ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 10(28), DOI: 10.17485/ijst/2017/v10i28/113837, July 2017*

Recognizing and Stopping Rumors Patterns in Social Networks

A. M. Meligy, H. M. Ibrahim and M. F. Torky*

Department of Computer Science, Faculty of Science, Menoufia University, Shebeen El Koom, Egypt;
meligyali@hotmail.com, hanimir78@yahoo.com, mtorky86@gmail.com

Abstract

Objectives: In this study, a proposed Colored Petri Net Model (CPNM) is used for recognizing and stopping rumors in Social Networks (SN). **Methods/Analysis:** Detecting and blocking rumors represent an open security issue in social networks. In response to this issue, the proposed CPNM is experimentally simulated on dataset consists of 863-newsworthy tweets collected from the trending topic *#CharlieHebdo* in Twitter. The performance of CPNM is analyzed and evaluated using Precision, Recall, and Accuracy metrics. In addition, the CPNM is verified against the Reachability as a major behavior property in Petri Nets. **Findings:** The practical results disclosed a superiority of the proposed CPNM in detecting accurately rumors patterns compared with other approaches in the literature. In addition, verifying the Reachability using *Reachability Graph* proved that detecting and blocking rumors tweets are reachable states according to the firing life-cycle of tokens. **Novelty/Improvement:** Detecting rumors in social networks in more accuracy and low False Positive Rate (FPR) as well as blocking its propagation over the Social Network.

Keywords: Colored Petri Nets (CPNs), Credibility Evaluation, Reachability, Rumors, Social Networks (SN)

1. Introduction

Propagating information patterns in Social Networks (SNs) may be in the form good information (credible and accurate information) or rumors information (incredible and deceptive information). Rumors (or Symantec attack)[1] have significance consequences on the people reputation, economical organization, politicians, and security of countries since it can create confusion, deceives, and mistrust among the information receivers[2]. Treating with such a type of information attacks requires firstly recognizing rumors patterns[3–5] then working to block its propagation in the social graph. Hence, Investigating rumors detection and block is a continuing concern within SN platforms. Recently, a considerable literature has grown up around the theme of detecting misinformation in SNs. In[1] introduced a cognitive psychology-based approach for detecting misinformation in online social networks, the proposed approach depend on verifying information consistency, information coherency, the credibility of sources, and general acceptability of message in order to detect misinformation. In[6] proposed machine learning-based algorithm for filtering health information in Twitter. A novel ranking approaches[3,5,7–9] is proposed to evaluate the credibility of tweets' sources and tweets' content in the Twitter social network. Other researchers have shown an increased interest in verifying the source of misleading information in social networks such as

*Author for correspondence

ranking-based algorithm, optimization-based algorithm[10], and identifying rumor source-based algorithm[11]. K-Effectors mechanism[12] is another approach proposed to identify a set of most *k nodes* that control the current activation status of the social network.In[13] introduced a mathematical approach to limit viral propagation of misinformation in OSNs, the authors study the family of β_T^I Node Protector problems for decontaminating misinformation with good information. In[14] conducted an extensive study of the problem of limiting the propagation of misinformation in a social network, but their results proved that this problem is NP-hard. However, SNs systems still suffer from a deterministic approach for detecting and blocking rumors in the social graph. Detecting and blocking rumors problem can be modeled using Colored Petri Net (CPN) tool[15]. *Tokens* appear into several data types (i.e. colored tokens). *Places* represented as several color sets. *Enabling-Firing* rule of the set of transitions depends on additional conditions, functions and parameters. The change from one state to another is represented as a set of *marking states*.

In this paper, a novel Colored Petri Net Model (CPNM) is introduced for detecting and blocking rumors patterns across OSNs. The proposed model is experimentally simulated and evaluated on dataset consists of 863 tweets collected from Twitter. The results cleared outperforming in detecting rumors tweets compared with other mechanisms in the literature according to the metrics of Precision, Recall, and Accuracy. In addition, the Reachability analysis demonstrated that detecting, and blocking rumors tokens are reachable marking states from the initial marking in the proposed CPN model.

