundary* and *sixer*. Most importantly, our Twitter-based approach can be readily extended to recognize social and physical events beyond Cricket sports as long as these events are witnessed by a large number of Twitter users. Therefore, the insights gained from this study will help other novel applications, such as *reporting traffic jam* and *festivals*, to use human sensors for the event detection at real-time.

3 Twitter API for *TwitterSports*

Twitter has become an ideal platform for people to publish spontaneously, as it limits its length to just 140 characters. As a result, it has the shortest delay in delivering user comments to citizens, compared to other social media platforms such as *blogs*. During international Cricket games, Twitter receives a huge volume of tweets from cricket fans and audience of the live game, tweeting about game moments that they find exciting or notable. *TwitterSports* leverages this activity, associating particular streams of tweets with game moments or sub-events (e.g. Boundary, Sixer, Catch) to perform robust event detection and event recognition in real-time.

Twitter supports three types of APIs namely *REST API, Search API* and *Streaming API*. The Representational State Transfer (REST) API allows application developers to access tweets stored in the main database that contains all tweets. The *Search API* will search only those tweets that were posted only in the past 7 days and return 100 tweets for a given query. The *Streaming API* returns tweets in real-time based on the filter predicates such as *follow* a user, *track* a keyword and a *location*.

The *Streaming API* is the most suitable type for our event detection task from live tweet streams. The advantages are as follows. It returns up to date tweets; there are no rate limits; we can filter tweets based on keywords. For example, we have used the keyword, *RCBvRPS* to stream all tweets at real-time. The keyword *RCBvRPS* denotes an IPL T20 game between the teams *Royal Challengers Bangalore* and *Rising Pune Supergiant.* Similarly, we have used another keyword, *DDvKXIP* to crawl all tweets of the game between the teams *Delhi Daredevil* (DD) and *Kings XI Punjab* (KXIP) at real-time. Our *TwitterSports* runs

[1] We use the terms *key moment* and *key event* interchangeably

continuously collecting tweets without any break during the entire game time, detects events from tweets at real-time and also archives all gathered tweets in JSON format for later offline analysis.

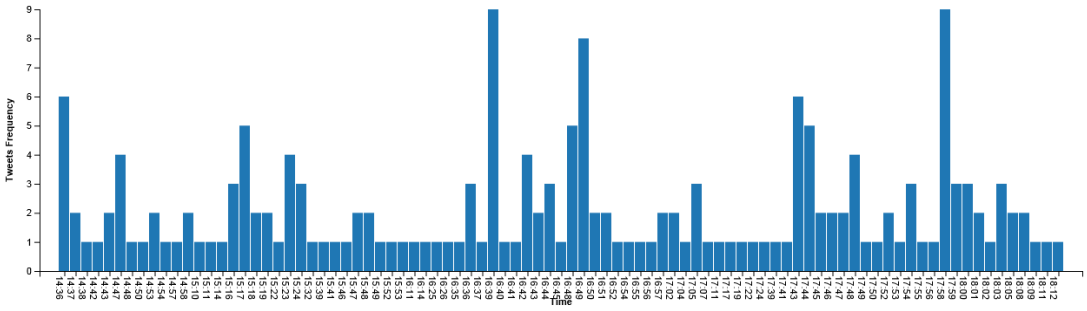


Figure 5. Tweet frequency of *Boundary* in RCBvRPS

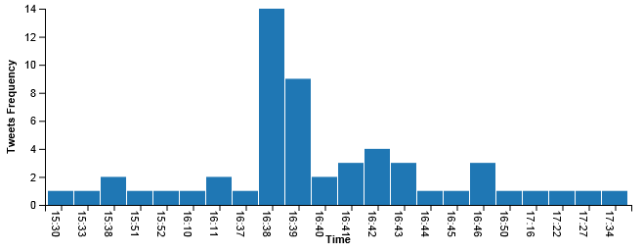


Figure 6. Tweet frequency of *Wide* in RCBvRPS

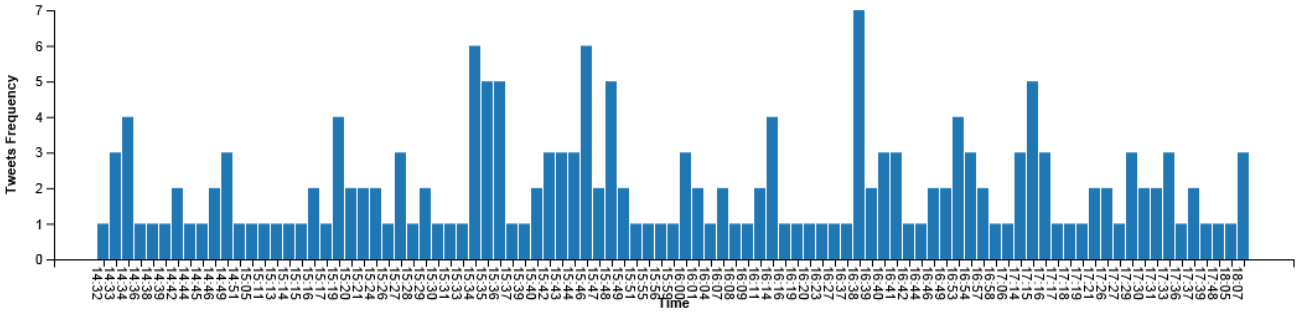


Figure 7. Tweet frequency of *Boundary* in DDvKXIP

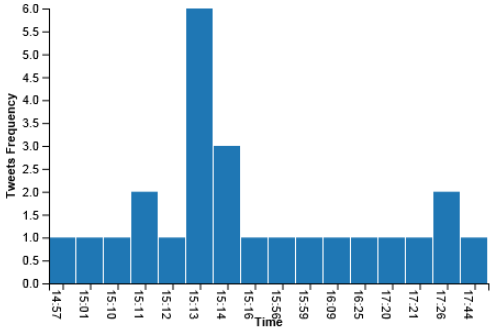


Figure 8. Tweet frequency of *Wide* in DDvKXIP

As a sports enthusiast, one can easily understand the interesting aspects of a game, by analyzing the collected tweets offline as well. We can also plot the timing of key events (aka, moments) such as *boundary, catch, wide* and so on by plotting the frequency of a term such as *catch* against time. Figure 5

to 8 depict the frequency of tweets about the events, *Boundary* and *Wide*, during the entire match time. The peaks in the graph indicate that an event might have happened in a crucial time of the game. Note that for an event, not all users will tweet at the same time, some may post their tweets little late, thereby some of the events are just noise as well.

4 Proposed Approach

In this section, we first describe our proposed approach for event detection and then present techniques to improve its performance in accuracy and responsiveness. We will describe our solution in the context of 2017 IPL T20 Cricket games and detect key events that happened during the Cricket games. The architecture of the proposed event detection framework is depicted in figure 9. Our two stage event detection and recognition solution for sports event, such as IPL T20 and ICC Champions is based on two complementing concepts namely, sliding windows and event lexicons.

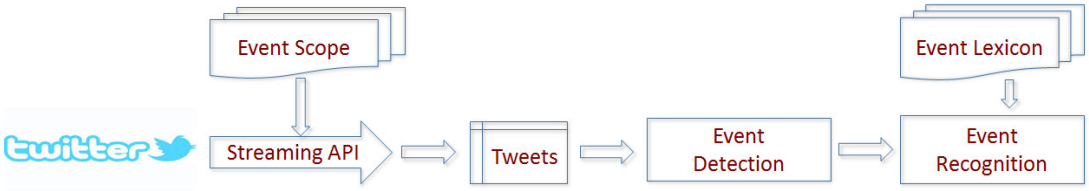


Figure 9. System architecture of *TwitterSports*

Our proposed event detection approach has the following critical steps.

1. Live Tweets collection: Tweets with the event scope are collected from the internet using Streaming Twitter API.
2. Removing noise from live tweets: Noisy tweets are removed from the stream of live tweets for further processing.
3. Adaptive sliding window based event detection: Sliding windows concept is applied to calculate the volume of tweets in a window. If tweet rate of the window is above a threshold, the window is marked as an event.
4. Lexicon-based event recognition: If the window contains enough tweets, it calculates the aggregated votes for the events in the tweets that occur in the middle of a window. Finally, using a lexicon defined for the events, called as *event lexicon*. An event with the maximum vote will be declared the event.

Figure 10 shows the algorithm for the proposed approach.

4.1 Live tweet collection

The rationale of event detection is that an interesting event may immediately trigger its audience and participants to talk about the happenings through social media platforms such as Twitter and Facebook in real-time. The spectators would talk about the event that may have either happened in physical world or reported in mass media such as newspapers, blogs and others. Note that *real-time* means that events need to be discovered as early as possible after they start unraveling in the online social networking service stream. Such information about emerging events can be immensely valuable if it is discovered in near real-time and made available timely to those people who are interested.

Input: Tweets of past 60 seconds
Output: Event name and its tweets
1: create event *lexicon* for pre-determined event types
2: **repeat**
3: *tweets* ← filtered live tweets of past 60 seconds
4: **for** each *tweet* in *tweets* **do**
5: *tweet_frequency* ← number of tweets in each *second*
6: *tweets_per_time* ← tweets in each *second*
7: **end for**
8: *window* = [6,10,20,30,60]
9: **if** *window* does not contain enough tweets **then**
10: select next window
11: **end if**
12: **if** *post rate of window* > pre-defined *threshold* **then**
13: get all tweets that appear in the middle of the window from *tweets_per_time*
14: select an event that has maximum votes using *lexicon*
15: display event name and its tweets using *lexicon*
16: **end if**
17: **until** connection closed

Figure 10. Proposed algorithm for key events detection

Generally, many physical world events, such as product announcements, celebrity deaths and natural disasters like earth quakes, attract a lot of viewers to witness the event. Therefore a sizeable increase in tweets about the physical world event on Tweet will occur, even if a small fraction of viewers talk about the physical event on Twitter. Further, it is a normal human tendency to share the current updates about the physical world event to others. So, people can thus be regarded as sensors who can be leveraged to get updates in real-time. With the help of Streaming API from Twitter, we will be able to collect live tweets continuously and to analyze them so as to detect all events as quickly as possible. The event detection method operates on the stream of live tweets based on the scope of events such as hashtags in Twitter.

4.2 Removing noise from live tweets

Noise elimination is an important preprocessing step for detection and recognition of key events from live tweets which are streamed at real-time. We first remove all tweets which contain up to 3 correct English words. Even though, Twitter specifies the language for those tweets is English, an underlying language used by the viewers is not always English. For example, many viewers of the game transliterate Hindi phrases in English. This step also takes care of short tweets that are unlikely giving us any additional information. Spam tweets are removed by using a dictionary of common words. Similarly, stop words are also removed from the raw tweets. Furthermore, we remove all tweets that contain URLs and pronouns. Obviously, this preprocessing step ensures that signals from tweets dominate noise, otherwise the performance will be very poor for event detection.

