RUMOUR INFLUENCE MINIMIZATION IN SOCIAL NETWORKS USING SFLA

ENROLLMENT NO(s). - 9915103267, 9915103265, 9915103276

NAME OF STUDENTS - PRATAYA AMRIT, PIYUSH VASHISTHA, SPARSH MAJHE

NAME OF SUPERVISOR – Dr. SHIKHA MEHTA.

MAY 2019

SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

TABLE OF CONTENTS

DECLARATION

We hereby declare that this submission is our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: JIIT NOIDA

Signature :

Date: 10/05/2019

Name : Prataya Amrit, Piyush Vashistha, Sparsh Majhe

Enrollment No: 9915103267, 9915103265, 9915103276

CERTIFICATE

This is to certify that the work titled **Rumour Influence Minimization in Social Networks Using SFLA** submitted by **Prataya Amrit, Piyush Vashistha, Sparsh Majhe** in partial fulfillment for the award of degree of **B. Tech** of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Signature of Supervisor

Name of Supervisor : **Dr. Shikha Mehta**

Designation : Associate Professor

Date : 10/05/2019

ACKNOWLEDGEMENT

The completion of any inter-disciplinary project depends upon cooperation, co-ordination and combined efforts of several sources of knowledge. We are grateful to **Dr. Shikha Mehta,** her even willingness to give us valuable advice and direction whenever we approached her with a problem. We are thankful to her for providing us with immense guidance for this project.

Signature(s) of Students :

Name of the Student's : Prataya Amrit, Piyush Vashistha, Sparsh Majhe

Enrollment Number's : 9915103267, 9915103265, 9915103276

SUMMARY

A Social Network in mathematical terms is a large directed graph of nodes interconnected with each other via a large number of connecting edges. These nodes in the graph depict the users in the social network and the edges represent which users are connected to each other in the social network. Each connection between the users has a probability by which they will transfer information between them. This will also depict the propagation probability of rumour between different users.

The diffusion of rumour between various users follows the approach which was introduced by the Independent Cascade model. Here, the root node(user) which starts the rumour possesses a single opportunity to transfer the rumour to one of its adjacent nodes with a fixed success probability of transfer. Then in the next time interval each currently activated nodes get the chance to transfer the rumour to their adjacent nodes. Similarly, this process of rumour transfer continues in discrete time interval triggering a cascade of activated nodes.

The Independent Cascade model describes the process of diffusion of rumour in a social network but fails to justify the basis on which the adjacent node of a previously actuated node gets activated. This justification is done by the Ising model. Ising model determines the success of rumour propagation with the help of following two aspects, firstly the intensity of activation of node *u*with the rumour which will determine the probability that *u*will spread the rumour, and, secondly, the tendency of the adjacent node *v*accepting the rumour. With the help of these two steps we can claim that a node *v*is activated by a node *u*.

The rumour spread generally includes 3 phase and is given by chi squared distribution. It leaves us with an explaination of a social phenomenon of the rumour spread. When the rumourspreads , it creates an environment generally known as rumour environment. All the users which comes in contact with this environment, affects the judgement and the thinking perspective of them and the people on the social network. There is a phase transition which involves both short ranged and long range interaction. The short ranged for nearest neighbours. And this gives us a cooperative result.

The rumour is assumed to spread for a time $t_0$in the network, after which the set of activated and inactive nodes are formulated. Then the survival function is used to calculate the probability of nodes that will survive after a given time t. Also, a cumulative distribution function is used which is the complement of survival function, and calculates the probability of node getting activated before the time t. Then the differentiated for of cumulative distribution function, i.e., the probability distribution function calculates the probability of each inactive node getting activated.Now comes the use of SFLA. In SFLA we pick the node with the highest probability of spreading the rumour. The probability we will consider is the probability with which that particular node will affect the other nodes with the rumour. And as soon as we find that node , which is influencing the network, we will block that node and the rumour spread is minimized.

LIST OF FIGURES

LIST OF ACRONYMS

Acronym	**Full Form**
SFLA	Shuffled Frog Leaping Algorithm
ISFLA	Improved Shuffled Frog Leaping Algorithm
OED	Orthogonal Experimental Design
TAP	Total Affinity Propagation
DRIMUX	Dynamic Rumour Influence Maximization with User Experience In Social Networks
APM	Acceptance Likelihood Maximization
IC	Independent Cascade
LT	Linear Threshold

LIST OF TABLES

Chapter 1 : INTRODUCTION

1.1 GENERAL INTRODUCTION

Regardless of the expanding utilization of internet based life stages for data and news assembling, its unmoderated nature regularly prompts the development and spread of bits of gossip, i.e., things of data that are unconfirmed at the season of posting. In the meantime, the receptiveness of online networking stages gives chances to contemplate how clients share and examine bits of gossip, and to investigate how to consequently survey their veracity, utilizing regular language handling and information mining systems. Internet based life stages are progressively being utilized as a device for get-together data about, for instance, societal issues, and to get some answers concerning the most recent advancements amid breaking news stories. This is conceivable in light of the fact that these stages empower anybody with a web associated gadget to partake progressively their musings as well as to post an update about an unfurling occasion that they might observer. Henceforth, web-based social networking has turned into a useful asset for writers yet in addition for conventional natives. Be that as it may, while internet based life gives access to an uncommon wellspring of data, the nonattendance of efficient endeavors by stages to direct presents likewise leads on the spread of falsehood, which at that point requires additional push to set up their provenance and veracity. Updates related with breaking news stories are frequently discharged piecemeal, which offers ascend to a noteworthy extent of those updates being unconfirmed at the season of posting, some of which may later be demonstrated to be false. Without a definitive explanation supporting or exposing a continuous gossip, it is seen that online networking clients will regularly share their own musings on its veracity by means of a procedure of group, between emotional sense-production that may prompt the presentation of reality behind the talk. By and by, notwithstanding this obvious vigor of online networking, its expanding propensity to offer ascent to bits of gossip propels the improvement of frameworks that, by social affair and investigating the aggregate decisions of clients, can decrease the spread of bits of gossip by quickening the sense-production process . Talk discovery framework that recognizes, in its beginning periods, postings whose veracity status is dubious, can be adequately used to caution clients that the data in them may end up being false. Similarly, gossip grouping framework that totals the advancing, aggregate decisions posted by clients can help track the veracity status of talk as it is presented to this procedure of aggregate sense-production. In this article, we present a review of the parts expected to grow such gossip minimization framework and talk about the achievement so far of the endeavors toward structure it.

1.2 PROBLEM STATEMENT

Rumour blocking is a significant issue in extensive scale informal organizations. Vindictive rumours could cause tumult in the public eye and consequently should be obstructed as quickly as time permits. Here we have come up with an approach of minimizing the rumour using Improved Shuffled Frog Leaping Algorithm.

1.3 EMPIRICAL STUDY

We have experienced our hands on Anaconda : Spyder and Jupyter Notebook, which are python based platform. We have studied the object oriented programming for our project implementation codes. Earlier in our project we have also studied about regression and classification algorithms in Machine Learning.

1.3.1 Python

Python is a deciphered, abnormal state, broadly useful programming language. Made by Guido van Rossum and first released in 1991. builds and item arranged philosophy plans to empower programming specialists to make clear, authentic code for little and substantial scale adventures. Python is continuously created and waste accumulated. It supports various programming guidelines including programming ideal models, including procedural, object-arranged, and utilitarian programming. Python is much of the time portrayed as a "batteries included" language due to its expansive standard library. Python was considered in the late 1980s as a successor to the ABC language. Python 2.0, released 2000, exhibited features like rundown understandings and a waste accumulation fit for get-together reference cycles. Python 3.0, released 2008, was a significant amendment of the language that isn't totally in reverse perfect, and much Python 2 code does not run unmodified on Python 3. As a result of stress over the proportion of code made for Python 2, support for Python 2.7 was reached out to 2020. Language architect Guido van Rossum sole obligation with respect to the endeavor until July 2018 anyway now shares his power as a person from a five-man coordinating social occasion. Python mediators are open to many working frameworks. An overall system of programming engineers makes and looks after CPython, an open source reference utilization. Anon-advantage affiliation, the Python Software Foundation, supervises Python and CPython

1.3.2 Spyder

Spyder is an unbelievable logical environment written in Python, for Python, and organized by and for scientists, masters and data inspectors. It incorporates a noteworthy mix of the propelled altering, investigation, troubleshooting and profiling usefulness of an extensive improvement apparatus with the information investigation, wise execution, significant evaluation and wonderful portrayal limits of a logical bundle and Furthermore, Spyder offers work in compromise with various noticeable logical bundles, including NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy, and that's only the tip of the iceberg. Past its numerous characteristic features, Spyder can be extended impressively, further by outsider modules. It can in like manner be used as a PyQt5 augmentation library, empowering you to expand upon its usefulness and install its parts, for instance, the intuitive support or propelled editorial manager, in your own product.

1.4 APPROACHES TO PROBLEM IN TERMS OF TECHNOLOGY /PLATFORM TO BE USED

In this project we were meant to apply Improved Shuffled Frog Leaping Algorithm for minimization of rumous in social network. The term networks and improvement are indicating the use of dataset and Python IDE, here we have used Spyder for the project. For the dataset part we have imported a text file consisting of multiple observations.

The research also demanded us to solve the problem of generations of frog for each loop and its alignment in the memeplex matrix. So we have placed the sorted frog in coloumnwise pattern in our memeplex matrix. In the minimization of rumour part, selection of most influencive node was our biggest problem. For this we have used the survival theory as followed in our research work. But dealing with all the above problem we have used the facilities of Spyder, Jupyter Notebook.