The rest of this paper can be organized as Section (2) discuss the proposed method, Section (3) presents the results and discussion, section (4) formulates the conclusion

2. Materials and Methods

The proposed CPNM model can perform two major functions: (1) Detecting rumors tokens based on proposed credibility evaluation algorithms, which assigned to the set of transitions.(2) Blocking the propagation of detected rumors tokens. Verifying information credibility depends on containing the shared information on a Unified Resource Locator (URL) in its content. The URL feature is a major feature can be used to evaluate the information sources. According to the modeling language in colored Petri Nets, the shared information patterns can be represented as a set of *colored token*s in the form $x_i v_i$ such that x_i is the number of occurrences of the token and v_i is the token-data type (i.e. Token Pattern).Processing information credibility can be represented as a set of *Marking States:*$R(M_0) = \{M_0, M_1, M_2, \ldots..M_n\}$. Each marking state$M_i$ describes number of tokens in all places in the form $M_i = \{P_1, P_2, P_3, \ldots..P_n\}$ with respect to the color set of each place. The algorithms used to evaluate information credibility can be represented as functions assigned to a set of *Transitions* in the form $t_1, t_2, t_3, \ldots \ldots t_n$. The change from one state to another state can be represented using set of *Arcs* which labeled with inspections in the form mathematical expressions that control in the firing process.

The methodology of the proposed CPNM can be depicted as in Figure 1.

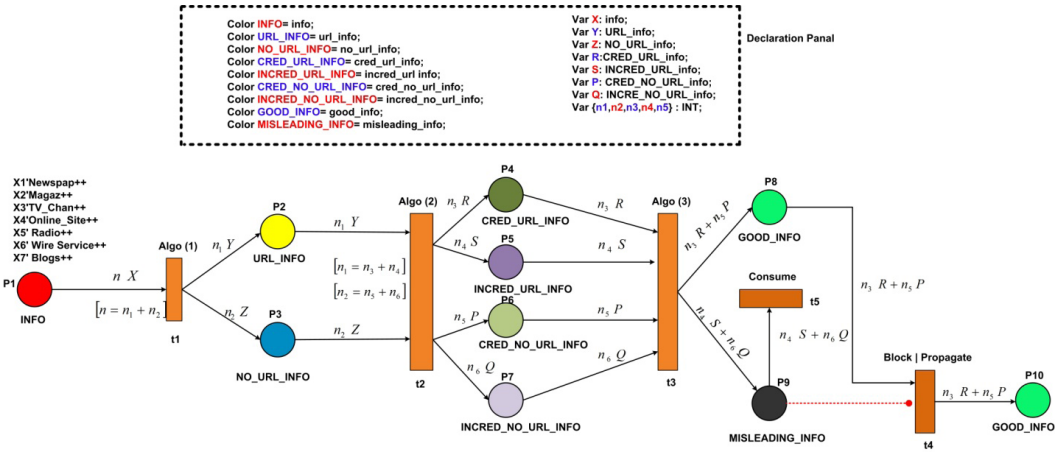

Figure 1. The Proposed CPNM Approach.

The declaration panel of the proposed CPNM involves nine color sets, the color set for each place is depicted in Table 1.

Table 1. Color sets of ten places in the proposed CPNM

Places	Color Set	Meaning
P1	INFO	Information
P2	URL_INFO	Information contain URL
P3	NO_URL_INFO	Information doesn't contain URL
P4	CRED_URL_INFO	Credible URL-information
P5	INCRED_URL_INFO	Incredible URL-Information
P6	CRED_NO_URL_INFO	Credible No-URL-Information
P7	INCRED_NO_URL_ INFO	Incredible No-URL-Information
P8	GOOD_INFO	Credible/Good Information
P9	MISLEADING_INFO	Misleading Information (Rumors)
P10	GOOD_INFO	Credible/Good Information

The initial marking M_0 in the proposed CPNM is initialized by allocating place P_1 with seven patterns of tokens, where each pattern represents a specific source of newsworthy information. The newsworthy information sources may be Newspapers, Magazines, TV channels, Online Sites, Radio, Wire Services, and Blogs. The number of tokens' occurrences for each source pattern is represented by the variables x_1, x_2, x_3, x_4, x_5, x_6, x_7 respectively. Firing the transition t_1 t_1 will classifies the input tokens (e.g. tweets) into two classes of information in the places P_2 and P_3, where P_2 is the repository of all tokens represent information contain URL in its content and P_3 is the repository of all tokens represent information doesn't contain URL in its content. The major functionality of transition t_1 can be described in **Algorithm 1**.