4.3 Adaptive sliding window based event detection

Our real-time streamer based on Twitter's Streaming API continuously streams tweets every second which are available in a pre-defined queue. In general, the live tweets that are gathered for the past 60 seconds fixed time window will be considered for a possible event. A fixed window approach usually computes the tweets volume of the first and second half of the window and subsequently calculates the post rate of the window as a ratio between the tweets volume of the second half of the window and the first half of the

window. We detect whether an event just happened by examining the volume of tweets. If the tweets volume is greater than the predefined threshold, the system concludes that some event might have occurred in a particular window time. This works based on a simple rationale that the percentage of change is the post rate obviously indicates the trend of an event related discussion.

Lexicon based event detection utilizing a fixed window of size 60 seconds would suffer from delay issues. Because, it is possible that an event would have occurred during the beginning of the window. Since our event detection will be a real-time detection system, longer delay will dampen the performance of the proposed system. In order to address the real-time challenge and to minimize the delay, we adopt an adaptive window approach. Here, the size of the window will have a significant impact on the tradeoff between the delay and accuracy of event detection. If the size of the window is short, the delay in event detection will also be small. However, the performance of the event detection may be poor, as there may not be several tweets posted during the window and thus post rate would be low.

In order to achieve a better trade-off, the window size should be selected adaptively based on two scenarios. First, viewers of the game have not posted tweets continuously for every second in the current window. That is, tweets were not available in each second of the window. Obviously, the size of the window should be increased. Second, the post rate of the current window is less than the predetermined threshold. Hence, the next window size should be selected automatically. We determine the value for the threshold by analyzing the tweets of the games using our offline dataset of IPL T20 2017 season. The performance of both fixed window and adaptive window approaches is discussed in section 5.3.2 where influence of different fixed window sizes are studied in section 5.3.3.

Our solution is based on an adaptive selection of a sliding window, as depicted in figure 11. The size of the sliding window can be a variable size of 6, 10, 20, 30, 60 seconds. The tweets in a window are sliced into two sub windows which contain tweets in the first half and second half of the window. For example, a window of 10 seconds includes two sub windows of 5 seconds each containing tweets. Our event detection system starts with the smallest window, 6 seconds. First it checks whether there are tweets spread in every second of the selected window. Otherwise, the window size will be incremented to have a size of 10 seconds. The basic assumption is that every key event will result into a continuous tweeting activity by the viewers throughout the window. It also avoids the computation of post rate of the window.

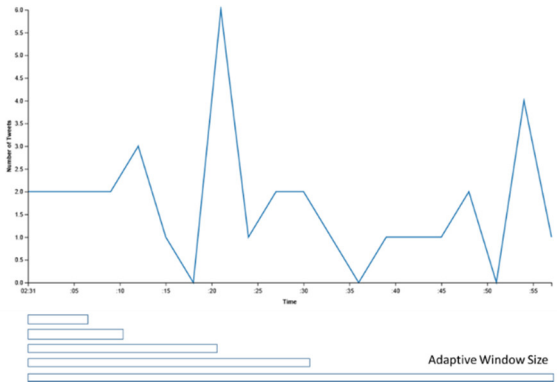


Figure 11. Tweets of one min at 14.31 hours from game DDvKXIP. The adaptive window moves from 6 seconds to 60 seconds.

Second, our event detection system checks whether the post rate of the window exceeds the threshold. If the post rate exceeds the threshold, the system proceeds to recognize the event, such as *sixer*, *catch* etc, otherwise the window size will increment. The post rate ratio is the volume of tweets in the second half window to the volume of tweets in the first half. The threshold value is set to 1.0 based on the analysis of tweets of games in our data set. The value of 1.0 indicates that the post rate in the second half window has to be at least 1.0 times of the post rate in the first half to proceed to recognize an event. The threshold denoting the average number of tweets helps to filter out the small spikes in the tweets. Because, the key events will result into huge spikes in the tweets frequency, which will be above the threshold. Once the events are detected in a time window along the tweet stream, the event represented inside the window is recognized using a lexicon based approach.

4.4 Lexicon-based event recognition

The event recognizer identifies the specific event in the detected event window based on the idea of maximum voting. For simplicity and consistency, we choose all tweets that are available in the middle of the window. We look for the occurrences of event related terms such as event names in each tweet and increase the vote for the occurrence of event related terms. The event recognizer selects the event whose has the maximum votes and declare the event as the winner.

We create a lexicon for 37 events in IPL T20 Cricket games. For example, *boundary*, *sixer*, *catch*, *bowled out*, *run-out*, *lbw* and *leg bye* are some of the key events in Cricket sports. A sample lexicon for the key event *boundary* would contain variants of the event name *boundary* such as *four, fours* and *4* in addition the term *boundary*. The way game viewers tweet the details of events are unique, the vocabulary of lexicons should be more descriptive in order to recognize the designated event. Further, the size of each tweet is limited to 140 characters and the event name is a highly preferred way to describe the event. Therefore, the domain specific lexicon for the key events should precisely describe the event names for the corresponding events. The vocabulary of our lexicon is populated with event terminologies collected from a website ESPNCricInfo[2]. Figure 12 shows a section of our event lexicon for Cricket sports.

```
BOUNDARY = ['boundary', 'four', 'fours', '4']
SIXER = ['sixer', 'six', '6']
ONE = ['one', '1']
CATCH = ['catch', 'c']
BOWLED_OUT = ['bowled out', 'bowled by', 'clean bowled', 'bowled off']
```

Figure 12. Lexicon for few events of *Cricket* sports

Our lexicon based event detection approach is simple in implementation, but at the same time enjoys a better performance for event detection from live tweets which are streamed in real-time. Unlike statistical learning approaches which require training data to build models for further prediction tasks, our lexicon based approach does not require them for event detection and recognition. Furthermore, there are real-time applications to whom training data are not available a priori. For instance, celebrity deaths and terrorist attacks have no training data available in advance. Nevertheless, the keywords related to such

[2] http://www.espncricinfo.com/ci/content/story/239756.html

events are predictable. Therefore, it is very practical to use lexicon based approaches for event recognition.

4.5 Preventing duplicate event alerts

Duplicate events have been an important issue for our event detection method, in which our algorithm reports duplicate events many times even after the specified event had occurred. The viewers of the game continue to discuss even after the event was just now over because this was a key event. For example, in a cricket match, the batsmen keep accumulating runs aggressively and the opponent is in need of a wicket desperately. Now, one of the batsmen gets out due to a catch and thus it is a key event for the audience, triggering a huge discussion. Therefore, our detection algorithm may repetitively recognize the event during the discussion. The primary reason for duplicate event alerts is associated with a shorter window size. Since small number of tweets are analyzed in the short window, the detection algorithm is unable to distinguish whether the short spike in tweets volume denotes a beginning of a new event or the continuation of the current event. Many users, in addition to viewers of a game, also forward the received tweet to other users, as retweets. Therefore, retweets can be considered as noise and retweeting is yet another important reason for the duplicate event alerts.

The solution to duplicate event alerts problem can be approached in two ways. First, we can ignore all retweets coming from the live streaming of tweets, thereby we can minimize the intense discussion over the current key event. Influence of the retweets in detecting events is experimentally studied in section 5.3.5. Second, we can assume that no event of the same kind can happen within 60 seconds. Based on this idea, if the detection algorithm reports the same event again within 60 seconds, we can ignore this event as a duplicate event alert. For example, in Cricket games, an over containing six balls should be delivered by a bowler to a batsman within 5 minutes. So, the process of bowling and batting should be finished within the time frame of 60 seconds. Therefore, an event of the same type cannot happen once again within 60 seconds.

As discussed above, our approach to event detection on live tweets first detects whether an event occurs and then recognizes the specific event type such as *boundary*, *sixer*, *catch* and others. Our method is computationally efficient as it detects key events without analyzing the content of tweets. It detects 37 types of pre-defined key events for Cricket games from live tweets posted by game viewers in real-time.

5 Experimental Results

This section presents the extensive evaluation of the proposed *TwitterSports* approach which we have implemented in Python. The following sections will present the dataset, evaluation criteria and parameter setup and evaluation results respectively.

5.1 Dataset

We collected tweets during game time with the Streaming API using IPL's official hashtags of every game, during 2017 IPL T20 season that was held during April and May 2017. We have successfully crawled tweets at real-time for 44 games. The total file size of the collected tweets is over 6GB. Out of 44 games, we have selected 3 games for detecting key events. Table 1 and 2 show the description of the 2 games. From the tweet statistics in terms of number of tweets and tweet rate, it can be assessed that the game on 16 April 2017, RCB vs RPS is an interesting and most anticipated match.

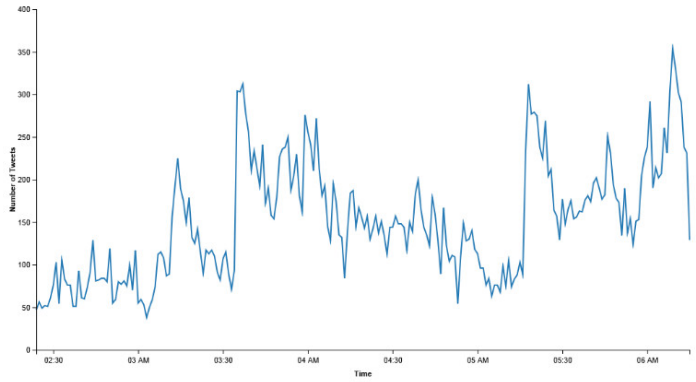
Table 1. Game statistics of RCBvRPS

RCBvRPS game	Total	Total min	Mean (re)tweets per min	Min (re)tweets per min	Max (re)tweets per min	Standard deviation
Tweets	34967	232	150.72	38	354	67.4779909455
Retweets	16162	232	69.664	13	176	34.4786150528

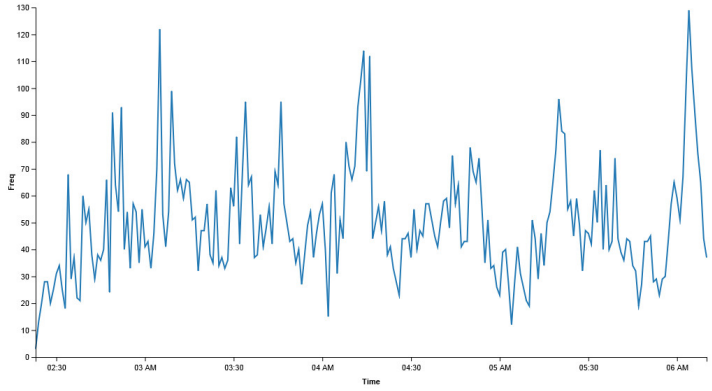
Table 2. Game statistics of DDvKXIP

DDvKXIP game	Total	Total min	Mean (re)tweets per min	Min (re)tweets per min	Max (re)tweets per min	Standard deviation
Tweets	11407	228	50.030	3	129	20.7684377065
Retweets	3473	228	15.234	1	80	9.48236586451

Figure 13 (a) and (b) depict the tweet post rate for games RCBvRPS and DDvKXIP correspondingly. Figures show that the volume of tweets posted during the end the game is high in both games. Both games contain several exciting moments throughout the entire game.