1.5 SUPPORT FOR NOVELTY

Our research work is one of its own kind, it not only minimizes the influence of rumours in the network but also proves and states our work towards the improvement of Shuffled Frog Leaping Algorithm. As SFLA is one of the nature inspired algorithms which is used in dealing many of the real world problems, by improving the basic algorithm we are also opening a gateway for more research into its improvement domain. Moreover, we have minimized the rumour using SFLA concept and survival theory.

1.6 COMPARISON OF OTHER SOLUTION TO THE PROBLEM

The solution for rumour minimization problem till date, were classic greedy algorithm, proposed greedy algorithm, dynamic blocking algorithm, minimization with user experience. In this we have built a minimization algorithm in which a matrix A is passed and the activation function is same which we have observed in our survival theory. We have used the dataset extracted from SinaWeibo with 23086 and 183549 nodes & edges respectively.

Chapter 2 : UPDATED LITERATURE REVIEW

2.1 SUMMARY OF THE PAPER STUDIED

Paper I:

Title of paper	**Detection and Minimization Influence of Rumor in Social Networks**
Authors	By Prof. Deepti Deshmukh, Kajal Hiray, Surekha Rawale, SapanaBirkure
Year of publication	2017
Summary	With the quick advancement of huge scale on-line informal organizations, on-line information sharing is getting to be inescapable day by day. Various data is proliferating through on-line informal communities correspondingly as each the positive and negative. All through this paper, we will in general will in general spotlight on the negative information issues simply like the on-line bits of gossip. Talk square likely could be a critical disadvantage in vast scale informal organizations. Noxious gossipy tidbits may make tumult in the public eye and looked for be obstructed when potential once being recognized. amid this paper, we will in general propose a model of dynamic gossip impact decrease with client ability (DRIMUX).Our objective is to curtail the impact of the talk (i.e., the quantity of clients that have acknowledged and sent the talk) by square an accurate arrangement of hubs. A dynamic Ising spread model considering each the overall quality and individual fascination of the gossip is given upheld reasonable situation. For sure, out and out totally not the same as existing issues with impact decrease, we will in general will in general require into thought the limitation of client experience utility. In particular, every hub is appointed a resilience time edge. On the off chance that the square time of every client surpasses that edge, the utility of the system will diminish. Underneath this imperative, we will in general watch out for then plan move back as a system conceptual suspected downside with survival hypothesis, and propose arrangements bolstered most likelihood rule. Trials zone unit actualized upheld substantial scale world systems and approve the viability of our approach.
Web Links	https://www.academia.edu/35273071/Detection_and_Minimization_of_Rumor_Influence_in_Social_Network

Paper II:

Title of paper	**DRIMUX: Dynamic Rumor Influence Minimization with User Experience in Social Networks**
Authors	Biao Wang, Ge Chen, Luoyi Fu, Li Song, Xinbing Wang, Xue Liu
Year of publication	2017
Summary	Gossip blocking is a difficult issue in extensive scale social networks. Malicious bits of gossip could cause turmoil in the public eye and hence should be hindered as quickly as time permits in the wake of being detected. In this paper, we propose a model of dynamic rumor influence minimization with client experience (DRIMUX).Our objective is to limit the impact of the talk (i.e., the number of clients that have acknowledged and sent the talk) by blocking a specific subset of hubs. A dynamic Ising propagation model considering both the worldwide notoriety and person fascination of the talk is exhibited dependent on realistic scenario. Moreover, not the same as existing issues of influence minimization, we consider the imperative of client experience utility. In particular, every hub is allocated a tolerance time edge. In the event that the blocking time of every client exceeds that edge, the utility of the system will diminish. Under this limitation, we at that point plan the issue as a network inference issue with survival hypothesis, and propose solutions based on most extreme probability standard. Examinations are executed dependent on expansive scale genuine world networks and approve the viability of our technique. In this paper, they research the issue of dynamic rumor impact minimization with client experience. To begin with, they consolidate the talk prevalence elements in diffusion model. We examine existing examinations on subject propagation dynamics and pick Chi square distribution to estimated the worldwide gossip notoriety. They at that point break down the individual propensity towards the gossip and present the likelihood of fruitful rumor propagation between a couple of hubs. At long last, roused by the idea in Ising model, they infer the helpful succeeding likelihood of rumor propagation that incorporates the worldwide talk notoriety with individual inclination. From that point forward, they present the idea of user experience utility capacity and dissect the effect of blocking time of hubs to the gossip spread process. We then embrace the survival hypothesis to clarify the likelihood of hubs getting initiated, and propose both voracious and dynamic algorithms dependent on greatest probability guideline.
Web Links	https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/download/11878/11666

Paper III:

Title of paper	**Detection and Resolution in Social Media : A Survey**
Authors	ArkaitzZubiaga, Ahmet Aker, KalinaBontcheva, Maria Liakata
Year of publication	2018
Summary	In spite of the expanding utilization of online life stages for data and news assembling, its unmoderated nature frequently prompts the rise and spread of bits of gossip, i.e., things of data that are unverified at the season of posting. In the meantime, the receptiveness of web based life stages gives opportunities to think about how clients share and talk about gossipy tidbits, and to investigate how to consequently survey their veracity, using regular language handling and information mining systems. In this article, we present and talk about two types of gossipy tidbits that flow via web-based networking media: long-standing bits of gossip that circle for significant lots of time, and recently rising bits of gossip brought forth amid quick paced occasions, for example, breaking news, where reports are discharged piecemeal and regularly with an unsubstantiated status in their beginning times. We give an outline of research into online networking gossipy tidbits with a definitive objective of building up talk characterization framework that consists of four segments: gossip location, talk following, talk position grouping, and rumour veracity classification. We dig into the methodologies introduced in the logical writing for the development of every one of these four components. We summarize the endeavors and accomplishments so far toward the development of talk order frameworks and close with proposals for roads for future research in social media mining for the discovery and goals of bits of gossip.
Web Links	https://dl.acm.org/citation.cfm?id=3161603

Paper IV:

Title of paper	**Distributed Rumor Blocking with Multiple Positive Cascades**
Authors	Guangmo (Amo) Tong, Student Member, IEEE, Weili Wu, Member, IEEE, and Ding-Zhu Du,
Year of publication	2018
Summary	In this paper, a P2P independent cascade model (PIC) mode, has been proposed for private social communications. In this paper we mainly focus on rumour blocking effect which includes the number of uses activated by the rumour. It is done by the agents which generates positive cascade non-cooperatively. In this paper it is also shown that how the cascade priority and activation order affect the rumour blocking problem.
Web Links	https://arxiv.org/pdf/1711.07412.pdf

Paper V:

Title of paper	**Prominent Features of Rumor Propagation in Online Social Media**
Authors	Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen
Year of publication	2016
Summary	In this paper the characteristics of rumour are identified, which helps in determining the rumours. Three aspects of diffusion are taken into consideration for characterizing the rumours: temporal, structural and linguistic. For temporal characteristics, a new periodic time series model is proposed, that considers daily and external shock cycles, where the model says that rumour may have fluctuation over time. We also identify the differences between key structural and linguistic in the spread of rumours and non-rumours.
Web Links	milab.snu.ac.kr/pub/ICDM2013.pdf

Paper VI:

Title of paper	**Efficient Influence Minimization in Social Networks**
Authors	Wei Chen, Yajun Wang, Siyu Yang
Year of publication	2009
Summary	Impact boost is the issue of finding a little subset of hubs (seed hubs) in an informal community that could expand the spread of impact. In this paper, we ponder the proficient impact boost from two correlative bearings. One is to improve the first covetous calculation of and its improvement to additionally diminish its running time, and the second is to propose new degree rebate heuristics that improves impact spread. We assess our calculations by analyses on two huge scholastic coordinated effort diagrams got from the online chronicled database arXiv.org. Our exploratory outcomes demonstrate that our improved covetous calculation accomplishes better running time contrasting and the improvement of with coordinating impact spread, our degree markdown heuristics accomplish much preferable impact spread over great degree and centrality-based heuristics, and when tuned for a particular impact course model, it accomplishes practically coordinating impact string with the avaricious calculation, and all the more critically the degree rebate heuristics run just in milliseconds while even the improved insatiable calculations keep running in hours in our test charts with a couple of a huge number of hubs. In view of our outcomes, we trust that adjusted heuristics may give genuinely adaptable answers for the impact boost issue with fulfilling impact spread and blazingly quick running time. In this manner, in opposition to what inferred by the finish of that customary heuristics are beated by the avaricious guess calculation, our outcomes shed new lights on the exploration of heuristic calculations.
Web Links	https://www.microsoft.com/enus/research/uploads/prod/.../kdd09_influence-1.pdf

Paper VII:

Title of paper	**A Fast Approximation for Influence Maximization in Large Social Network**
Authors	Jong-Ryul Lee, Chin-Wang Chun
Year of publication	2014
Summary	This paper manages a novel research work about another proficient estimate calculation for impact amplification, which was acquainted with augment the advantage of viral promoting. For productivity, we devise two different ways of abusing the 2-bounce impact spread which is the impact spread on hubs inside 2-jumps from hubs in a seed set. Initially, we propose another avaricious strategy for the impact amplification issue utilizing the 2-jump impact spread. Besides, to accelerate the new covetous technique, we devise a powerful method for expelling pointless hubs for impact boost dependent on ideal seed's nearby impact heuristics. In our tests, we assess our technique with genuine datasets, and contrast it and late existing strategies. From test results, the proposed strategy is at any rate a request of greatness quicker than the current strategies in all cases while accomplishing comparable precision
Web Links	wwwconference.org/proceedings/www2014/companion/p1157.pdf

Paper VIII:

Title of paper	**Distributed Rumor Blocking with Multiple Positive Cascades**
Authors	Jong-Ryul Lee, Chin-Wang Chun
Year of publication	2018
Summary	We in this paper define the rumour blocking game what's more, give an amusement hypothetical investigation. As per whether or on the other hand not the operators know about the gossip, we in this build up the talk mindful amusement and the gossip unaware diversion, separately. We demonstrate that the steady state (for example Nash Equilibrium) of the amusement ensures the 2-guess and 2·e−1 e-1- estimation under the situations of best-reaction and surmised reaction,separately. As confirmed by the investigations performed on real world systems, the gossip blocking diversion is compelling in constraining the spread of talk.
Web Links	https://arxiv.org/pdf/1711.07412

Paper IX:

Title of paper	**Social Influence Maximization using Genetic Algorithm with Dynamic Probabilities**
Authors	Sakshi Agarwal and Shikha Mehta (Computer Science and Information Technology, JIIT, Noida)
Year of publication	2018
Summary	The primary goal of influence maximization in the social network is to select k optimal nodes from the network, which has the highest impact in the environment. Suppose if the given graph is a directed G = (V, E, W) where V is set of vertices, E represents set of edges(u, v), W is the weight function. In contrast to cascade model, every node in the network is either active or inactive. Active node means a node v is already influenced and inactive state means node u is not influenced yet. According to the cascade theory, active nodes further activate the neighbouring nodes from inactive at time t to active at time t+1. We calculate the dynamic probability of each edge in the network, keeping that initially constant (1/n). First we calculate feature function i.e. node_influence_function g(vi, yi) = (wi * yi) / $\Sigma\, j \in NB(i)(wij + wji)$ yi ≠ i g(vi, yi) = $\Sigma\, j \in NB(i)(wji)$ / $\Sigma\, j \in NB(i)(wij + wji)$ yi = i where yi represent the node representative for node vi from set of nodes {NB(vi) U vi} with highest edge sum i.e. Edge_sum(vi) = $\Sigma\, k \in NB(i)wik$ Therefore the algorithm generates the single probability value with respect to each edge and updates the graph G(V,E,W) to G'(V,E,Wd). The updated graph is then given as input for genetic algorithm. After the completion of these steps, algorithm generates the optimal set of k seed nodes, which is maximizing the influence propagation. The work is continued for various number of generations and if the set of optimal nodes comes often then
Web Links	https://ieeexplore.ieee.org/abstract/document/8530626

Paper X:

Title of paper	Social Influence Analysis in Large-scale Networks
Authors	Jie Tang, Jimeng Sun, Chi Wang and Zi Yang
Year of Publication	2009
Summary	Now a days social media is prevalent, subtle force that grows the dynamics of all social networks. So we need such an algorithm, methods and techniques which analyse and quantify the social influence This paper mainly deals with the problem of scale of the influence on the nodes. Different types of nodes get influenced by different sources. Let us consider a real life example, a corporate worker will get more influenced by his co-workers as compared to other people, or friends will get more influenced by their group of friends as compared to other people. So different nodes (people), get different level of influence by different types of sources. We have to distinguish these level of influence. We have to define a factor which measures the level of influence. For this purpose we propose an algorithm known as TAP(topicalaffinity propagation) to model the topic-level social influence on large networks. TAP along with the efficient distributed algorithms is implemented. This is then tested under the map-reduce framework. Further in this paper the characteristics of distributed learning algorithms for map-reduce are mentioned. In this paper we measure the strength of topic level social influence quantitatively we propose the tropical factor graph(TFG) model. This model is used for analyzing the topic-based social influence quantitatively.
Web Links	http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.446.9959&rep=rep1&type=pdf

2.2 INTEGRATED SUMMARY OF THE LITERATURE CITED

In the early stages a dynamic model was proposed known as the ising propagation model. It is dynamic model which considers the global popularity as well as the how an individual gets attracted towards a rumour. It also takes real world scenarios into consideration. This model is different from existing models of minimization as it takes user experience into account which is one of the major issues in social influence minimization. User experience include a threshold time upto which a user can tolerate and can remain blocked on the social media. This is also a major issue as if the user remains block for time greater than the threshold time , the user will not use that social media further , which is also a drawback of blocking the user. But this model takes care of this issue also. Furthermore, a rumour blocking game is created. By this we a theoretical analysis is given. According to this analysis it id decided whether the users are aware of rumour. And a rumour aware game and a rumour-obvious game is created. To the best of our information, a suggestion intended to supply steerage for a client to reliably approach his friending target has not been investigated for existing on-line long range interpersonal communication administrations. to amplify the probability that the friending target would agree to a call for support from the client, we tend to figure a substitution enhancement drawback, in particular, Acceptance probability Maximization (APM), and build up a polynomial time rule, known as Selective welcome with Tree and In-Node Aggregation (SITINA), to search out the best goals. we tend to actualize a full of life friending administration with SITINA on Facebook to approve our arrangement. Our client ponder and exploratory outcomes uncover that SITINA outperforms manual decision and in this manner the benchmark approach in goals quality with effectiveness.

Extending our work , four pernicious applications are created, and assessed Andromaly capacity to identify new malware dependent on tests of known malware. They assessed a few mixes of inconsistency location calculations, include choice technique and the quantity of top highlights so as to discover the mix that yields the best execution in distinguishing new malware on Android. Observational outcomes recommend that the proposed structure is viable in distinguishing malware on cell phones when all is said in done and on Android specifically. In another paper we studied the practical impact amplification from 2 correlative bearings. One is to upgrade the principal avaricious equation and its improvement to more scale back its timeframe, and furthermore the second is to propose new degree rebate heuristics that improves impact unfurl. we tend to gauge our calculations by tests on 2 monster instructive coordinated effort charts.

Two or three new affordable guess algorithmic program for impact amplification, that was acquainted with expand the beneficial thing about irresistible operator advancing. For strength, we will in general devise 2 of abusing the 2-jump impact unfurl which is that the impact unfurl on hubs inside 2-bounces expelled from hubs in a very seed set. Right off the bat, we will in general propose a spic and span covetous approach for the impact augmentation downside abuse the 2-bounce impact unfurl. Furthermore, to rush up the new insatiable system, we will in general devise a decent way of evacuating uncalled-for hubs for impact augmentation Based on ideal seed's local impact heuristics.

Chapter 3 : ANALYSIS, DESIGN AND MODELING

3.1 OVERALL DESCRIPTION OF THE PAPER

3.1.1 Social Network:

A social network, in mathematical terms, can be defined as a directed graph G = (V,E) comprising of a set of nodes V representing the users, and a set of directed edges E indicating the connection between users (for example following or being pursued). Let |V | = N denote the number of nodes, and (u, v) ∈ E signify the directed edge from node u to node v (u, v ∈ V), and $\alpha_{uv} \in \{0, 1\}$ indicate the edge coefficient, where $\alpha_{uv} = 1$ represents to the presence of edge (u, v), and $\alpha_{uv} = 0$, otherwise. We use p_{uv} to denote the likelihood of u sending the rumour to v and v accepting it, i.e., the achievement likelihood of u activating v. Here, D(u) indicates the in-degree of node u.

3.1.2 Diffusion Model:

Diffusion models depict the process of information propagation through the social network. Two exemplary diffusion models are Linear Threshold (LT) and Independent Cascade (IC).
In Linear Threshold model, an inert node u becomes initiated if the proportion of its actuated parent nodes outperforms a certain predefined threshold $0 < \theta < 1$. In spite of the fact that it demonstrates an inactive tendency of a node getting to be actuated when its neighbors do, the LT model neglects to think about the individual contrast of every node in a social network. Consequently, we embrace the more complex Independent Cascade model in our work.

The Independent Cascade model has been generally embraced in information propagation issues. The entire diffusion process continues in discrete time steps t_0, t_1, t_2,... .At first, the cascade is activated by a set of enacted nodes, i.e., the seed nodes at t_0. In our rumour diffusion context, we expect the rumour is begun by one root node s in the system, and the other nodes are latent. We use su(t) ∈ {0,1}to signify the condition of node u at time step t, where $s_u(t) = 1$ depicts u is enacted and $s_u(t) = 0$, latent. For each subsequent time step $t \geq 1$, every node u which was enacted at time step $(t - 1)$ has a solitary chance to initiate any of its presently idle adjacent nodes with a success probability p_{uv}. In our unique circumstance, it implies in each time step, any node that has acknowledged the rumour in past time step gets an opportunity to give their idle adjacent nodes a chance to acknowledge the rumour. For effortlessness, we expect that once a node is enacted by the rumour, it will remain actuated until the finish of the diffusion process.

3.1.3 Ising Model for Dynamic Rumor Propagation

(Kempe, Kleinberg, and Tardos 2003) considered the achievement likelihood puv as a framework parameter and is fixed at the absolute initial start of the course. Be that as it may, in light of the subject elements we talked about in a past area, at various time ventures of the propagation procedure, a theme can change significantly in its ubiquity. In addition, the rumor attraction (Han et al. 2014) for every individual node u ∈Vis additionally a reasonable factor we should consider. That implies the achievement of rumor propagation between neighbors incorporates two perspectives: first, the initiated hub u must be so pulled in by the rumour that it will send the rumour to its neighbors; second, one of u's latent neighbors chooses to acknowledge the rumour. Simply after those two stages, would we be able to guarantee that v is initiated. At the end of the day, the accomplishment of rumor propagation depends both on the worldwide fame and the individual propensity of the rumour topic, which can be viewed as a summed-up highlight of the Ising model.

Presently we explore the two stages of a successful rumor propagation. In the initial step, at any time stamp t_j,u is one of the actuated nodes in time stamp (t_j−1). In view of the work in (Han et al. 2014), we give the modified variant of the likelihood of node u sending the rumour to one of its dormant neighbors as

$$p_u^{\text{send}}(t_j) = \frac{p_0}{\lg(10+t_j)}$$

where p_o is the underlying sending likelihood at time stamp 0. The likelihood of node v accepting the rumour is likewise given as p_v^{acc}= 1/D_v where D_v is the in-degree of node v. Along these lines, we give the likelihood of successful rumor propagation from u to v as

$$p_{\text{ind}}(t_j) = p_u^{\text{send}} \cdot p_v^{\text{acc}} = \frac{1}{\mathcal{D}_v} \frac{p_0}{\lg(10+t_j)},$$

which can be characterized as the individual propensity between various pair of nodes in the system.