Algorithm 1: Information Classification (Transition t_1)

1: **Input:** Info $F = \{F_1, F_2, F_3, \ldots..F_n\}$
2: **Output**: URL_ Info $UF = \{F_1, F_2, F_3, \ldots..F_{n1}\}$
3: **Output**: No_URL_Info $NUF = \{F_1, F_2, F_3, \ldots..F_{n2}\}$
4: **Procedure** Info_Classification
5: **for** each $F_i \in F$ *Do*
6: **if**$(F_i.url\ IsTrue(\square))$ *Then*
7: $UF = UF \cup F_i$
8: **else**
9: $NUF = NUF \cup F_i$
10: **End if**
11: **End for**
12: return $UF = \{F_1, F_2, F_3, \ldots..F_{n1}\}$
13: return $NUF = \{F_1, F_2, F_3, \ldots..F_{n2}\}$
// Guard Expression Condition.
14: if$(n == n_1 + n_2)$ Then
15: return True
16: else
17: return False.
18: End Procedure

Firing t_1 depends on holding the guard expression $n = n1 + n2$, where n is the number of all input tokens in the color set of place P_1, $n1$ is the number of tokens in the color set of place P_2, and $n2$ is the number of tokens in the color set of place P_3.

The transition t_2 is responsible for evaluating the credibility of all tokens in the places P_2 and P_3. Firing t_2 produces four color sets of tokens in the places P_4, P_5, P_6 and P_7. The tokens places P_4, P_5, P_6 and P_7 represent four levels of information credibility according to the color set of each place. In addition, Firing t_2 depends on holding two guard expressions $n1 = n3 + n4$ and $2 = n5 + n6$, where, $n3$ is s the number of all tokens in the color set of place P_4, $n4$ is s the number of all tokens in the color set of place P_5, $n5$ is s the number of all tokens in the color set of place P_6, $n6$ is s the number of all tokens in the color set of place P_7 respectively. The major functionality of transition t_2 can be described in **Algorithm 2**.

Algorithm 2: Information Credibility Evaluation (Transition t_2)

1: **Input:** URL_ Info $UF = \{F_1, F_2, F_3, \ldots..F_{n1}\}$
2: **Input:** No_URL_Info $NUF = \{F_1, F_2, F_3, \ldots..F_{n2}\}$
3: **Output**: Cred_URL_info $CUF = \{F_1, F_2, F_3, \ldots..F_{n3}\}$
4: **Output:** Incred_URL_info $ICUF = \{F_1, F_2, F_3, \ldots..F_{n4}\}$

5: **Output:** Cred_NO_URL_info
$CNUF = \{F_1, F_2, F_3, \ldots..F_{n5}\}$

6: **Output:** Incred_NO_URL_info
$ICNUF = \{F_1, F_2, F_3, \ldots..F_{n6}\}$

7: **Procedure** *Info_Cred_Evaluation*

8: **for** each $F_i \in UF$ *Do*

// CredibilityEvaluation of URL info.

9: $Score(F_i) = BM25F\ (F_i.URL)$

10: $Cred_{Threshold1} = \sum_{i=1}^{i=n1} \frac{Score(F_i)}{n1}$

11: **for** each $F_i \in UF$ *Do*

12: if $(Score(F_i) \geq Cred_{Threshold})$

13: $CUF = CUF \cup F_i$

14: else

15: $ICUF = ICUF \cup F_i$

16: **for** each $F_i \in NUF$ *Do*

// CredibilityEvaluation of No-URL info.