(a)



(b)

Figure 13. Post rate of tweets of games (a) RCBvRPS and (a) DDvKXIP

The ground truth of all events are gathered from the online IPL live commentary site (*http://www.iplt20.com/*). The validity of the event timings have been validated with other IPL commentary sites. Table 3 shows the description of ground truth events in these games.

Table 3. Summary of events in ground truth

Game	No. of ground truth events	No. of Boundaries	No. of Catches	No. of Sixers	Other events
RCBvRPS	81	24	6	9	42
DDvKXIP	89	34	10	8	37

5.2 Evaluation Criteria and Parameter Setup

The performance of the approaches are evaluated using ROC curves and its corresponding Area Under Curve (AUC). Detection result is compared against the ground truth events of a game. A detection is considered a *hit* if the detection event is reported within a particular time (such as 1, 5, 10, 15 minutes) from when the actual event happened. It will be considered a *miss* if the event is not detected within the window time. From this, True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) are computed. Then, True Positive Rate and False Positive Rate are computed as:

$$TPR = TP/(TP+FN) \quad (1)$$

$$FPR = FP/(FP+TN) \quad (2)$$

For a particular approach, different sets of results are obtained by adjusting the tweet rate threshold (0.2, 0.4, 0.5 and 1.0). Different sets of TPRs and FPRs are computed where the True Positive Rates are plotted against False Positive Rates which results in ROC. For completeness of the curve, the curve starts and ends in the coordinates (0,0) and (1,1) respectively. In addition, Area Under Curve of the ROC is calculated for each approach. The AUC of each ROC line depicts the degree of performance in terms of detection accuracy where AUC is high when true positive rate is high and false positive rate is low and AUC is low when true positive rate is low and false positive rate is high.

5.3 Results

We evaluate *TwitterSports* using IPL T20 games and show the accuracy of the event detection for major events such as *boundary*, *catch*, *sixer* and *boundary+catch*. The evaluation results prove that the adaptive window approach can detect key events faster than fixed window approach.

5.3.1 Performance on detecting events

Figure 14 shows the performance of the adaptive window approach in detecting different events such as *boundary*, *catch*, *sixer* and major events (*boundary+catch*) in RCBvRPS game. The evaluation is conducted for different evaluation window sizes such as 1, 5, 10 and 15 minutes.

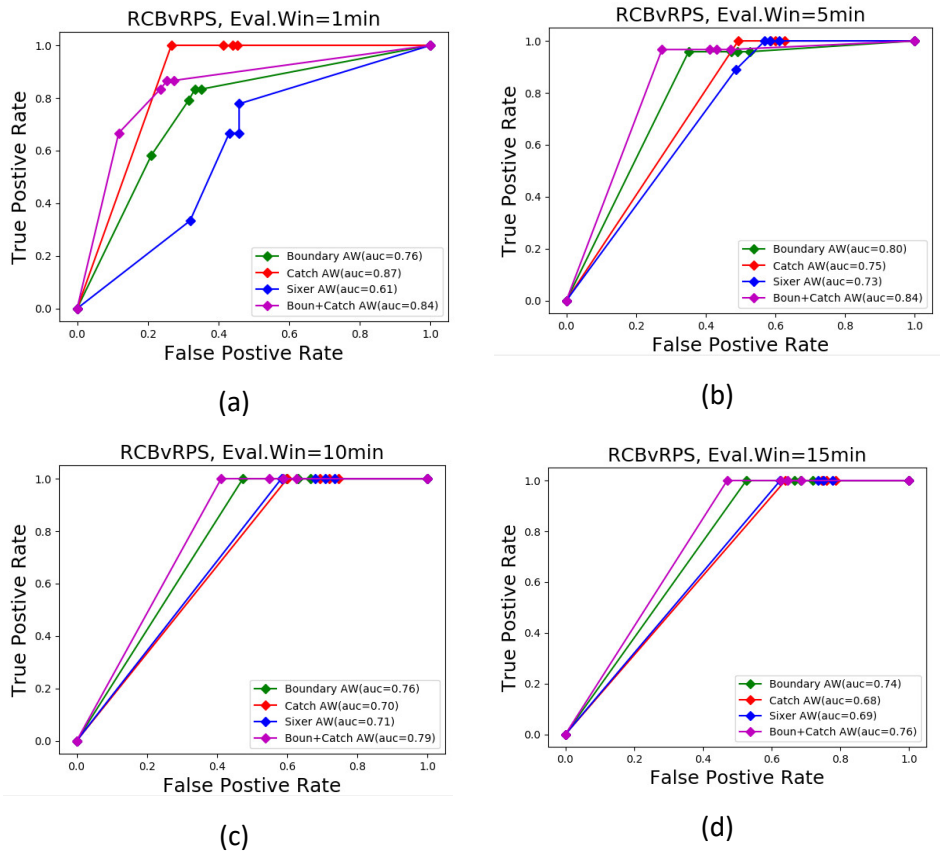


Figure 14. Detection performance for individual events

Results show that the adaptive window approach gives decent performance in detecting the game events. Boundary events are detected quicker than the *sixer* and *catch* events even within an evaluation window of 1 minute. Major events (*boundary+catch* combined) of a Cricket game are also detected well with the adaptive window based approach. Almost all game events are detected well with decent accuracy within an evaluation window of size 5 minutes where the performance for evaluation windows 10 and 15 minutes are highly similar. It shows that most of the events are detected and reported well even within a time of 5 minutes from the actual game event happened. Here performance in detecting each event can be directly assessed by the AUC of the ROC line.

5.3.2 Performance of fixed vs. adaptive windows

Figure 15 shows the performance of adaptive window and fixed window approaches in detecting major events for different evaluation windows. The results are shown for performance in both games. The evaluation for comparing fixed and adaptive window approaches is conducted for different evaluation window sizes such as 1, 5, 10 and 15 minutes in detecting major events of the games DDvKXIP (Figure15 (a), (b), (c) and (d)) and RCBvRPS (Figure15 (e), (f), (g) and (h)). For fixed window approach, the performance is also shown for different fixed window sizes such as 6, 10, 20 and 30.

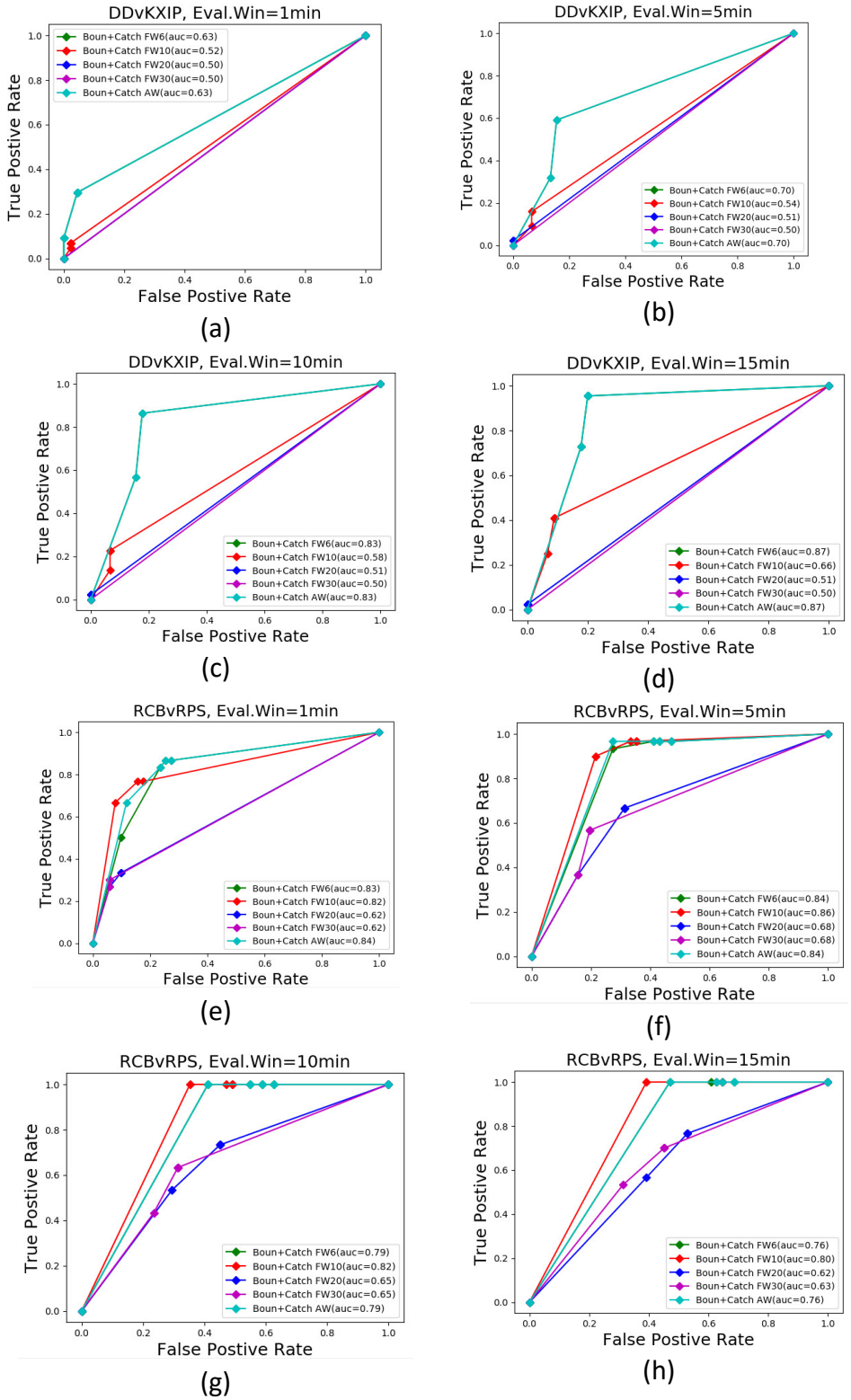


Figure 15. Detection performance of fixed and adaptive window approaches

In game DDvKXIP, both adaptive window approach and fixed window approach (window size 6) show similar performance under different evaluation windows. In RCBvRPS game, adaptive window approach outperforms fixed window approach in small evaluation window where fixed window approach (window

size 10) marginally outperforms adaptive window approach in other evaluation window sizes. From the results we can assess that there is no standard best performing window size for fixed window approach. Since the best performing window size for fixed window approach is changing for different games and cannot be known forehand, adaptive window approach is mostly preferred for any game.