Presently we talk about the worldwide subject prevalence of the rumour. As referenced, the rumour fame, for the most part incorporates three stages and roughly subject to the chi-square distribution, which is given by

$$p_{\text{glb}}(t;k) = \frac{2^{(1-\frac{k}{2})} t^{k-1} e^{-\frac{t^2}{2}}}{\Gamma(\frac{k}{2})},$$

where k >0 speaks to the level of opportunity, Γ(・) is the Gamma function. It clarifies a typical social wonder that when rumour spreads for some time, it might make "rumour atmosphere" that could influence the decisions or choices of users on online social networks.

As indicated by the Ising model (Chelkak and Smirnov 2012), the "stage change" of a turn includes both short-go collaborations with its closest neighbors and long-haul framework advancement and is an agreeable outcome. Motivated by that, we propose the agreeable spread likelihood coordinating $p_{glb}(t; k)$ with $p_{ind}(t)$ as

$$\begin{aligned} p_{uv}(t) =& \beta_1 \cdot p_{\text{glb}}(t; k) + \beta_2 \cdot p_{\text{ind}}(t) \\ =& \beta_1 \frac{2^{(1-\frac{k}{2})} t^{k-1} e^{-\frac{t^2}{2}}}{\Gamma(\frac{k}{2})} + \beta_2 \frac{1}{\mathcal{D}_v} \frac{p_0}{\lg(10 + t_j)}, \end{aligned}$$

where $\beta_1, \beta_2 \in (0, 1)$ are the parity coefficients which fulfill$\beta_1 + \beta_2 = 1$.

In light of this agreeable proliferation likelihood, the likelihood of node v getting enacted at time stamp t_j can be given by:

$$Pr[s_v(t_j) = 1] = 1 - \prod_{u \in \mathbb{P}_v} [1 - s_u(t_{j-1}) p_{uv}(t_j)],$$

where $P_r[\cdot]$ speaks to likelihood, and P_v speaks to the parent nodes of v.

3.1.4 Survival Theory

In this model, we assume that the rumour has spread for quite a while, and it is distinguished at time t_0 by the framework. It is likewise expected that by time t_0, there have just been an absolute number of N_1 actuated nodes, and $N_2 = N - N_1$ nodes are stay inert. Let V_{N1} and V_{N2} indicate the arrangement of actuated and inert nodes at time t_0 separately. In this manner, from t_0 onwards, the framework can be seen as N_1 independent spreading through the system, and our objective is to choose K nodes and block them with the goal that the last number of actuated nodes amid the perception time window T can be limited. Let $C = (c_1,...,c_{N1})$ indicate the arrangement of cascades activated by N_1 enacted nodes by time t_0. A course $c_i \in C$ can be spoken to by a N-dimensional time vector $t^{ci} = (t^{ci}_1, ..., t^{ci}_{N2})$ where $t^{ci}_j \in [t_0, t_0 + T] \cup \{\infty\}$, j =1, 2,...,$N_2$ is the actuated time of node j in course c_i. The perception time window T is chosen by the client experience utilityconstraint, and ∞ implies the node isn't initiated until the finish of the perception time $(t_0 + T)$.

Here we consider just a single course and the outcomes can be stretched out to different falls. Characterize $\alpha_v(t|s(t))$ as the risk rate of node v molded on the arrangement of nodes initiated by time t. Our objective currently is to investigate the effect of the peril rate of various nodes to the gossip impact minimization issue.

3.1.5 Survival Function

To start with, we present the survival work characterized as (Aalen, Borgan, and Gjessing 2008)

$$S(t) = Pr(t < T),$$

where T is the event time of an occasion of intrigue, t is some predefined time. The survival work speaks to the likelihood that the occasion of intrigue happens after the perception "due date". On the off chance that we utilize the wording "passing" to speak to the event of the occasion, we can guarantee that the objective "endures". At that point we have the aggregate circulation work F(t):

$$F(t) = Pr(T \leq t) = 1 - S(t).$$

In like manner, the likelihood thickness work f(t) is given by

$$f(t) = \frac{d}{dt}F(t).$$

3.1.6 Hazard Rate

The Hazard rate which portrays the quick rate of event of an occasion is characterized as:

$$\begin{aligned} \alpha_v(t|\boldsymbol{s}(t)) &= \lim_{dt \to 0} \frac{Pr(t \leq T \leq t + dt | T > t)}{dt} \\ &= -\frac{S'(t)}{S(t)}, \end{aligned}$$

where S (t) is the subordinate of S(t). As needs be, we can have

$$S(t) = e^{-\int_0^t \alpha_v(\tau|\boldsymbol{s}(\tau))d\tau},$$

also, for a specific node v, we have

$$F_v(t|\boldsymbol{s}(t)) = 1 - e^{-\int_0^t \alpha_v(\tau|\boldsymbol{s}(\tau))d\tau}.$$

In light of the survival investigation, we propose an added substance survival model where the danger rate is given by

$$\alpha_v(t|\boldsymbol{s}(t)) = \boldsymbol{\alpha}_v^T \boldsymbol{s}(t) = \sum_{u:t_u<t} \alpha_{uv} p_{uv}(t),$$

where $\alpha_v = (\alpha_{uv})$, u = 1, 2,...,N is a non-negative parameter vector demonstrating the presence of the edge between node u and v. $\alpha_{uv} = 1$ if there is an edge among them; and $\alpha_{uv} = 0$, generally.

We characterize a coefficient lattice $A := [\alpha_v] \in R_+^{N\times N}$ to indicate the structure of system, and A_0 be the first system coefficient framework before any nodes are blocked. At that point we ascertain $F_v(t|s(t))$ as:

$$\begin{aligned} F_v(t|\boldsymbol{s}(t)) =& 1 - e^{-\int_{t_u}^{t} \sum_{u:t_u<t} \alpha_{uv} p_{uv}(\tau) d\tau} \\ =& 1 - e^{-\sum_{u:t_u<t} \int_{t_u}^{t} \alpha_{uv} p_{uv}(\tau) d\tau} \\ =& 1 - \prod_{u:t_u<t} e^{-\alpha_{uv} \int_{t_u}^{t} p_{uv}(\tau) d\tau}. \end{aligned}$$

Likewise, we have the probability capacity of the initiation of node v, fv(t|s(t)), as following:

$$f_v(t|\boldsymbol{s}(t)) = \sum_{u:t_u<t} \alpha_{uv} p_{uv}(t) \prod_{\varrho:t_\varrho<t} e^{-\alpha_{\varrho v} \int_{t_\varrho}^{t} p_{\varrho v}(\tau) d\tau}.$$

Given the actuation probability of a solitary inert node $v \in V_{N2}$, now we consider any number of latent nodes ina cascade. Amid the whole perception window T, $t \leq T = (t_1, \ldots, t_i, \ldots, t_N \,|t_0 \leq t_i \leq t_0 + T)$. We expect that each enactment is restrictively autonomous on actuations happening later given past initiations. At that point we can register the actuation probability as:

$$\begin{aligned} f(\mathbf{t}^{\leq T}; \mathbf{A}) = \prod_{i:t_i<T} \sum_{u:t_u<t_i} & \alpha_{uv} p_{uv}(t_i) \times \\ & \prod_{\varrho:t_\varrho<t_i} e^{-\alpha_{\varrho v} \int_{t_\varrho}^{t_i} p_{\varrho v}(\tau) d\tau}. \end{aligned}$$

In light of the initiation probability work, we plan the blocking calculations. To begin with, we select and hinder all K nodes in the meantime t_0. As is appeared, the initiation probability of a latent node v is identified with the risk rate originating from all recently actuated nodes. In this manner, the early actuated nodes assume a huge job in the whole procedure.

Henceforth, we propose the accompanying avaricious calculation to limit the impact of the rumour inside one time stamp after it is distinguished. We accept that there are Mtime steps: $t_1,...,t_M$ amid the entire perception window T, with each time step enduring T/M.

3.1.7 Original SFLA:

Beginning from exemplary SFLA, it is a meta-heuristic drew nearer for taking care of complex genuine issues, it is a nature propelled agreeable for a given populace. A populace is as set of people. Every individual has a related wellness esteem that estimates how close it Is from sustenance. SFLA comprises of a lot of frogs partitioned into certain memeplexes. The calculation depends on the advancement of images conveyed by the iterative people, and a worldwide trade of data among themselves.

We should talk about the procedure through arrangement of steps:

1. Initialize the number of inhabitants in the frogs of size of P.

2. Calculate the wellness of the considerable number of frogs of populace and sort them in climbing request of their wellness. The wellness is the benchmark for advancement i.e the lower its esteem the more streamlined it is. It is utilized to assess the situation of frog.

3. Divide the frog populace into m memeplex for each containing N frogs. Conveyance would resemble first frog goes to first memeplex, second frog goes into second memeplex, etc.

4. We play out the nearby looking inside the each memeplex. Let the frog with the best wellness is X_b and the frog with the most exceedingly terrible wellness is X_w separately. The worldwide best frog is spoken to by X_g. X_w gets refreshed by X_g as pursues:

$X_{w'} = X_w + r.(X_b-X_w)$

where 'r' is arbitrary number in range (0,1).