17: $ER(F_i) = \frac{RE + RT}{FL} \times 100$

18: $Cred_{Threshold2} = \sum_{i=1}^{i=n2} \frac{ER(F_i)}{n2}$

19: **for** each $F_i \in NUF$ *Do*

20: if $(ER(F_i) \geq Cred_{Threshold2})$

21: $CNUF = CNUF \cup F_i$

22: else

23: $ICNUF = ICNUF \cup F_i$

24: **return** $CUF = \{F_1, F_2, F_3, \ldots..F_{n3}\}$

25: **return** $ICUF = \{F_1, F_2, F_3, \ldots..F_{n4}\}$

26: **return** $CNUF = \{F_1, F_2, F_3, \ldots..F_{n5}\}$

27: **return** $ICNUF = \{F_1, F_2, F_3, \ldots..F_{n6}\}$

// Guard_ Expression Condition

28: **if** [$(n1 == n_3 + n_4)$ *AND* $(n2 == n_5 + n_6)$] Then

29: **return** True

30: **else**

31**: return** False.

32: **End Procedure**

The transition t_3 is responsible for unifying two patterns of colored tokens. Firing t_3 unifies the tokens in P_4 with the tokens in P_6 and produces the unification result into places P_8 as credible information-tokens; in addition, it unifies the tokens in P_5 with the tokens in P_7 and produces the unification result into places P_9 as rumors-tokens. The major functionality of transition t_3 is depicted in the **Algorithm 3.**

Enabling or disabling transition t_4 is depending on the *inhibitor arc* from places P_9 to transition t_4. With respect to the functionality of inhibitor arc, it enables t_4 if place P_9 doesn't contain any tokens, but it disables t_4 if P_9 contains any tokens even if one. The functionality of transition t_4 is depicted in the **Algorithm 4.**

Algorithm 3:Credible/ Rumor Information detection (Transition t3)

1: Input: Cred_URL_info $CUF = \{F_1, F_2, F_3, \ldots..F_{n3}\}$

2: Input: Incred_URL_info $ICUF = \{F_1, F_2, F_3, \ldots..F_{n4}\}$

3: Input: Cred_NO_URL_info
$CNUF = \{F_1, F_2, F_3, \ldots..F_{n5}\}$

4: Input: Incred_NO_URL_info
$ICNUF = \{F_1, F_2, F_3, \ldots..F_{n6}\}$

5: Output: Good Info
$GF = \{F_1, F_2, F_3, \ldots..F_{n7}\}$, $(n7 = n_3 + n_5)$

6: Output: Misleading Info
$MF = \{F_1, F_2, F_3, \ldots..F_{n8}\}$, $(n8 = n_4 + n_6)$

7: Procedure Good/Misleading Info Detection

8: $GF \leftarrow CUF \cup CNUF$

9: $MF \leftarrow ICUF \cup ICNUF$

10: return $GF = \{F_1, F_2, F_3, \ldots..F_{n7}\}$

11: return $MF = \{F_1, F_2, F_3, \ldots..F_{n8}\}$

12: End Procedure

Algorithm 4: Propagating/Blocking Information (Transition *t4*)

1: **Input:** Good Info $GF = \{F_1, F_2, F_3, \ldots..F_{n7}\}$

2: **Input:** Misleading Info $MF = \{F_1, F_2, F_3, \ldots..F_{n8}\}$

3: **Output:** CPN_{Output}

4: **Procedure** *Good/Misleading Info Detection*

5: *if* $(MF = \emptyset)$

6: $CPN_{Output} \leftarrow Propagate(GF)$

7: else

8: $CPN_{Output} \leftarrow Block(MF)$

9: **End Procedure**

The general Flowchart of detecting and blocking rumors patterns with respect to the methodology of the proposed CPNM is depicted in Figure 2

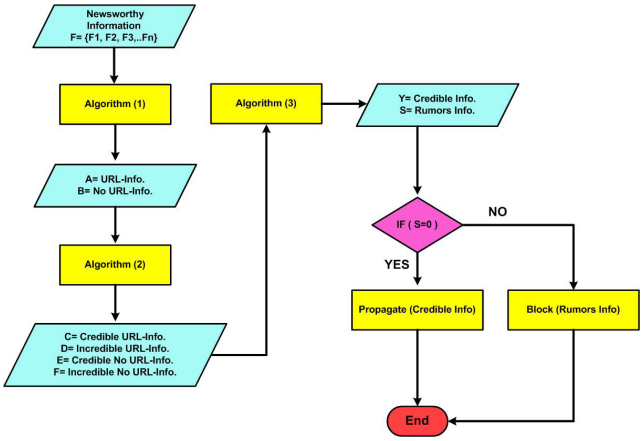

Figure 2. Flowchart of CPNM Functionality.