5.3.3 Performance of fixed window for different windows sizes

Figure 16 shows the performance of fixed window approach for different window sizes such as 6, 10, 20 and 30 in detecting the *boundary*, *catch* and *sixer* events of a game. The evaluation is conducted for different evaluation window sizes for the game RCBvRPS.

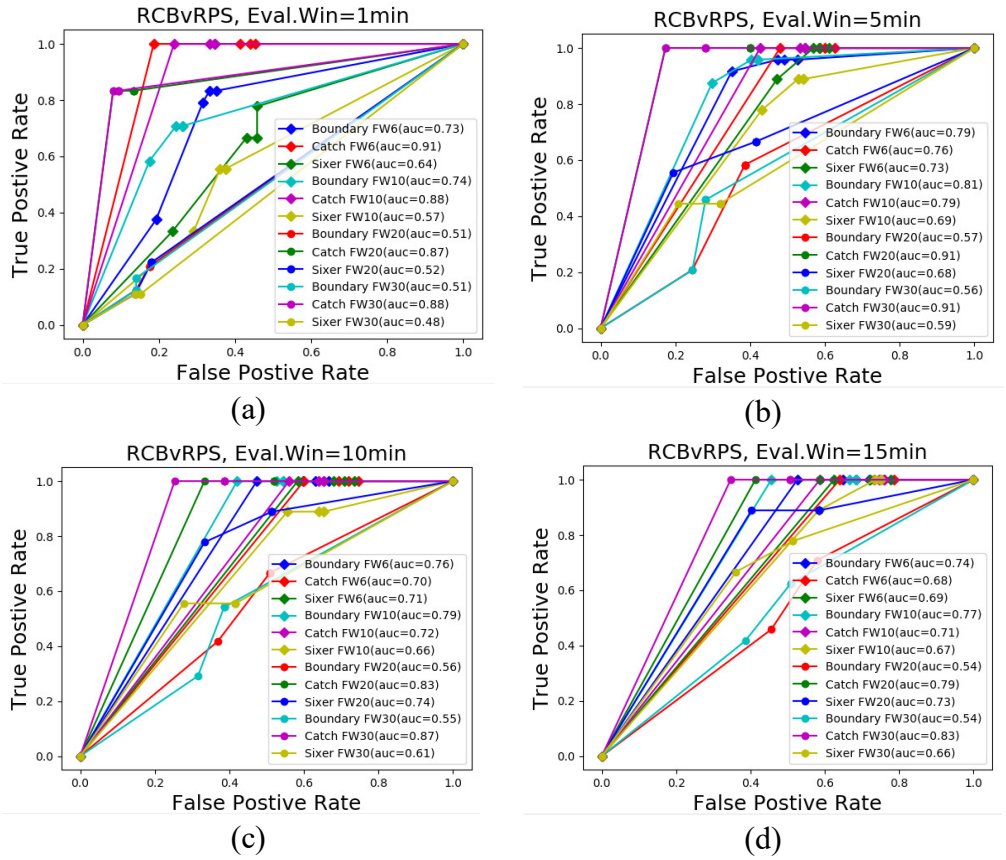


Figure 16. Detection performance of fixed widow approach with different window sizes

For a small evaluation window 1 minute, detection performance degrades as the size of the fixed window increases for almost all the events. However, for a bigger evaluation window 5 minute, detection accuracy improves for medium sized fixed windows such as 10 and 20 for detecting events such as *catch*. It can be concluded that for detecting events quickly in real-time, fixed windows with smaller sized windows can be preferred. However, for consistent performance, medium sized windows can be used in situations where bigger event detection time like 5 minutes is considered. As discussed in section 5.3.1, the performance is similar for evaluation window sizes 10 and 15 minutes.

5.3.4 Performance under different evaluation windows

Figure 17 depicts the detailed performance in detection of individual events under different evaluation window sizes such as 1, 5, 10 and 15 minutes. This evaluation is conducted for gauging the effect of

different evaluation window sizes by performing adaptive window approach in detecting events of the game RCBvRPS.

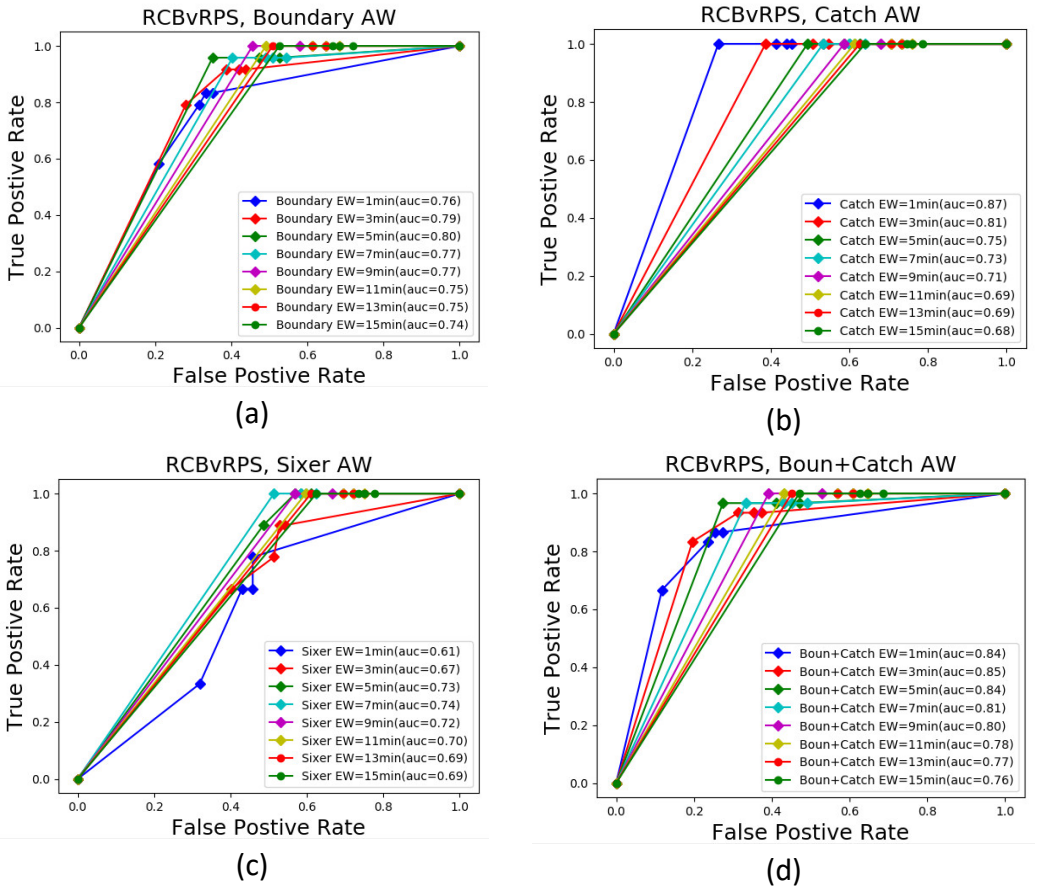


Figure 17. Detection performance for different evaluation windows

The results show that *catch* event is quickly detected within evaluation window size of 1 minute where the performance degrades as the evaluation window size increases. Events such as *boundary* and *sixer* are detected well within a time delay of 5 minutes where the accuracy reduces as the evaluation window size is increased further. Similarly major events (*boundary+catch* events) are detected well within smaller evaluation window sizes such as 1, 3 and 5 minutes where larger evaluation window sizes show lesser detection accuracy.

5.3.5 Performance of all tweets vs. no retweets

Since the processed tweets contain both tweets and their corresponding retweets, influence of retweets in detecting events is analyzed by evaluating the adaptive window approach with *all tweets* and with *no retweets*. The evaluation is conducted for adaptive window based approach in detecting *boundary*, *catch* and *sixer* events in the game RCBvRPS under different evaluation windows as depicted in figure 18.

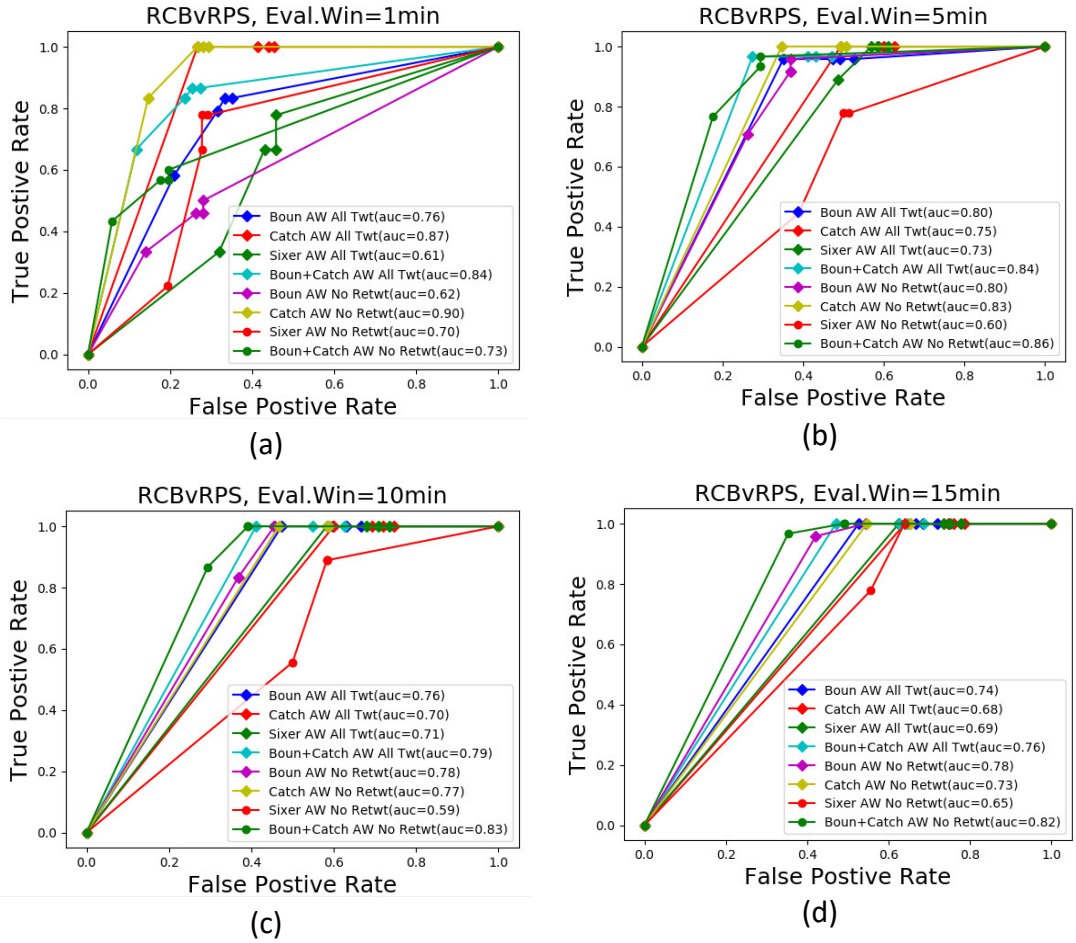


Figure 18. Detection performance for *all tweets* and *no retweets*

The results show that *catch* events are detected well within small evaluation windows (for ex. 1 and 3 minutes) when *no retweets* are considered for detection. The detection improves as the evaluation window size increases where tweets including the retweets help the event detection. However, *boundary* events are detected easily when *all tweets* are considered along with retweets. Similarly, as the evaluation window size increases the *sixer* events are detected well when all tweets are considered. Overall, it can be concluded that for quick detection of rare events such as catches, removal of retweets helps the detector. Also, inclusion of retweets highly influences detection accuracy if considerable time delay in detection is permissible i.e. bigger window sizes.