We need to do it under specific limitations: $|X_b – X_w| <D_{max}$, where D_{max} is the most extreme conceivable difference in frog. On the off chance that the above condition improves the most exceedingly awful frog position towards best region, at that point the wellness work is refreshed correspondingly. Else, the X_b parameter in the above condition is supplanted with the worldwide best frog (X_g) and again its wellness isn't not exactly the most exceedingly awful frog wellness at that point, X_w is supplanted by the frog which is haphazardly produced with discretionary wellness. Subsequent to finishing the neighborhood look inside the memeplex, all the populace are rearranged and the worldwide data are passed away those frogs in the rearranged procedure.

5. Local inquiry and rearranging process proceed to fill until the particular criteria is satisfied.

3.1.8 Improved SFLA using Memetic Reconfiguration

We have talked about different calculations or techniques which are improving the outcomes. A few calculations, proposed are improving the combination rate inside the memeplex. A portion of the calculations depended on the Orthogonal learning. Some have conveyed forward their work on memetic advancement, which prompts quicker intermingling by creating intelligent learning strategies, in which the most exceedingly awful frog position improved utilizing the frogs past encounters in each measurement. One of the methods depended on refreshing the nearby inquiry venture by allotting the estimation of arbitrary in the scope of (0,2). This implies the refreshed most exceedingly terrible frog wellness can be superior to anything the best frog wellness, therefore staying away from the union snare. Out of all these, we can see that none of them have done their work on reconfiguration of memeplex. How about we presently expand our musings on what do we mean by reconfiguration of memeplexes. Expecting there are (m=5) memeplexes and populace estimate (P=25). Presently how about we comprehend the circulation of the frogs in great SFLA calculation with figure appeared as follows:

F1	F2	F3	F4	F5
F6	F7	F8	F9	F10
F11	F12	F13	F14	F15
F16	F17	F18	F19	F20
F21	F22	F23	F24	F25

Figure 1: Classic SFLA (row wise distribution)

In our algorithm we are changing the order of frogs as shown below:

F1	F6	F11	F16	F21
F2	F7	F12	F17	F22
F3	F8	F13	F18	F23
F3	F9	F14	F19	F24
F5	F10	F15	F20	F25

Figure: 2 ISFLA (column wise distribution)

Extending our work on further level, now we are distributing the frogs in memeplex randomly as shown in figure below:

F1	F10	F3	F18	F6
F13	F2	F9	F4	F21
F24	F15	F22	F16	F25
F17	F23	F20	F5	F14
F12	F11	F8	F19	F7

Figure 3: ISFLA (random distribution)

Finally, we would check the test aftereffects of the proposed calculation with differentiation to great SFLA. We will think about most noticeably awful frog, best frog, middle, standard deviation and mean for the end criteria appointed.

3.2 REQUIREMENT ANALYSIS

3.2.1 SOFTWARE REQUIREMENTS

- Windows 7 or higher
- Spyder (IDE)
- Jupyter Notebook
- Eclipse (IDE)

3.2.2 LANGUAGE REQUIREMENTS

- Python
- Java

3.2.3 HARDWARE REQUIREMENTS

A machine with minimum 8 GB RAM, minimum 250 GB disk space, minimum 1.8 GHz processor.

3.2.4 FUCTIONAL REQUIREMENTS

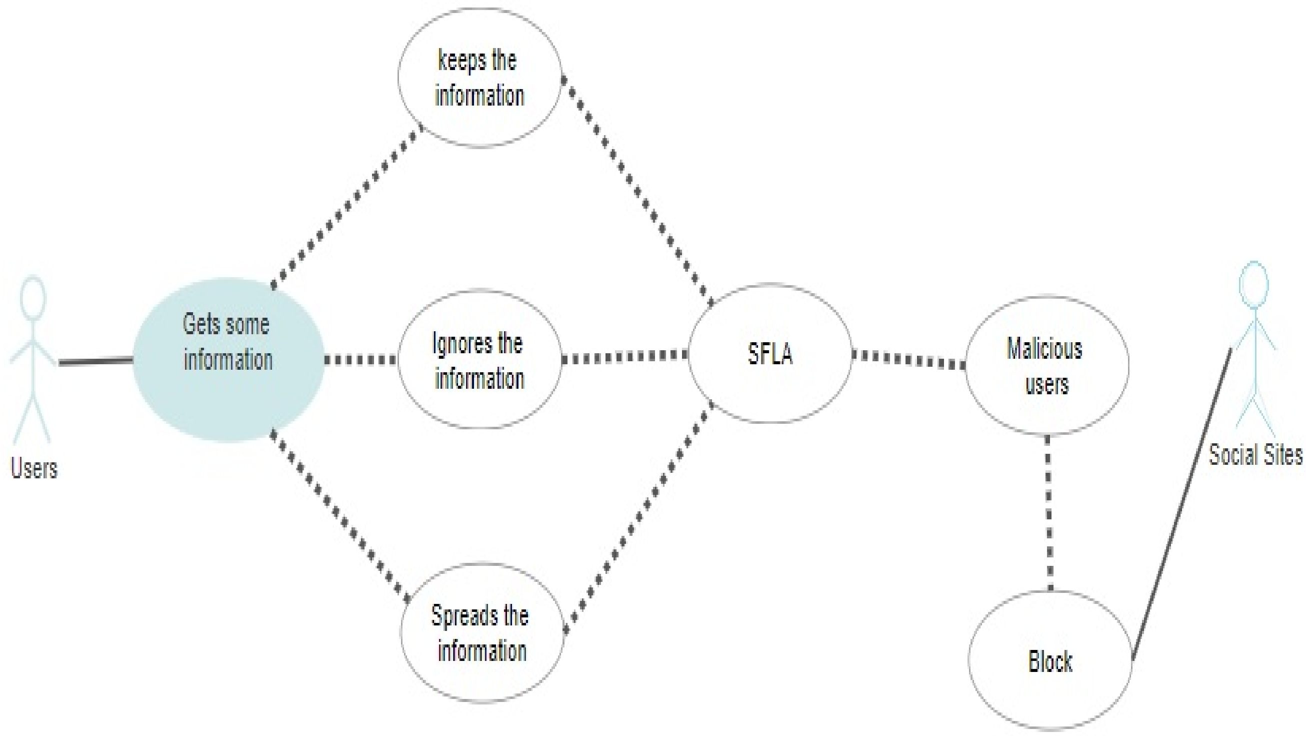
The dataset used for rumour minimization and dynamic probability generation in case of SFLA.

3.2.5 NON FUNCTIONAL REQUIREMENTS

- The system must be easy to use. The system should be user friendly because anyone can be master instead of the person who did the code.
- System will apply an appropriate algorithm of SFLA to generate the best and worst frog, also the survival theory for rumour minimization.