3. Results and Discussion

The CPN simulation tool[16] is used for investigating the performance of the proposed CPNM approach in detecting and blocking rumors patterns on a dataset of 863 newsworthy tweets that collected from Twitter. The dataset is described as trending topic *#CharlieHebdo*, which involves several newsworthy tweets from different sources of information. Table 2 provides numbers of tweets in each pattern of information sources.

Table 2. Tokens values (or number of tweets) according to the sources of tweets in *Dataset(#CharlieHebdo)*

Dataset 3: (#CharlieHebdo)	
Source	**# Tweets (Tokens Values)**
Newspapers	116
Magazines	87
TV Channels	101
Radios	76
Online Sites	174
Wire Service	92
Blogs	217
Sum	**863**

The 863-tweets are classified into seven patterns according to the sources of information. The *Twitter R library*[17] tool is used for collecting all tweets in the handled dataset. The practical simulation has started by initializing the proposed CPNM as $M_0 = \{863, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$. Table 3 provides the results obtained from simulating the tokens-firing in the ten places of CPNM according to the firing sequence $\sigma = t_1, t_2, t_3, t_5, t_4$.

Figure3 shows the tokens distribution according to the color set of the places $P_1, P_2, P_3, \ldots . P_{10}$.

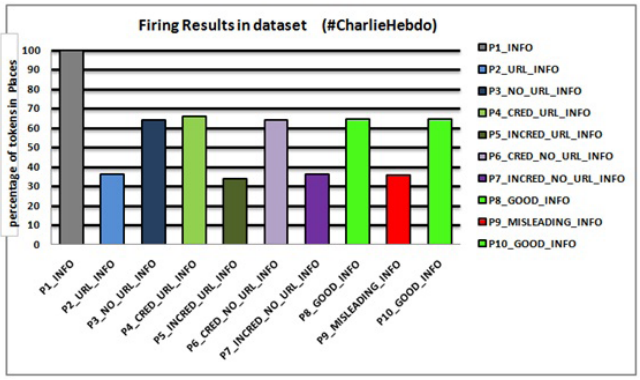

Figure 3. Percentages of tokens in the ten places according to places color sets.

Figure 4 shows a pie plot of the percentage of credible tweets and rumors tweets in the handled dataset *#CharlieHebdo*.

Table 3. Tokens-distribution according to the firing sequence $\sigma = t_1, t_2, t_3, t_5, t_4$ in dataset *(#CharlieHebdo)*

Dataset 3: #CharlieHebdo										
	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**
t1	0	347	516	0	0	0	0	0	0	0
t2	0	0	0	263	84	294	222	0	0	0
t3	0	0	0	0	0	0	0	**557**	**306**	0
t5	0	0	0	0	0	0	0	557	0	0
t4	0	0	0	0	0	0	0	0	0	**557**

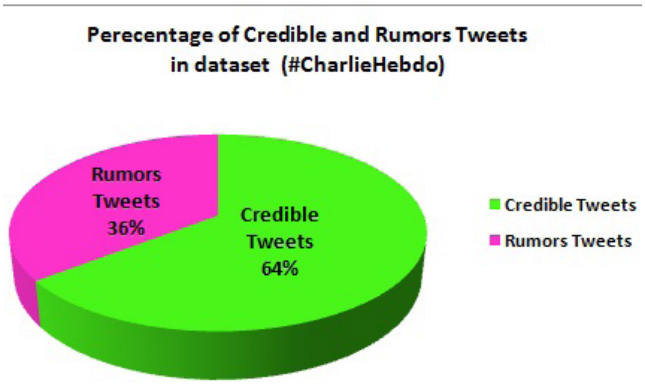

Figure 4. percentage of credible tweets and rumors tweets in *#CharlieHebdo* dataset.

Table 4 summarizes the performance evaluation results of the proposed CPNM approach in detecting rumors patterns.

The experimental simulation demonstrated some interesting findings. One interesting finding is that the proposed CPNM achieved competitive values of exactness (i.e. Precision =0.91), completeness (i.e. Recall=0.82), and Accuracy = 0.90 in detecting rumors-tokens in the handled dataset. Figure 5 provides the comparison results with other mechanisms in terms of detecting rumors information in Twitter based on different features.