5.4 Limitations of *TwitterSports* and Twitter

Although our *TwitterSports* is a simple, yet powerful lexicon based event detection algorithm, performance will be directly impacted by the amount of tweets posted by the viewers. The algorithm needs enough tweets so that the post rate would be above the predefined threshold. This will enable our event detector to recognize a key event, otherwise the detection algorithm will perform badly. We can notice such an average performance of our *TwitterSports* in the game DDvKXIP for a key event *sixer*. The area under ROC value is just an average and below average value for the evaluation windows 1, 5, 10 and 15 minutes as shown in figure 19.

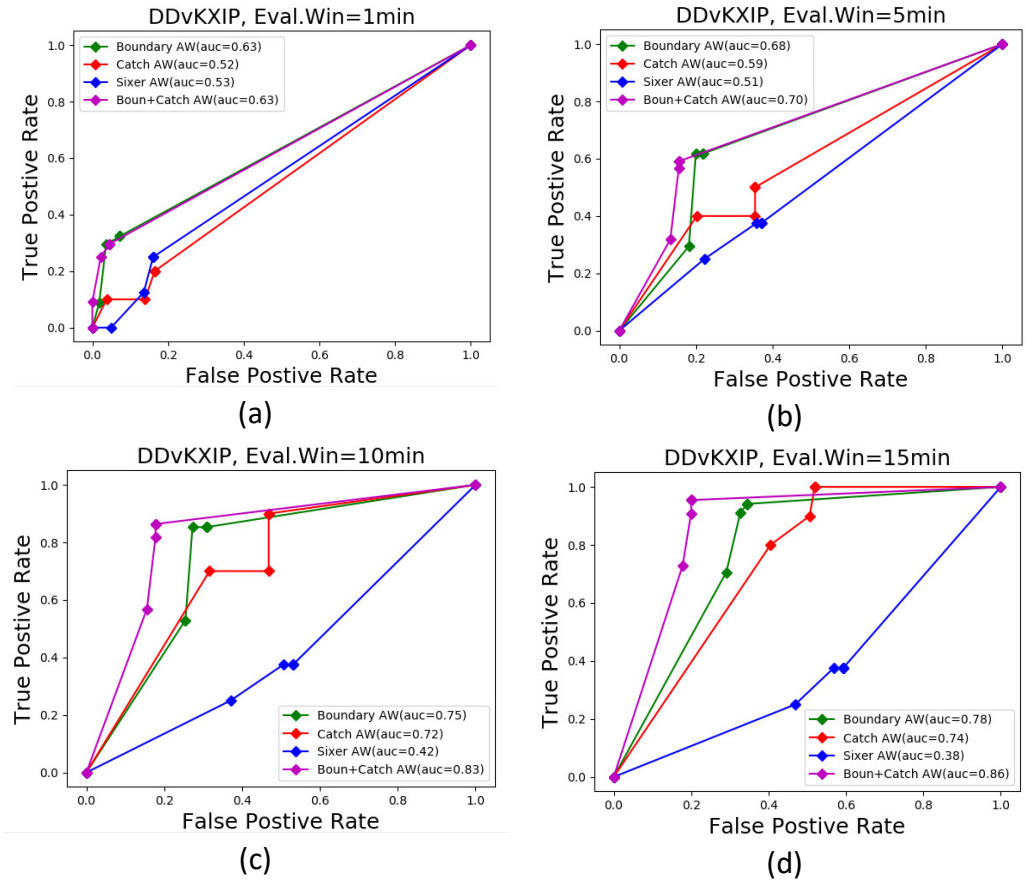


Figure 19. Detection performance of key events from DDvKXIP game

Performance of our Twitter event detection approach *TwitterSports*, also depends on many factors. One such factor is *Delay* or *Latency* in the flow of signals from the human sensors in the social media. Delay plays a vital role in determining the performance of a real-time event detection system. There are three types of delays encountered in the Twitter social media, namely human delay, Twitter delay, and processing delay [3]. Many applications require game events to be detected at real-time. So that, the information provided to society will be meaningful. Therefore, we need to analyze the delay of our *TwitterSports* approach in detecting key events.

Human delay is a period of time between a user observing an event such as *boundary* and typing and posting a tweet about this event. Human delay depends on how fast humans observe, react and publish tweet about the event to the social media, such as Twitter. In this case, delay is caused by the humans in various degrees. For example, delay in reporting the events depend on the user's interest in reporting the event, ability to type faster, type device (PC, laptop, or mobile) of used for tweeting, type of mobile (Nokia, Blackberry, iPhone, Samsung etc), speed of internet, location of the user (watching the match live in stadium or in TV in home).

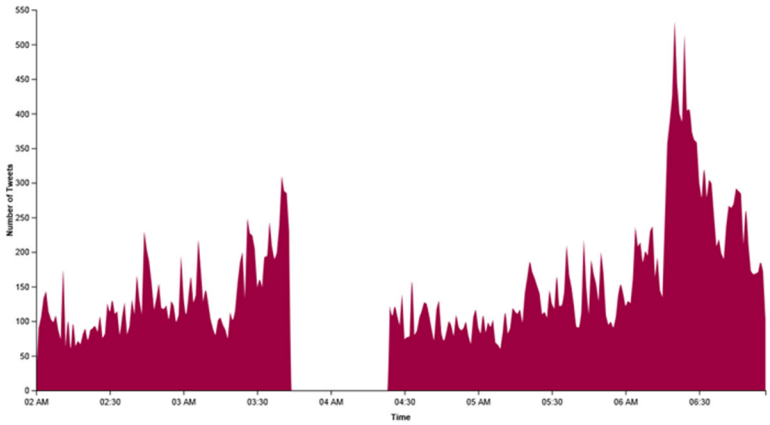


Figure 20. Delay in Twitter streaming of tweets for game RPSvMI held on 21 May 2017. There were no tweets streamed from 3.44 PM to 4.24 PM, which includes an innings break of 15mins.

Twitter also introduces a delay in providing tweets to the querying users. On the technical part of the data streaming, Twitter often faces heavy workload from the millions of users and its indexing mechanism leads to certain delay in delivering the relevant tweets to the crawler through its API. One way to find the delay is to compare the timestamp of the tweet and the timestamp when we acquire the tweet. There is no explicit mentioning about the degree of delay introduced by Twitter API. One such Twitter API issue can be noted in figure 20 where there was no streaming of Tweets for 40 minutes despite there were no internet or API connection problem for streaming. Due to the different Twitter indexing mechanisms, sometimes the choice of keywords for crawling Twitter also makes difference in retrieval speed of tweets. Therefore, the quality of search predicates defined by queries for crawling is another factor for delay.

In addition, minimal amount of delay is also introduced by our event detection engine. This is, due to the processing time involved in data collection and analysis of data. Due to the faster nature of the proposed approach, this delay is highly reduced during the processing. In case of extremely high tweet rate, this analysis delay can be reduced by using parallel processing.

6 Conclusion

In contrast to the existing event detection approaches for sports domain, *TwitterSports*, a novel real-time event detection approach is presented in this paper. *Twittersports* is based on two complementary ideas event lexicon and adaptive sliding windows. An event is declared detected when the post rate of the adaptive window exceeds the pre-defined threshold. The detected event is recognized utilizing the domain specific event lexicon for Cricket sports. The event lexicon can handle all possible name variations for a possible key event and greatly reduces the computation overhead by eliminating the need for a natural language preprocessing. Further, the predefined threshold efficiently reduces noise by ignoring small spikes in the volume of tweets.

Results of the extensive experiments have shown the efficacy of the proposed adaptive window based event detection. In the experiments, it was found that certain events such as boundary are detected easily and quickly than other events, which are found to be more appealing to the audience to react quickly in Twitter. Comparative evaluation between fixed and adaptive window based detection has revealed the advantages of using an adaptive window approach for robust and consistent performance. Influence of different fixed window sizes were also analyzed in the experimental evaluation. It was found that different fixed windows sizes perform differently in detecting each event in Cricket game. The time delay in

detecting the events were studied using evaluation with different evaluation window sizes. Influence of including retweets for detection of different events were analyzed. Challenges in real time scenario in terms of various delays such as human, streaming API and processing delays were also discussed.

There are many challenges left in the area of real time event detection. The proposed event detection algorithm detects only events that are documented in the event lexicon. It would be useful to detect interesting but unexpected events that are not included in the lexicon by performing NLP preprocessing steps.

REFERENCES

[1] Boyd, D. M and N. B. Ellison. *Social network sites: Definition, history, and scholarship*. Journal of Computer-Mediated Communication, 2007. 13(1): p. 210–230.

[2] Atefeh, F and Khreich, W. *A survey of techniques for event detection in twitter*. Computational Intelligence, 2015. 31(1): p. 132-164.

[3] Zhao, S., Zhong, L., Wickramasuriya, J., Vasudevan, V., LiKamWa, R and Rahmati, A. *Sportsense: Real-time detection of NFL game events from Twitter*. ArXiv preprint, 2012. arXiv:1205.3212.

[4] Zhao, D and M. B. Rosson. *How and why people Twitter: The role that micro-blogging plays in informal communication at work*. In Proc. ACM International Conference on Supporting Group Work, GROUP '09, ACM, New York, NY, 2009. p. 243–252.

[5] Hurlock, J and M. Wilson. *Searching Twitter: separating the tweet from the chaff*. In Proc. International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 2011.

[6] Jiang, L., M. Yu., M. Zhou., X. Liu and T. Zhao. *Target-dependent Twitter sentiment classification*. In Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – vol. 1, HLT '11, ACL, Stroudsburg, PA, 2011. p. 151–160.

[7] Amer-yahia, S., S. Anjum, A. Ghenai, A. Siddique, S. Abbar, S. Madden, A. Marcus and M. El-haddad. *MAQSA: A system for social analytics on news.* In Proc. ACM SIGMOD International Conference on Management of Data, SIGMOD '12, ACM, New York, NY, 2012. p. 653–656.