3.3 DESIGN DIAGRAM

3.3.1 USE CASE DIAGRAM



Fifure 2: Use Case Diagram

3.3.2 CLASS DIAGRAM

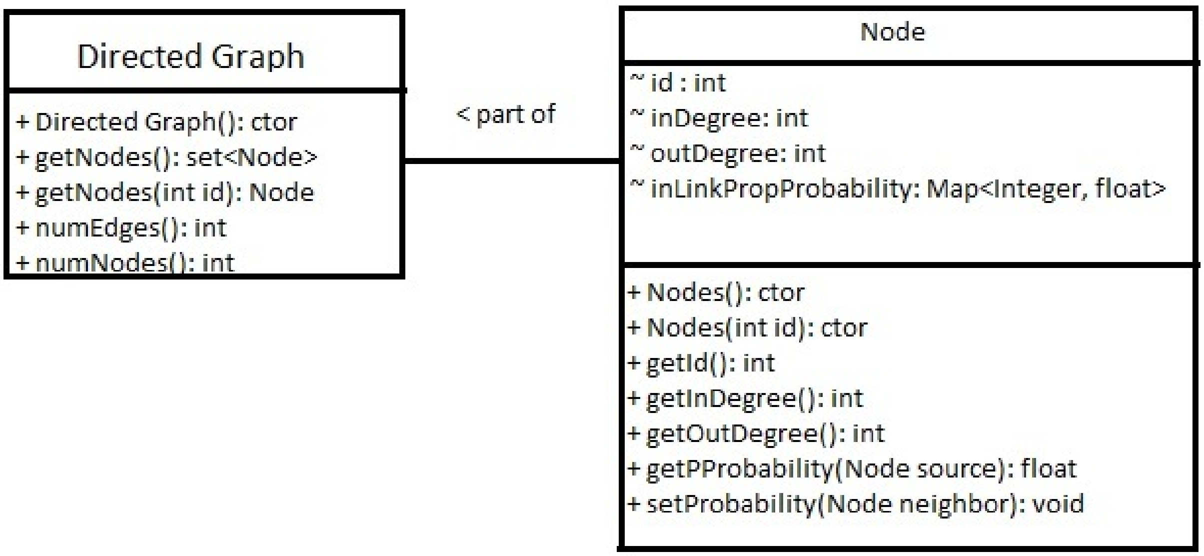


Figure 3: Class diagram of Node and Directed Graph

3.3.3 ACTIVITY DIAGRAM

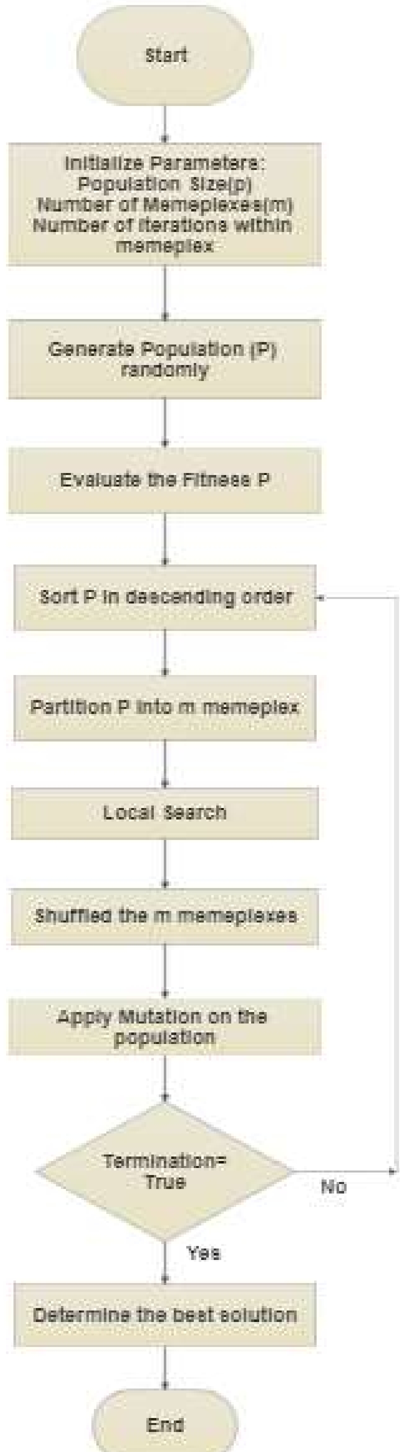


Figure 4: Flowchart of SFLA

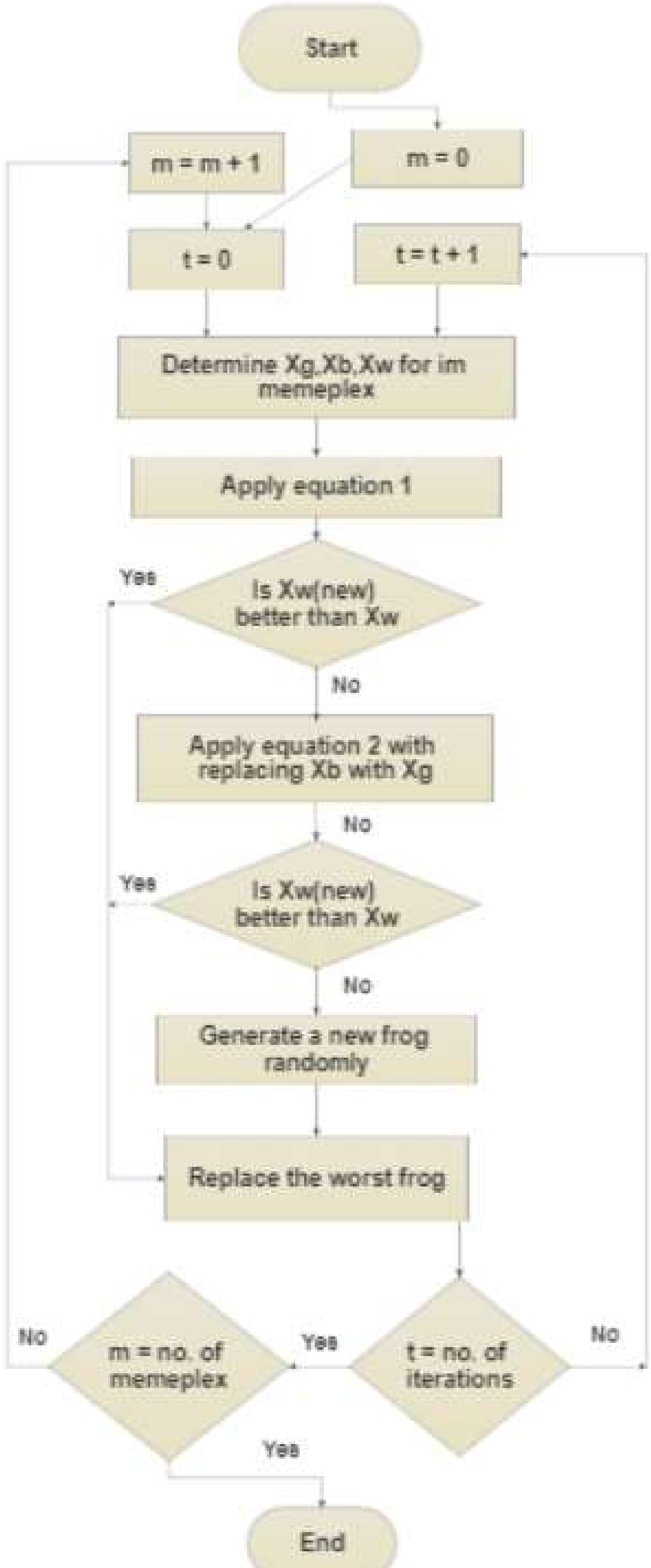


Figure 5: Flow chart of local search

Chapter 4 : IMPLEMENTATION DETAIILS AND ISSUES

4.1 IMPLEMENTATION DETAILS

Part -1 :

```
import numpy as np
import matplotlib.pyplot as plt
import random
import pandas
population = 50
memplex = 5
variables = 10
upperlimit = 100
lowerlimit = -100
Dmax = 100
total = 0
global_best = 0
n = int(population / memplex)
miteration = 8
bestfrog = []
worstfrog = []
fitness = [0 for i in range(population)]
# calculating the fitness value
def fitness_fn(frogs):
global fitness
for i in range(population):
total = 0
for j in range(variables):
total = total + frogs[i][j] * frogs[i][j]
fitness[i] = total
return fitness
def fit(check):
total = 0
for j in range(variables):
total = total + check[j] * check[j]
return total
def run(dimensions):
# generating random frogs
frogs = [[0 for i in range(variables)] for j in range(population)]
for i in range(population):
for j in range(variables):
frogs[i][j] = (random.random() * upperlimit) + (random.random() * lowerlimit)
if frogs[i][j] < 0:
frogs[i][j] = frogs[i][j] + 100
```

```
for z in range(population):
fitness[z] = fit(frogs[z])
# arranging in ascending order
for j in range(population - 1):
for k in range(j+1,population):
if fitness[j] > fitness[k]:
temp = fitness[j]
fitness[j] = fitness[k]
fitness[k] = temp
temp2 = frogs[j]
frogs[j] = frogs[k]
frogs[k] = temp2
# assigning the global best
global_best = fitness[0]
# creating the memplexes
pop = 0
memplexes = [[[0 for i in range(variables)] for j in range(n)] for k in range(memplex)]
for z in range(10000):
for i in range(memplex):
for j in range(n):
for k in range(variables):
memplexes[i][j][k] = frogs[pop][k]
pop += 1
for i in range(memplex):
print ("memplex", i)
for j in range(miteration):
bestfrog = memplexes[i][0]
worstfrog = memplexes[i][n-1]
fw = 0
fw = fit(worstfrog)
for t in range(variables):
di = random.random()*(bestfrog[t]-worstfrog[t])
if(di<-Dmax):
di=Dmax
if(di>Dmax):
di=Dmax
worstfrog[t]=worstfrog[t]+di
fn = 0
fn = fit(worstfrog)
print("old worst ",fw," new worst ",fn)
if(fn > fw):
bestfrog=frogs[0]
for u in range(variables):
di = random.random()*(bestfrog[u]-worstfrog[u])
if(di<-Dmax):
di=Dmax
```

```
if(di>Dmax):
di=Dmax
worstfrog[u]=worstfrog[u]+di
fn = fit(worstfrog)
print("old worst1 ",fw," new worst1 ",fn)
if(fn < fw):
memplexes[i][n-1] = worstfrog
fitness[i*5]=fn
for jj in range(population - 1):
for kk in range(j+1,population):
if fitness[jj] > fitness[kk]:
temp = fitness[jj]
fitness[jj] = fitness[kk]
fitness[kk] = temp
elif(fn > fw):
for v in range(variables):
worstfrog[v] = random.random() * upperlimit + random.random() * lowerlimit
memplexes[i][n-1] = worstfrog
fn = fit(worstfrog)
print("old worst2 ",fw," new worst2 ",fn)
fitness[i*5]=fn
elif(fn < fw):
memplexes[i][n-1] = worstfrog
fitness[i*5]=fn
for jj in range(population - 1):
for kk in range(j+1,population):
if fitness[jj] > fitness[kk]:
temp = fitness[jj]
fitness[jj] = fitness[kk]
fitness[kk] = temp
y = fitness[0]
print(y)
if __name__ == "__main__":
for d in range(0, 21):
run(d)
```

Part – 2 :

Fitness Function = (Active in Edges/ Total In Edges) * (Active Out Edges/Total out Edges)

- We would take a memeplex matrix in which the column will be denoting the number of nodes in the network and row would be denoting the activated nodes connected with it respectively.
- Now replace the activated nodes with the probabilities generated from the activation function. Now we will sort the matrix in decreasing order just like we were doing it in SFLA.
- The value coming at the first place in the memplex row of each nodes will be the node which needs to be deleted or blocked from the network.
- According to Cascade Model each activated node can activate one of its neighboring nodes in one time stamp.
- The nodes having none of the activated nodes in the network will have zero value in the memplex, and will stay unactivated.
- So each of the activated nodes connected to its parent node would be deleted in every time step one by one .
- The process continues till each of the activated nodes would get blocked and the network would become rumour free.

4.2 ALGORITHMS

Our implementation consists of approaching different algorithms, so here we are dividing our implementation work algo into three categories namely: Rumour Propagation, Rumour Survival and Rumour Minimization.