Table 4. Evaluating the Performance of CPNM

Metric	Formula	#CharlieHebdo
True Positive (TP)	Correct Detection	279T
False Positive (FP)	Incorrect Detection	27T
True Negative (TN)	Correct Rejection	495T
False Negative (FN)	Incorrect Rejection	62T
Condition Positives (P)	$P = TP + FN$	341T
Condition Negatives (N)	$N = FP + TN$	522T
Precision (PPV)	$PPV = \frac{TP}{TP + FP}$	0.91
Recall (TPR)	$TPR = \frac{TP}{TP + FN}$	0.82
Specificity (TNR)	$TNR = \frac{TN}{FP + TN}$	0.95
Negative Predictive Value (NPV)	$NPV = \frac{TN}{TN + FN}$	0.89
False Positive Rate (FPR)	$FPR = \frac{FP}{FP + TN}$	0.05
False Discovery Rate (FDR)	$FDR = \frac{FP}{FP + TP}$	0.09
False Negative Rate (FNR)	$FNR = \frac{FN}{FN + TP}$	0.18
Accuracy (Acc)	$Acc = \frac{TP + TN}{P + N}$	0.90

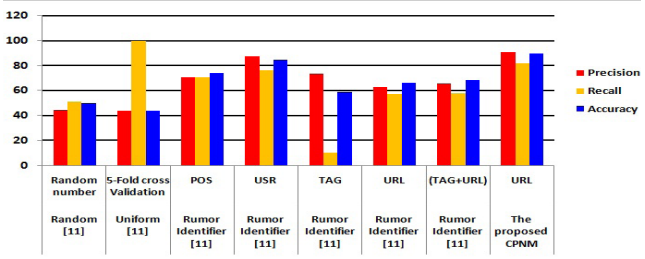

Figure 5. Comparison Results with other Methods in terms of Detecting Rumors in Twitter.

Another interesting finding is that verifying the proposed CPNM against Reachability proved that the marking state $M_3 = \{0, 0, 0, 0, 0, 0, 0, \mathbf{557}, \mathbf{306}, 0\}$ is the state in which CPNM could detect 306 tokens in place P_9 as rumors tweets and could detect 557 tokens in place P_8 as credible tweets. In addition, the marking state $M_4 = \{0, 0, 0, 0, 0, 0, 0, \mathbf{557}, 0, 0\}$ is the state in which CPNM could block and remove rumors tweets (i.e. 306 tokens) from place P_9. Finally, the marking state $M_5 = \{0, 0, 0, 0, 0, 0, 0, 0, 0, \mathbf{557}\}$ is the last marking state in which the CPNM produce only 557 tokens as credible tweets in place P_{10}. Figure 6 demonstrates the Reachability graph, which represent all marking states according the firing sequence $\sigma = t_1, t_2, t_3, t_5, t_4$ while simulating the proposed CPNM in the handled dataset *#CharlieHebdo*.

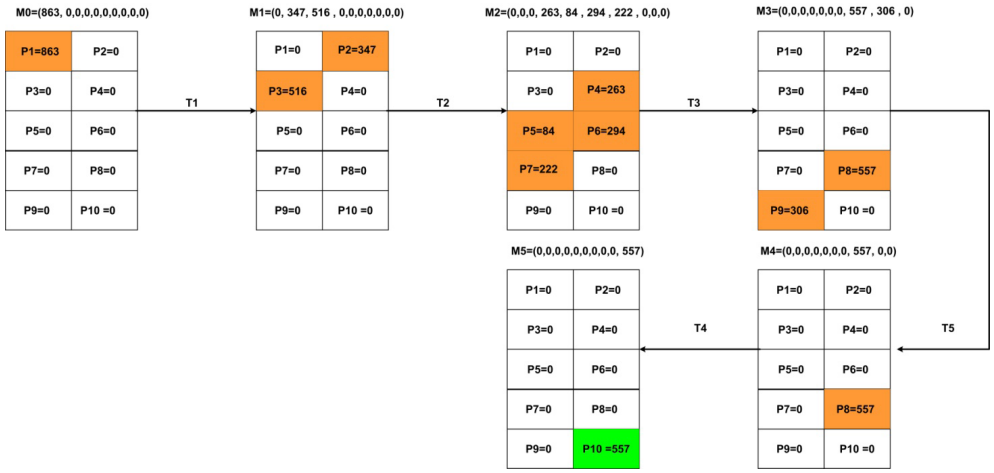

Figure 6. Reachability Graph of CPNM when simulated on *#CharlieHebdo* dataset.