[8] Tumasjan, A., T. O. Sprenger., P. G. Sandner and I. M. Welpe. *Predicting elections with Twitter: What 140 characters reveal about political sentiment.* In Proc. 4th International Conference on Weblogs and Social Media, ICWSM. The AAAI Press: Washington, DC, 2010.

[9] Wang, X., Gerber, M. S and D. E. Brown. *Automatic crime prediction using events extracted from Twitter posts*. In Proc. 5th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction, SBP'12. Springer-Verlag: Berlin, Heidelberg, 2012. p. 231–238.

[10] Troncy, R., B. Malocha and A. T. S. Fialho. *Linking events with media*. In Proc. 6th International Conference on Semantic Systems, I-SEMANTICS '10, ACM, New York, NY, 2010. 42: p. 1–42:4.

[11] Zhao, S., Zhong, L., Wickramasuriya, J and Vasudevan, V. *Human as real-time sensors of social and physical*

events: A case study of twitter and sports games. ArXiv preprint, 2011. arXiv:1106.4300.

[12] Hasan, M, Orgun, M. A and Schwitter, R. *A survey on real-time event detection from the Twitter data stream.* Journal of Information Science, 2017. 0165551517698564.

[13] T. Sakaki, M. Okazaki and Y. Matsuo. *Earthquake shakes Twitter users: real-time event detection by social sensors*. In Proc. ACM WWW '10, 2010.

[14] Y. Qu, C. Huang, P. Zhang and J. Zhang. *Microblogging after a major disaster in China: a case study of the 2010 Yushu earthquake*. In Proc. ACM 2011 conference on Computer supported cooperative work, 2011.

[15] S. Vieweg, A. L. Hughes, K. Starbird and L. Palen. *Microblogging during two natural hazards events: what twitter may contribute to situational awareness*. In Proc. ACM CHI '10, 2010.

[16] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman and J. Sperling. *TwitterStand: news in tweets.* In Proc. ACM SIGSPATIAL, 2009.

[17] J. Hannon, K. McCarthy, J. Lynch and B. Smyth. *Personalized and automatic social summarization of events in video.* In Proc. ACM IUI, 2011.

[18] D. Chakrabarti and K. Punera. *Event Summarization using Tweets*. In Proc. AAAI ICWSM, 2011.

[19] Ekin, A. M. Tekalp and R. Mehrotra. *Automatic soccer video analysis and summarization*, Image Processing, IEEE Transactions on, 2003. 12: p. 796-807.

[20] Y. Rui, A. Gupta and A. Acero. *Automatically extracting highlights for TV Baseball programs*. In Proc. ACM Multimedia 2000.

[21] D. Zhang and S.-F. Chang. *Event detection in baseball video using superimposed caption recognition*. In Proc. ACM Multimedia, 2002.

[22] K. Petridis, S. Bloehdorn, C. Saathoff, N. Simou, S. Dasiopoulou, V. Tzouvaras, S. Handschuh, Y. Avrithis, Y. Kompatsiaris, and S. Staab. *Knowledge representation and semantic annotation of multimedia content*. *Vision*, Image and Signal Processing, IEE Proceedings, 2006. 153: p. 255-262.

[23] C. Xu, Y.-F. Zhang, G. Zhu, Y. Rui, H. Lu and Q. Huang. *Using Webcast Text for Semantic Event Detection in Broadcast Sports Video*, Multimedia, IEEE Transactions on, 2008. 10: p. 1342-1355

[24] Mathioudakis, M and Koudas, N. *TwitterMonitor: Trend Detection over the Twitter Stream*. In Proc. SIGMOD/ PODS, 2010. p. 1155–1158.

[25] Weng, J and Lee, B.-S. *Event Detection in Twitter.* In Proc. ICWSM, 2011. p. 401–408.

[26] Shane Fitzpatrick. *Improving new event detection in social streams*. 2014. Master Thesis.

[27] Petrovi´c, S., Osborne, M and Lavrenko, V. *Streaming First Story Detection with Application to Twitter*. In Proc. NAACL HLT, 2010. p. 181–189.

[28] Becker, H., Naaman, M and Gravano, L. *Beyond Trending Topics: Real-Wrold Event Identification on Twitter*. In Proc. ICWSM, 2011. 11: p. 438–441.

[29] Cataldi, M., Di Caro, L and Schifanella, C. *Emerging Topic Detection on Twitter Based on Temporal and Social Terms Evaluation*. In Proc. MDM/KDD, 2010. p. 4:1–10.

[30] Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S and Miller, R. C. *TwitInfo: Aggregating and Visualizing Microblogs for Event Exploration.* In Proc. CHI, 2011. p. 227–236.

[31] Valkanas, G and Gunopulos, D. *How the Live Web Feels About Events*. In Proc. In Proc. 22nd ACM International Conference on Information and Knowledge Management CIKM, 2013. p. 639–648.

[32] Popescu, A. M and M. Pennacchiotti. *Detecting controversial events from Twitter*. In Proc. 19th ACM International Conference on Information and Knowledge Management, CIKM '10, ACM, New York, NY, 2010. p. 1873–1876.

[33] Benson, E., A. Haghighi and R. Barzilay. *Event discovery in social media feeds*. In Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, HLT '11, Association for Computational Linguistics, Stroudsburg, PA, 2011. p. 389–398.

[34] Lee, R and K. Sumiya. *Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection.* In Proc. 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, LBSN '10, ACM, New York, NY, 2010. p. 1–10.

[35] Sakaki, T., M. Okazaki and Y. Matsuo. *Earthquake shakes Twitter users: Real-time event detection by social sensors*. In Proc. 19th International Conference on World Wide Web, WWW '10, ACM, New York, NY, 2010. p. 851–860.

[36] Becker, H., F. Chen, D. Iter, M. Naaman and L. Gravano. *Automatic identification and presentation of Twitter content for planned events.* In Proc. International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 2011.

[37] Becker, H., M. Naaman and L. Gravano. *Selecting quality Twitter content for events*. In Proc. International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 2011b.

[38] Massoudi, K., M. Tsagkias, M. De Rijke and W. Weerkamp. *Incorporating query expansion and quality indicators in searching microblog posts*. In Proc. 33rd European Conference on Advances in Information Retrieval, ECIR'11. Springer-Verlag: Berlin, Heidelberg, 2011. p. 362–367.

[39] Weerkamp, W and M. De Rijke. *Credibility improves topical blog post retrieval.* In Proc. ACL, Columbus, OH, 2008. p. 923–931.

[40] Gu, H., X. Xie, Q. Lv, Y. Ruan and L. Shang. *ETree: Effective and efficient event modeling for real-time online social media*. In Proc. Web Intelligence and Intelligent Agent Technology, WI-IAT 2011, IEEE/WIC/ACM International Conference, 2011. 1: p. 300–307.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Volume 5 No. 6
ISSN 2054-7390

Academic Performance: Text Anxiety during examinations of Freshmen Engineering students MLRIT, Hyderabad

[1]Dr. P. Bhaskara Reddy, [2]M. Kantha Reddy, [3]R.Praveen Reddy
[1]Department of Electronics & Communication Engineering
MLR Institute of Technology, Dundigal, Hyderabad, Telangana, India
[2]Department of Computer Science Engineering IUCEE, Hyderabad, Telangana, India
[3]Department of Master of Business Administration, MLR Institute of Technology
Dundigal, Hyderabad, Telangana, India
pbhaskarareddy@rediffmail.com; kanthareddy_m@yahoo.com; rpreddy82@gmail.com

ABSTRACT

The main aim of this research was to find the relationship between test anxiety and academic performance of students at the under graduate level. A sample of 414 students was randomly selected from five different departments of Freshman Engineering at MLR Institute of Technology in Hyderabad, TS, India. Data was gathered by utilizing the Test Anxiety Inventory (TAI) developed and improved by Spielberger. Pearson correlation, multivariate English and regression analysis was run for data. It was found that a major negative relationship observed between test anxiety scores and students performance scores. Results proved that a cognitive factor contributes more in test anxiety than effective factors. Therefore, it is examined that test anxiety is one of the key factors that results the students' underachievement and poor performance but it can be handled by rigorous training of individual students in linked with factors resulting test anxiety.

Keywords: Potential, Emotional, Performance, Worry, Inventory

1 Introduction

Tests and examinations at all stages of education, especially at higher education level have been considered an important and powerful tool for decision making in our competitive society, with people of all ages being evaluated with respect to their achievement, skills and abilities. Zollar and Ben-chain (1990) have the opinion that "the era in which we live is a test- conscious age in which the lives of many people are not only greatly influenced, but are also determined by their test performance". Test and examination stress is thought to prevent some individuals from reaching their academic potential. It has been found that students consistently perceive examination as a source of increase in anxiety and a situation engulfed with uncertainty/unfairness in letting them demonstrate their true achievements (Zollar & Ben-chain, 1990; Spielberger, 1985). Such feelings among students' limit their potential performance during the test situation, resulting in higher text anxiety (Hill & Wigfield, 1984) directly causing drop in the student achievement. Therefore, it can be seen as a measurement error towards measuring student achievement as tests are not meant to measure student achievement under intimidating situation but to know their level of achievement in an environment fair enough to let them demonstrate their abilities to the fullest.

DOI: 10.14738/tmlai.56.3505
Publication Date: 11th Oct, 2017
URL: http://dx.doi.org/10.14738/tmlai.56.3505

The researchers have suggested various means to minimize test anxiety with managing external factors like environment of examination hall; behaviour of examiners etc. internal factors like organization of questions in a test, sufficient description of the context, clarity in instruction for students etc. Despite these measures to minimize test anxiety it is generally agreed that it has become most upsetting and a disruptive factor for students. There are number of researches reporting text anxiety as one of the major cause for students' underachievement and low performances at different levels of their educational life (Oludipe, 2009) and has been shown to affect students' ability to profit from instruction (Schonwetler, 1995).

It is worth discussing some studies showing the statistically significant inverse relationship between test anxiety and students' achievement since long time. Gaudry and Spielberger (1971) discussed that high test anxiety is considered as one of the main factor for low performance of students at university level. A study conducted by Nicholson (2009) to explore the effects of test anxiety on student achievement of grade 11 students, revealed that anxiety and achievement are related to each other. Khalid and Hasan (2009) conducted a study on a purposively selected sample of 187 undergraduate students to explore the relationship between test anxiety and academic achievement and found that students with academic achievement have low test anxiety scores and vice versa. Chapell, Blanding, Takahashi, Silverstein, Newman, Gubi, and McCann (2005) conducted a research study to explore the relationship between test anxiety and academic performance. They collected data from a large sample of graduate and undergraduate students and found a significant and negative relationship between test anxiety and academic achievement.