4.2.1 RUMOUR PROPAGATION:

1. **INDEPENDENT CASCADE**

- The IC model has been broadly embraced in data dispersion issues.
- The entire spread procedure continues in discrete time steps t0, t1, t2,....
- At first, the course is activated by a lot of initiated nodes, i.e., the seed nodes at t0.
- In our talk dispersion setting, we expect the rumour is begun by one source node s in the system, and different nodes are idle.
- We use $s_u(t) \in \{0, 1\}$ to signify the condition of node u at time step t, where $s_u(t)=1$ speaks to u is actuated and $s_u(t)=0$, something else.
- For each after time step $t \geq 1$, every node u which was initiated at time step $(t - 1)$ has a solitary chance to actuate any of its as of now inert neighbours v with a triumph likelihood p_{uv}.
- In our specific situation, it implies in each time step, any hub that has acknowledged the gossip in past time step gets an opportunity to give their dormant neighbours a chance to acknowledge the talk.
- For straightforwardness, we expect that once a node is initiated by the rumour, it will remain actuated until the finish of the dispersion procedure.

2. **ISING MODEL**

- Consider the achievement likelihood p_{uv} as a framework parameter and is fixed at the absolute initially start of the course.
- The accomplishment of talk proliferation between neighbours incorporates two viewpoints: first, the actuated node u must be so pulled in by the rumour that it will send the rumour to its neighbours; second, one of u's dormant neighbours v chooses to acknowledge the rumour.
- Simply after those two stages, would we be able to guarantee that v is activated. As it were, the achievement of rumour engendering depends both on the worldwide notoriety also, the individual inclination of the rumour subject, which can be viewed as a summed up highlight of the Ising model.

  Now we discuss the two steps which leads to the successful rumour propagation:

  1. Let us consider a time stamp t_j , and let us assume u as the activated node in the time stamp t_j-1. Now, u has a tendency to spread the rumour to its neighbouring node v. The probability with which it will spread the rumour is $p^{send}_u(tj) = p0 / lg(10+tj)$, where p0 denotes the probablility with which it send the rumour at t0.

The probability with which v accepts the rumour is denoted by $p^{acc}_v = 1/D_v$, where $D_v$denotes the in-degree of node v. Thus, we give the probability of successful rumor propagation from u to v as $p_{ind}(tj) = p^{send}_u \cdot p^{acc}_v = (1/D_v) * (p0/ \lg(10 + tj))$

2. Presently we talk about the worldwide theme prevalence of the rumour. As referenced in related work, the rumour prevalence by and large incorporates three stages and around subject to the chi square appropriation.

4.2.2 RUMOUR SURVIVAL:

SURVIVAL THEORY

- Let us assume the rumour has spread for some time t_0 in the network.
  - Now, activated nodes = N_1 and inactive nodes = $N_2 = N - N_1$
- V_{N1} and V_{N2} denote the set of activated and inactive nodes at t_0.
- The survival function represents the probability of nodes that will remain inactive (i.e., survive) after a given time t.
- The cumulative distribution function represents the probability of nodes that will get activated before a given time t.
- Now the probability distribution function is formulated by differentiating the cumulative distribution function.
- The probability distribution function calculates the probability of activation of the nodes present in the set V_{N2}.
- Now we calculate the product of the probability of all activated nodes in V_{N1} to determine the overall activation likelihood of the nodes in V_{N2}.

4.2.3 RUMOUR MINIMIZATION:

SFLA

Firstly we generate random frogs, we randomly generate population of 50 frogs. For each individual frog in the population we calculate the fitness of each frog. Fitness of each frog is calculated using the fitness function "fn". Here in this implementation we have used X^2 as the fitness function. The calculated fitness of each frog is then stored in a fitness array.
We now sort the fitness array in ascending order. Since we are dealing with the minimization problem the frog with the least fitness value will be the best frog. So our global best frog will be the frog whose fitness will be on top in the fitness array, we store this value of fitness in Xgb. Similarly the frog with the greatest fitness value will be the worst frog i.e the last frog in the fitness array.

Now we divide the population P into m memeplexes. Each memeplex represents the group of frogs. We equally divide the frogs in memeplexes. So, each memeplex has P/m frogs in it.

Here we are dividing the frogs into memeplexes in two ways.

1. Row wise distribution of frogs.
2. Column wise distribution of frogs
3. Random distribution.

We then place the frogs in the memeplex by above mentioned distribution ways. We now find the local best and local worst frogs from the memeplex. The local best is the best frog in a memeplex and the local worst frog in the worst frog in a memeplex. The best frog in a memeplex is the frog who is at the first position in the memeplex and the worst frog is the frog who is at the last position in that memeplex.

If the fitness of the newly generated frog is less than the fitness of the local worst frog, then we replace the local worst frog with the newly generated frog. This newly generated frog uses the formula : $X_w' = X_w + \text{rand}(X_b - X_w)$

If the fitness of the newly generated frog is greater than the local worst frog then again we calculate the new position of the local worst frog with respect to the global best frog.

$$X_w' = X_w + \text{rand}(X_g - X_w)$$

If the fitness of the frog with the newly generated position (w.r.t. global best frog) is less than the fitness of the local worst frog then we replace the local worst frog with the new frog generated randomly.

$$X_w' = X_w * D_{max} + X_w * -D_{max}$$

Each time we replace the local worst frog we update the fitness array i.e. we replace the fitness of the local worst frog with the fitness of the newly generated frog.
We run this algorithm 10,000 times and we iterate this whole code 25 times. Each iteration gives us one best frog i.e. which is on the top of the fitness array. We store these 25 values in another array. After this we calculate the mean, median and mode of these 25 values.

We do this for all the mentioned distribution and then compare the results i.e. the mean, median, standard deviation, best frog and worst frog for all those distributions.

```
Begin;
Generate random population of P solutions (frogs);
For each individual i in P: calculate fitness f (i);
Sort the population P in ascending order of their fitness;
Determine the fitness of global best frog X_gb as  f_gb;
Divide P into m memeplexes;
// here we change the configuration of frogs assigning in memeplexes(as discussed in theory)
For each memeplex m
       Determine the fitness of local best( X_lb )frog as f_lb;
      Determine the fitness of local worst( X_w ) frogs as   f_w ;
     For each iteration k
       For each dimension d  in individual i
          // Improve the worst frog fitness using Eqs. (1) with respect local best frog(X_lb);
            Change in frog position (D_d) = rand () x (X_lb- X_w)   (1)        (D_max >= D_d >= -D_max)
            New Position X_w = X_w + D_d ;
           Compute fitness of  X_w ;
        // If the fitness of the new frog is less than the fitness of the local worst frog;
          If fitness improves
              New Position X_w = X_w + D_d ;
         Else
       // Improve the worst frog position using Eqs. (2) with respect global best frog(X_gb);
        Change in frog position (D_d ) = rand () x (X_gb - X_w)   (2)        (D_max >= D_d >= -D_max)
         New Position X_w = X_w + D_d ;
         Compute fitness of  X_w ;
      // If the fitness of the new frog is less than the fitness of the local worst frog;
         If fitness improves
              New Position X_w = X_w + D_d ;
   // Generate the new frog and replace the local worst frog with this random frog;
        Else
            New Position X_w = rand() * D_max  +  rand() *(- D_max );
   End;
  End;
End;
Combine the evolved memeplexes;
Sort the population P in descending order of their fitness;
Check if termination is true;
End;
End;
```

Pseudo code of Improve Shuffled Frog Leaping algorithm.

4.3 RISK ANALYSIS AND MITIGATION PLAN

Risk ID	Risk Description	Risk Area	Probability	Impact	Mitigation Plan	Contingency Plan
1	System Power Failure: If the system running the algorithm breaks down due to some hardware issue, we will lose important information. So we may need to run the full algorithm again wasting time and resources	Hardware	Low	High	-	Take some buffer time before-hand such that if system crash occurs then you have some time to restart the process
2	Icompetent performance of the algorithm. This nature inspired algorithm might not be efficient enough to recommend a product.	Algorithm	High	High	We will have to improve the algorithm or case the computation by using other techniques.	-
3	Accidental human interference – Accidental file deletion, Unauthorized users perform illegal activities	File	Moderate	High	Continue monitoring persistent changes, privileged users and backups.	Provide user Authentication on at all times, to stop any unauthorized user to enter.

Chapter 5 : TESTING

5.1 TETSING PLAN

Table 2: Testing plan

Type of test	Will the Test be performed	Explanation	Software component
Requirement testing	Yes	Spyder and Notebook working needs to be checked	Spyder, Notebook
Unit testing	Yes	All the different phases need to be evaluated	All the modules
Integration	Yes	Survival theory need to be integrated with Shuffled Frog Leaping Algorithm	All the modules
Performance	Yes	Need to check the best frog and worst frog and parallelization efficiency of rumour	Time complexity of algorithms
Stress	No	User cannot have interaction with the application’s backend	None
Compliance	No	User cannot have interaction with the application’s backend	None
Security	No	User cannot have interaction with the application’s backend	None
Load	Yes	Correctness of the rumour propagation and best & worst frog needs to be checked	None

Table 3: Test Team Details

Role	Name	Responsibility
Member	Prataya Amrit	Part of all tests
Member	Piyush Vashistha	Part of all tests
Member	Sparsh Majhe	Part of all tests

Table 4: Test Schedule

Test Activity	Start Date	End Date	Hours	Comments
Requirement Testing	19-April-2019	19-April-2019	5	Spyder and Notebook working fine
Unit Testing	22-April-2019	25-April-2019	72	All the phases working fine
Integration	19-April-2019	19-April-2019	3	Prediction on the rumour dataset is working fine
Performance	25-April-2019	27-April-2019	40	Rumour is get minimized in each of the loops.
Load	27-April-2019	27-April-2019	2	Minor load on the system but the results are working fine.