4. Conclusion

In this study, we proposed a Colored Petri Net Model (CPNM) for recognizing rumors information and blocking its propagation over social networks. The proposed approach is experimentally simulated on 863 tweets collected from Twitter. The experimental results have shown that the CPNM achieved a competitive level of exactness (i.e. precision=91%), Completeness (i.e. Recall=82%), Accuracy 90%, and Low False Positive Rate (i.e. FPR=5%) in detecting rumors tweets compared with other methods in the literature. In addition, the Reachability analysis proved that the CPNM is able to block the propagation of detected rumors tokens and produce credible tokens with respect to the firing sequence life cycle. More research trials are needed to improve the accuracy of the proposed CPNM on different datasets of different social network platforms as a future work in this area.

5. References

1. Kumar KP, Geethakumari G. Detecting Misinformation in Online Social Networks using Cognitive Psychology. Human-Centric Computing and Information Science Springer. 2014; 4(1):1–22. Crossref
2. Karlova NA, Fisher KE, Plz RT. A Social Diffusion model of Misinformation and disinformation for understanding human information behavior. Inform Res. 2013; 18(1):1–17.
3. Gupta A, Kumaraguru P. Credibility Ranking of Tweets During High Impact Events. Proceedings of the 1st Workshop on Privacy and Security in Online Social Media, ACM Lyon France: 2012. p. 2–6. Crossref
4. Liu B. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies. 2012; 5(1):1–167.
5. Torky M, Babars R, Ibrahim R, Hassanein AE, Schaefer G, Zhu SY. Credibility Investigation of Newsworthy Tweets Using a Visualising Petri Net Model. Proceedings of IEEE International Conference on Systems Man and Cybernetics, 2016. p. 003894–8. Crossref
6. Nivedah R, Sairam N. A Machine Learning based Classification for Social Media Messages. Indian Journal of Science and Technology. 2015; 8(16):1–4. Crossref
7. Cstillo C, Mendoza M, Poblete B. Information Credibility on Twitter. Proceedings of WWW 2011 international Conference on Information Credibility, 2011. p. 675–84. Crossref
8. Morris MR, Counts S, Roseway A, Hoff A, Schwarz J. Tweeting is Believing Understanding Microblog Credibility Perceptions. Proceedings of the CSCW 2012 Conference, ACM, Seattle Washington USA: 2012. p. 441–50.
9. Abbasi MA, Liu H. Measuring user credibility in social media. Proceedings of International Conference on Social Computing Behavioral-Cultural Modeling and Prediction, Springer Berlin Heidelberg; 2013 Apr. p. 441–8.Crossref
10. Nguyen DT, Nguyen NP, Thai MT. Sources of Misinformation in Online Social Networks Who to Suspect. Proceedings of Military Communications Conference IEEE, Florida USA: 2012 Oct. p. 1–6.Crossref
11. Qazvinian V, Rosengren E, Radev DR, Mei Q. Rumor has it Identifying misinformation in microblogs. Proceedings of

the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics; 2011. p. 589–1599.
12. Lappas T, Terzi E, Gunopulos D, Mannila H. Finding effectors in social networks. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, Washington: 2010 Jul. p. 1059–68. Crossref
13. Nguyen NP, Yan G, Thai MT. Analysis of misinformation containment in online social networks. Computer Networks Elsevier. 2013; 57(10):2133–46.Crossref
14. Budak C, Agrawal D, Abbadi A. Limiting the spread of misinformation in social networks. Proceedings of the 20th international conference on World wide web ACM, Hyderabad India: 2011. p. 665–74. Crossref
15. Jensen K. Coloured Petri nets basic concepts analysis methods and practical use. 2nd ed. Springer Science Business Media; 1997. Crossref
16. Simulator functions CPN Tools. Crossref
17. R project. Crossref