Hancock (2001) investigated the effects of students' test anxiety and teacher's evaluation practices on students' achievement and motivation at post the secondary level**.** He found statistically significant results which revealed that all students, especially students with high anxiety level**,** performed poorly and were less motivated to learn. Thus he concluded that that when students who are particularly test-anxious are exposed to a highly evaluative assessment environment in their educational institution, they perform poorly and are less motivated to perform (Hancock, 2001). A research study conducted by Cassady & Johnson (2002) "to investigate the effect of cognitive test anxiety on students' academic performance and found that cognitive test anxiety exerts a significant stable and negative impact on academic performance measures".. Albero, Brown, Eliason & Wind (1997), on the basis of their research study, concluded that students having high test anxiety had significantly lower scores. Oludipe (2009) conducted a study to explore how test anxiety affects students' performance levels in the sciences, especially in Computational Methods & Integral Calculus, and concluded that "low test- anxious students performed better than high test-anxious students on both numerical and non-numerical tasks in Computational Methods & Integral Calculus". On the other hand, Schonwetter, (1995) by relating this phenomenon to classroom instruction, the researchers further discussed "how high test- anxious students were unable to benefit directly from organized instruction, which ultimately affected their performance in class".

Several researchers explored gender differences with respect to test anxiety and found that females have higher levels of overall test anxiety than males (Chapell et al., 2005; Cassady & Johnson, 2002; Bandalos et al., 1995; Mwamwenda, 1994). Cassady & Johnson, (2002) explained "that one explanation for differences in test anxiety on the basis of students' gender is that males and females feel same levels of test worry, but females have higher levels of emotionality". Zeidner (1990), on the basis of his research,

concluded that difference in test anxiety scores of male and female is due to gender difference in scholastic ability.

It is quite evident from the arguments given above and results of the studies reported that text anxiety affects achievement along with other variables such as motivation to learn, ability to benefit from formal instruction and gender. This diversification of effects of text anxiety lead researchers to think of text anxiety as at least bi-dimensional construct (Berk & Nanda, 2006; Chapell et al., 2005; Cassady & Johnson, 2002; Diaz, 2001) with affective and cognitive components. The affective dimension (emotionality) refers to behavioural or physical reactions to testing situations, such as fear, nervousness, and physical discomfort (Hanckock, 2001; Pintrich & Schunk, 1996; Williams, 1994). This high level of emotionality is evident through physiological responses experienced during evaluative situations (Cassady & Johnson, 2002). The cognitive dimension (worry) refers to cognitive concerns about performance, such as worry about the testing situation or negative performance expectations (Humbree, 1988; Morris, Davis, & Hutchings, 1981; Depreeuw, 1984) .It is the cognitive aspect of test anxiety which has been significantly accounted for declines in academic achievement of adolescents and postsecondary students (Bandlos, Yates, & Thorndike-Christ, 1995; Williams, 1991; Humbree, 1981).

The discussion above has intrigued researchers to investigate text anxiety as a contributing factor in student achievement among Indian students in institutions of higher education as it is generally perceive that institutions of higher education in India have very rigid system of tests/examination having high stakes in students' academic career. The study addressed following questions to pursue the above stated broader objective.

1. Determine the relationship between the Test Anxiety total scale scores and academic achievement scores of students in different science subjects.
2. Determine the relationship between the Test Anxiety Emotional scale scores and academic achievement scores of students in different science subjects.
3. Determine the relationship between the Test Anxiety Worry scale scores and academic achievement scores of students in different science subjects.

2 Research Methodology

This study being a descriptive in nature utilized survey techniques. This section will describe sample, research instrument and procedure of the data collection.

2.1 Sample

Five departments were randomly selected from the science faculty of a MLR Institute of Technology in Hyderabad, India**.** From each selected department; intact classes were used in the sample. As a result, sample comprised of 414 randomly selected under graduate students (Male = 124, Female = 290). The detail distribution of sample is given in table 1.

Table 1: Detail of sample of the study

Department	Number of Students		
	Male	Female	Total
Differential Equations and Applications	9	19	28
Applied Computational Methods & Integral Calculus - I	20	06	26
Engineering Chemistry	20	50	70
Computational Methods & Integral Calculus	34	28	62
Computer Programming	12	45	57
English	25	77	102
Engineering Drawing	04	65	69
Total	124	290	414

3 Research Instrument

There are several instruments developed by various authors for measuring test anxiety but they all use text anxiety as unitary construct. Thus they insist on finding a unitary number representing text anxiety level of students. As mentioned earlier in this paper that this research is based on assumption that test anxiety is at least bi-dimensional construct comprising of emotionality and worry scale. Thus, researchers preferred using Test Anxiety Inventory (TAI) to capture the bi-dimensionality of the selected construct. The same argument is put forward by the Smith (2000) while using this instrument for his study. He compared different test anxiety scales (Test anxiety scale by Sarason, 1978; Test anxiety questionnaire by Mandler & Sarson, 1952; and the State-trait anxiety inventory by Spielberger, Gorsuch, & Luschene, 1970) to conclude that they yield global test anxiety scores that combine components, emotionality and worry, of test scores". Whereas, as discussed above, researchers considered test anxiety, a bi- dimensional construct, and when someone intends to study the influence of test anxiety on academic achievement, it is necessary to study both components of test anxiety because of the fact that these both factors are related to academic performance (Berk & Nanda, 2006; Chapell et al., 2005; Cassady & Johnson, 2002; Hanckock, 2001; Smith, 2000; Pintrich & Schunk, 1996; Bandlos, Yates, & Thorndike-Christ, 1995; Williams, 1994;Williams, 1991, & Humbree, 1981).

Table 2: Description of sub-constructs, their scope, number of items, example items and reliability of research instrument

Sub-construct	Scope	No of items	Example item	[1]Reliability range (α)	[2]Reliability (α)
Emotionality	Behavioral or physical reactions to testing situations, such as fear, nervousness, and physical discomfort.	8	2. While taking examination I have an uneasy upset feeling	0.85 to 0.91	0.767
Worry	Cognitive concerns about performance, such as worry about the testing situation or negative performance expectations.	8	6. The harder I work at taking a test the more confused I get.	0.83 to 0.91	0.720
TAI total	General feeling about the test anxiety in addition to items already included in emotionality and worry scale. The Test Anxiety Inventory) TAI total score.	20 (16+4*)	13. During important tests I am so tense that my stomach gets upset.	0.92 to 0.96	0.868

[1]as reported in other studies. [2]as reported found in this study.

*There were four items in the scale not included in any sub-construct but were part of the Total score.

Chapell, Blanding, Takahashi, Silverstein, Newman, Gubi, and McCann (2005) reported that test anxiety inventory is extensively used to explore students test anxiety at different levels of education all over the world. Table 2 shows that TAI comprised of 20 Likert Scale type self-report items (Four point scale: indicating from "Almost never" to "Almost always") which are designed by its author (Spielberger, 1980) to measure test anxiety symptoms. The scale is further divided into two subscales: Worry Scale (8 items), and Emotional Scale (8 items). Cronbach alpha (α) reliability coefficient reported for total scale (TAI-Total) ranged from 0.92 to 0.96 and for its two sub-scales: Worry scale (0.83 to 0.91) and Emotional scale (0.85 to 0.91). For present study, the Cronbach Alpha (α) for total scale was 0.868**,** while the reliability for emotional scale items was 0.767 and for worry scale items was 0.720. The difference in the reliability found in other studies and present study is due do difference in sample size as reliability is directly proportional to number of subjects in sample. Despite difference in reliability on each sub-scale the values of alpha (α) are reasonably high and statistically acceptable.

To collect information about demographic variables, a Demographic Variables Information Proforma was developed by the researchers. It was comprised of information regarding a student's gender, department, semester and achievement scores (Achievement scores were verified by the officials of concerned departments).

4 Procedure of Data Collection

The data was collected personally by the researchers with prior arrangement with the department concerned and teachers. Intact classes were used for this purpose. To avoid any measurement related error, standardization of procedure was insured by giving uniform instruction to students, each time the data was collected. Similar instruction, environment, and execution timing was provided to students in each department during data collection. The consent of the participants, privacy of information collected and other ethical sureties were provided to the participants.

5 Analysis and Interpretation of Data

Data were analyzed by using SPSS-15 Software Package. Descriptive English in table 3 were to provide an understanding of the dimensions of data while inferential analysis focused on finding the relationship of emotionality scale, worry scale and TAI total score with student achievement as described in research questions.

Table 3 exhibited the descriptive values for Emotionality scale scores, Worry scale scores, Total test anxiety scale scores and achievement scores for male and female students in different departments. It is evident from table that for emotionality component, mean value is ranging between a minimum of 15.36 for male students of department of English to a maximum of 20.00 for female students studying in department of Engineering Chemistry. Similarly, female students of department of English possess a minimum mean score of 16.16 on worry aspect of test anxiety to a maximum of 20.31 for male students of Engineering Chemistry department.

Table 3: Descriptive English by department, gender, test anxiety sub-scales and student achievement

Department/ Gender		Emotionality		Worry		Total TA		Achievement	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
Engineering Chemistry	Male	19.96	6.36	20.33	6.13	49.04	14.45	63.45	10.65
	Female	19.98	6.57	20.16	6.03	48.96	14.19	63.52	10.33
Computational Methods & Integral Calculus	Male	17.21	4.82	17.45	4.89	42.89	10.36	68.51	9.26
	Female	16.93	4.70	17.37	4.84	42.55	10.46	68.82	9.28
English	Male	15.39	4.35	16.24	4.77	38.95	10.71	69.72	8.52
	Female	15.34	3.97	16.18	4.58	40.66	10.80	70.46	8.63
Applied Computational Methods & Integral Calculus - I	Male	15.27	4.07	16.49	4.04	44.39	9.93	68.89	5.49
	Female	15.99	1.43	19.55	0.65	40.02	5.63	73.63	10.34
Engineering Drawing	Male	-	-	-	-	-	-	-	-
	Female	17.15	4.62	18.03	4.37	43.54	7.47	69.45	7.16
Differential Equations and Applications	Male	16.35	2.74	17.14	2.14	40.15	5.77	73.02	6.29
	Female	17.57	3.90	18.83	4.61	41.99	9.25	69.84	7.74
IER	Male	15.53	1.55	16.83	1.19	41.47	10.41	71.05	3.11
	Female	16.46	4.16	16.67	3.03	42.70	9.36	66.90	5.58

On total test anxiety scale scores, the mean value is ranging between a minimum of 38.81 for male students of department of English to a maximum mean score of 49.14 for male students of department

of Engineering Chemistry. With respect to students achievement scores, male students of department of Engineering Chemistry are at lowest level (mean=63.52) to a maximum of 73.62 for female students of Institute of Applied Computational Methods & Integral Calculus - I.