Table 5: Test Environment

Software Items	Jupyter Notebook
	Spyder
	Python
	Java
Hardware Items	
	Wifi router

5.2 COMPONENT DECOMPOSITION AND TYPE OF TETSING REQUIRED

Table 6 : Component Decomposition and Identification of Test required

S.No.	List of Various Components that require Testing	Type of Testing Required	Technique for writing test cases
1	Module 1 Understanding of the mathematics of SFLA, Survival theory and number of generations.	Required Testing	Black Box (State transition analysis)
2	Module 1 Understanding of the mathematics of SFLA, Survival theory and number of generations.	Unit Testing	Black Box (Error guessing)
3	Module 2 Rumour cluster	Integration Testing	White Box
4	Module 3 Rumour minimization with SFLA and also hazard rate	Performance Testing	Black Box (Error guessing)
5	Module 4 Blocking of rumour nodes in network using SFLA	Load Testing	Black Box
6	Complete Tool	Integration Testing	White Box
7	Complete Tool	Performance Testing	Black Box
8	Complete Tool	Load Testing	Black Box

5.3 LIST ALL TEST CASES

Table 7: Test Case 1

Test Id	T1 (Module 1): Understanding of the mathematics of SFLA, Survival theory and number of generations.
Input	None
Expected Output	Understood
Status	Pass

Table 8: Test Case 2

Test Id	T2 (Module 1): Understanding of the mathematics of SFLA, Survival theory and number of generations.
Input	Run for one generation
Expected Output	The result calculated on Spyder Concole
Status	Pass

Table 9: Test Case 3

Test Id	T3 (Module 2)Rumour cluster
Input	Initiate the cluster
Expected Output	All the nodes and edges are generated
Status	Fail

Table 10: Test Case 4

Test Id	T4 (Module 3): Rumour minimization with SFLA and also hazard rate
Input	Run for Survival Activation function
Expected Output	Result on x square is none and is compared
Status	Pass

Table 11: Test Case 5

Test Id	T5 (Module 4): Blocking of rumour nodes in network using SFLA
Input	Run for rumour cluster using benchmark function and survival function
Expected Output	Result of benchmark is known
Status	Pass

5.4 ERROR AND EXCEPTION HANDELING

Table 12: Error Exception Handling Test case

Test Case Id	Test Case	Debugging Technique
T3	To check the creation of memplex matrix consisting of rumours.	Check the memplex matrix
		Check whether the rumours are regenerated for T generations

5.5 LIMITATION OF THE SOLUTION

The solution can be optimized in the running time using better parallelization tool Google Colab. The SFLA algorithm can be developed for frogs in memplex matrix which in this paper are considered as rumours.

Chapter 6 : FINDINGS & CONCLUSIONS

6.1 FINDINGS

SNo.	Time Stamp	Activated Nodes	Inactive Nodes	Blocked Nodes
1.	T0	512	9488	0
2.	T1	512	9232	256
3.	T2	510	8978	512
4.	T3	510	8723	767
5.	T4	504	8468	1022
6.	T5	500	8226	1274

Table 13: Results after SFLA

Node	Out node 1	Out node 2	Out node 3	Out node 4	Out node 5	Out node 6	Out node 7	Out node 8	Out node 9	Out node n
0	806	1268	1571	1907	4179	7339	7647	9195	9850	-
1	1756	2723	3203	3278	3899	4093	4476	6022	8846	-
2	130	1231	2936	4358	4546	5677	5875	6991	7347	-
3	2251	2800	3240	5156	5336	8252	8955	-	-	-
4	2046	5153	6437	7390	9247	9841	-	-	-	-
5	1024	2213	4455	4723	5268	5795	7100	7303	8701	-
6	11	952	1029	2367	3411	5734	6382	8573	9718	-
7	319	1893	2261	3577	3907	4017	5337	6756	9120	-
8	1320	4659	5197	5369	6663	7411	7441	7777	8743	-
9	4291	4462	6721	7567	-	-	-	-	-	-
10	65	416	1087	1392	6288	6530	7066	8803	9938	-
11	2115	4391	6301	6542	724	7427	-	-	-	-
12	29	445	1034	3146	4665	6062	8179	8393	9229	-
13	984	994	1599	1743	2091	2936	4995	5164	7379	-
14	309	415	3833	4114	4913	5445	9300	-	-	-
15	657	1311	1922	2188	2850	2959	377	5560	8253	-
16	157	688	1709	2348	2974	3474	5741	6944	8258	-
And	-	-	-	-	-	-	-	-	-	-
So	-	-	-	-	-	-	-	-	-	-
On…	-	-	-	-	-	-	-	-	-	-

Table 14: Memetic table of rumours (understanding)

6.2 CONCLUSION

Our aim was to minimize the rumour using the SFLA algorithm. We started our work by implementing the Independent Cascade diffusion model and Ising model which tells us how the rumour propagates in the social network. Then we carried our work by implementing Survival theory, which helps us in determining the probability with which the inactive nodesin the social network will get affected by the rumour or not. The node which is affected by the rumoir is the activated node. Taking our aim inconsideration i.e minimizing the rumour, we then used SFLA i.e Shuffled Frog Leaping Algorithm to find out the node which has maximum influence in spreading the rumour. In SFLA we created a fitness arrary which contains the probability of each edge transferring the rumour. We then sor this array in descending order which will give us the node with maximum chance of getting activated. And thus we easily find out which node is to blocked next.We find out that node and blocks it. In every iteration we find one node which is spreading the rumour maximum and blocks that node. But since a node cannot be blocked forever, so after a time stamp we unblock the nodes, sothat the user dose not stop using the social media.By doing so, maximum of the nodes which had maximum influence in spreading the rumour are blocked which definitely will minimize the rumour spreading .

6.3 FUTURE WORK

In our research work we are blocking the number of nodes coming in the first position of memeplex matrix, one can carry forward their work by limiting the amount of time for blocked users. As the user might would register the complaint, asking for unblocking him. One could also plan to work on different time threshold and design different blocking strategies. One could also plan on to work on user utility based environment in which one could design a function which is checking the positive experience of users if being blocked in past.

Chapter 7 : REFERENCES

[1] Biao Wang[1], Ge Chen[1], Luoyi Fu[1], Li Song[1], Xinbing Wang[1], Xue Liu[2], "DRIMUX: Dynamic Rumor Influence Minimization with User Experience in Social Network",[1]Department of Electronic Engineering, Shanghai Jiao Tong University, China.[2]School of Computer Science, McGill University, Canada.

[2] De-Nian Yang[1] , Hui-Ju Hung[1] , Wang-Chien Lee[2] , Wei Chen[3], "Maximizing Acceptance Probability for Active Friending in Online Social Networks", [1]Academia Sinica, Taipei, Taiwan,[2]The Pennsylvania State University, State College, Pennsylvania, USA, [3]Microsoft Research Asia, Beijing, China.

[3] Ceren Budak, Divyakant Agrawal, Amr El Abbadi, "Limiting the Spread of Misinformation in Social Networks",Department of Computer Science, UCSB Santa Barbara, USA.

[4] Wei Chen[1], Yajun Wang[1], Siyu Yang[2], "Efficient Influence Maximization in Social Networks",[1]Microsoft Research Asia Beijing, China, [2]Dept. of Computer Science Tsinghua University Beijing, China.

[5] Jong-Ryul Lee, Chin-Wan Chung, "A Fast Approximation for Influence Maximization in Large Social Networks", Div. of Web Science and Technology & Dept. of Computer Science, KAIST 291 Daehak-ro, Yuseong-gu, Daejeon, Korea.

[6] Surekha Rawale[1], Sapana Birkure[1], Kajal Hiray[1], Prof. Deepti Deshmukh[2], "Detection and Minimization Influence of Rumor in Social Networks",[1]B.E. Students, Dept. of Computer Engineering, D. Y. Patil Institute of Engineering and Technology, Ambi, Pune University, Maharashtra, India,[2]Assistant Professor, Dept. of Computer Engineering, D. Y. Patil Institute of Engineering and Technology, Ambi, Pune University, Maharashtra, India.

[7] Arkaitz Zubiaga[1], Ahmet Aker[2], Kalina Bontcheva[3], Maria Liakata[4], " Detection and Resolution of rumours in Social Media : A Survey", [1]University of Warwick, UK, [2]University of Sheffield, UK and University of Duisburg-Essen, Germany, [3]University of Sheffield, UK, [4]University of Warwick and Alan Turing Institute, UK.

[8] Liang'anHuo,Chenyang Ma, "Optimal Control of Rumor Spreading Model with Consideration of Psychological Factors and Time Delay", Business School, University of Shanghai for Science and Technology, Shanghai, China.

[9] Guangmo (Amo) Tong, Student Member, IEEE, Weili Wu, Member, IEEE, and Ding-Zhu Du, "Distributed Rumor Blocking with Multiple Positive Cascades"

[10] Guangmo (Amo) Tong, Student Member, IEEE, Weili Wu, Member, IEEE, and Ding-Zhu Du, "Distributed Rumor Blocking in Social Networks: A Game Theoretical Analysis"

[11] Thai-Hoang Huynh, "A Modified Shuffled Frog Leaping Algorithm for Optimal Tuning of Multvariable PID Controllers, " University of Technology, Ho Chi Minh City, Vietnam.

[12] Wang Qiusheng, Yang Hao, Sun Xiaoyao, “A Modified Shuffled Frog Leaping Algorithm with Convergence of Update Process in Local Search,” School of Automatic Science and Electrical Engineering, Beihang University, Bejing University,Bejing,China.
[13] Cheng, Ran, and Yaochu Jin, “A Social learning particle swarm optimization algorithm for scalable optimization.” Information Sciences, 2105, pp. 43-60.
[14] Xiahong Duan, Tiaong Niu, Qi Huang, “An Improved Shuffled Frog Leaping Algorithm and It’s Application in Dynamic Emergency Vehicle Dispatching,” School of Economics and Management, North China University of Technology, Beijing , China.
[15] Vajiheh Dehdeleh, Adeleh Ebrahimi, Ali Broumand Nia, “Improved Shuffled Frog Algorithm by Using Orthogonal Experimental Design, “ Islmic Azad University, South of Tehran Branch, Tehran, Iran.