Table 4: Relationship between students' achievement scores and scores on test anxiety scale (Total scale, Worry Scale and Emotional scale)

Aspect	N	Pearson r	Significance
Total scale scores and achievement scores	414	- 0.653*	0.000
Worry scale scores and achievement scores	414	- 0.694*	0.000
Emotional scale scores and achievement scores	414	- 0.663*	0.000

It is evident from table 4 that a strong negative and significant relationship exists between students' achievement scores and Total scale scores as well as on subscales scores. It is also found that achievements is significantly inversely related to both emotional and worry scales as well. The magnitude of the relationship is slightly higher on worry scale as compared to emotionality scale and total score. The range of relationship of each scale is more than 65% which is quite strong in magnitude. This stronger relationship encouraged to further analysis to explore the possibility of test anxiety as a predictor of students' achievement. Therefore, a Regression analysis was run to explore the cause- effect relationship between achievement scores and test anxiety scale scores. The result is given in table 5.

Table 5: Regression analysis

Model	β	t-value	Significance	Model R square
Total anxiety scale	-0.251	-0.6.700	0.000	
Worry scale scores	-0.697	-0 4.160	0.000	0.535
Emotional scale scores	-0.140	-0.890	0.374	

Table 5 shows that 53% of variance is explained by the regression model which shows that test anxiety affects students' achievement. It is further evident from the table that the worry scales scores are the major contributor with respect to the difference in students' achievement scores.

Table 6: Effect of gender on sub-scales of test anxiety

Multivariate Results				
Test	Value	Hypoth. df	F	Significance
Wilk' Lambda	0.874	2.000	29.567	0.000
Univariate F –Tests				
Variable		df	F	Significance
Emotional sub-scale		1	29.569	0.000
Worry sub-scale		1	0.085	0.771

It is evident from table 6 that F-value (29,567, df = 2.000, p=0.000) is significant both for multivariate test and also for univariate dimension on emotionality component on the basis of students' gender. However, for worry aspect of test anxiety scale, the difference between male and female students is not significant.

6 Conclusion and Discussion

Keeping in view the focus of the study to find the relative relationship of student achievement with affective and cognitive factors of test anxiety, the results revealed that cognitive factors (worry scale) are pivotal in generating anxiety in students more that affective (emotionality) factors. This finding was of interest as it is in line with the findings of the studies reported in literature (Chapell et al., 2005; Cassady & Johnson, 2002; Birenbaum & Nasser, 1994) and it diminishes the assumption that test anxiety is a function of the stakes involved in a test score. It was assumed that tests in Indiai higher education institutions were more structured and rigid in structure, thus causing greater test anxiety as compared to students in countries where exams/test formats are relatively flexible. The students feel equally anxious with every test they are asked to take.

Moreover, it is reiterated through these results that pressure of scoring high on tests, fear of passing a course, consequences of failing in test and incompatibility of preparation for test and demand of test were the reason for cognitive text anxiety. This showed the complexity of thinking process student go through while preparing for tests. This increases as they think more into the consequences or implication related to the achievement in tests. Worrying about a test cannot be regarded as negative phenomenon as a certain level of anxiety contributes positively in successful performance of a test but it accumulates into a negative force when student enters into a cyclic, non-productive process of speculating outcomes based on consequences of the test scores. It is possible to guide students to avoid getting indulged into thinking cycle letting anxiety take over their actions. Teachers, parents and peers can be considerable help for students to keep them motivated to perform better without unnecessarily letting the anticipated consequences of failure taking over the positive force bringing performance of student compatible with their abilities and skills.

Although cognitive aspects are seen as greater reason of text anxiety but emotional (affective) factors also contribute reasonably. The feeling student experience on or before the text also make him/her anxious. As students have reported that they feel uneasy, upset, nervous, tense and panic. These feelings arise irrespective of the extent of preparation of examination on the part of the student; therefore, can be assumed as not specific to tests, but anxiety we all experience during any unseen endeavour of life we go through. Students can be trained to minimize affective test anxiety by providing opportunities to handle unforeseen problem situations and letting them experience test situation more often.

It is evident that feelings (affective) and worry (cognitive) related anxiety are sources of drop in student achievement. Student achievement can be improved by training/educating students about handling stress situations in academic life. If students can manage their emotional anxiety it can assist in improved achievement. Academic programmes in institution of higher education should also focus on grooming students in skills to stabilize their emotional response to potentially difficult situations like tests. The faculty can benefit from popularly used techniques to handle both cognitive and emotional anxiety among students.

Erbe (2007), Berk & Nanda (2006), Stober (2004), Haris & Coy (2003), Foster, Paulk, & Dastoor (1999), Kondo (1996), and Serok (1991) discussed various measures and strategies which can be applied by faculty members to reduce test anxiety among their students. The strategies which can be contextually relevant and useful for teachers in India can be; task orientation and preparation, positive thinking, seeking social

support, avoidance, relaxation training, coaching/ guided imagery, self-instructional training, establishing purpose, affirmation, modalities, positive Anchors, mental simulations, use of humour, preparation of cheat sheet and study skills training.

To summarize this discussion, it is concluded that we live in a test- taking society and that when students are anxious before and during tests, test anxiety has a significant and effective impact on their performance. To effectively manage test anxiety, students can be helped by teachers, parents and educational administrators through use of cognitive, affective and behavioral strategies. It is further suggested that the students should be fully informed by the faculty and administration of departments about the nature of courses, duration of the semester, and level of commitment necessary for the successful completion of the course. The students with higher test anxiety must be identified and treated in order to increase their academic achievement.

ACKNOWLEDGEMENT

We would like to thank the University Grants Commission (UGC), Government of India, for its full support and funding to conduct this research study.

REFERENCES

[1] Albero, P; Brown, A; Eliason, S; & Wind, J. (1997): Improving Reading through the use of Multiple Intelligences. *Master's Action Research. Project*, Saint Xavier University & IRI Skylight. U. S.

[2] Bandalos, D.L., Yates, K., & Thorndike-Christ, T. (1995). Effect of math self-concept, perceived self-efficacy, and attribution for failure and success on test anxiety. *Journal of Educational Psychology, 11*, 351- 360.

[3] Berk, R.A. & Nanda, J. (2006). A randomized trial of humor effects on test anxiety and test performance. *Humor: International Journal of Human Research*, 19 (4), 425-454.

[4] Berk, R.A. (2000). Does humor in course tests reduce anxiety and improve performance? *College Teaching, 48* (4), 151-58.

[5] Birenbaum, M., & Nasser, F. (1994). On the relationship between test anxiety and test performance. *Measurement and Evaluation in Counselling and Development, 27*(1), 293-301.

[6] Cassady, J.C., & Johnson, R.E. (2002). Cognitive test anxiety and academic performance. *Contemporary Educational Psychology, 27*, 270-295.

[7] Chapell, M.S., Blanding, Z.B., Takahashi, M., Silverstein, M.E., Newman, B., Gubi, A., & Mccann, N. (2005). Test anxiety and academic performance in undergraduate and graduate students. *Journal of Educational Psychology, 97* (2), 268-274.

[8] Depreeuw, E. A. M. (1984). A profile of test anxious student. *Applied Psychology,* 33(2), 221-232.

[9] Erbe, B. (2007). Reducing test anxiety while increasing learning. *College Teaching, 55* (3), 96-97.

[10] Foster, S. K., Paulk, A., & Dastoor, B. R. (1999). Can we really teach test- taking skills? *New Horizons in Adult Education, 13* (1), 4-12.

[11] Gaudry, E., & Spielberger, C. D. (1971). *Anxiety and educational achievement*. New York: Wiley.

[12] Hancock, D. R. (2001). Effect of test anxiety and evaluative threats on students' achievement and motivation. *The Journal of Educational Research, 94* (5), 284-290.

[13] Haris, H. L., & Coy, D. R. (2003). Helping students cope with test anxiety. *ERIC Digest.* (ERIC Document Reproduction Service No. ED 479355). Retrieved June 25, 20008 from a World Wide Web: http://www.ericdigest.org/2005-2/anxity.html

[14] Hill, K.T., & Wigfield, A. (1984). Test anxiety: A major educational problem and what can be done about it. *Elementary School Journal, 85*, 105-126.

[15] Humbree, R. (1988). Correlates, causes, effects, and treatment of test anxiety. *Review of Educational Research, 58* (1), 47-77.

[16] Khalid, R., & Hasan, S. S. (2009). Test anxiety in high and low achievers. *India Journal of Psychological Research, 24*(3-4).

[17] Kondo, D. S. (1996). Strategies for coping with test anxiety. *Anxiety, Stress and Coping, 10*, 203-215.

[18] Morris, L. W., Davis, M. A., & Hutchings, C. J. (1981). Cognitive and emotional components of anxiety: Literature review and a revised worry-emotionality scale. *Journal of Educational Psychology, 73*, 541- 555.

[19] Mwamwenda, T. S. (1994). Gender differences in scores on test anxiety and academic achievement among South African University graduate students. South African Journal of Psychology, 24 (4),

[20] Nicholson, A. M. (2009). Effects of test anxiety on student achievement (ACT) for college bound students. *Dissertation Abstract International*. DAI-A-70/07, AAT 3366126

[21] Oludipe, B. (2009). Influence of test anxiety on performance levels on numerical tasks of secondary school Computational Methods & Integral Calculus students: *Academic Leadership: Online Journal, 7* (4)

[22] Orpen, C. (1996). The interactive effects of social support and test anxiety on students academic performance. *College Student Journal, 116*, (30), 464-465.

[23] Pintrich, P.R., & Schunk, D. (1996). *Motivation in education: Theory, research, and applications*. Upper Saddle River, NJ: Erlbaum.

[24] Schonwetter, d. J. (1995). An empirical investigation of effective college teaching behaviours and students difference: Lecture organization and test anxiety. Paper presented at the annual meeting 1 *American Educational Research Association* (San Franscisco) Canada.

[25] Serok, S. (1991). The application of Gestalt methods for the reduction of test anxiety in students. *Assessment and Evaluation in Higher Education, 16* (2), 157-64.

[26] Smith, K. H. (2000). The self-concept and verbal academic achievement of primary and secondary teachers. Unpublished Doctoral Dissertation, University of Melbourne, Australia.

[27] Stober, J. (2004). Dimensions of test anxiety: Relations to ways of coping with pre-exam anxiety and uncertainty. *Anxiety, Stress and Coping, 17* (3), 213-226.

[28] Williams, J. F. (1991). Modeling test anxiety, self-concept and high school students' academic achievement. *Journal of Research & Development and Education*, 25, 51-57.

[29] Williams, J.E. (1994). Anxiety measurement: Construct validity and test performance. *Measurement & Evaluation in Counseling & Development, 27* (1),

[30] Zeidner, M. (1990). Does test anxiety bias scholastic aptitude test performance by gender and socio-cultural group? *Journal of Personality Assessment, 55*, 145-160.

[31] Zoller, U., & Ben-Chain, D. (1990). Gender differences in examination type, test anxiety, and academic achievement in college science: a case study. *Computer Programming, 74*(6), 597-608.