International Journal of Pure and Applied Mathematical Sciences.
ISSN 0972-9828 Volume 10, Number 1 (2017), pp. 69-80

http://www.ripublication.com

RESEARCH FINDINGS OF GRAPH MINING MODELS AND TRANSFORMATIONS FOR SECURE BIG DATA NETWORK ANALYSIS

K.Rama Krishna[1] Dr. S.Sreekanth[2] Dr.M.Upendra Kumar[3]

[1]Research Scholar Mathematics Noida International University, NIU Noida India
Email: ramakrishna.koli@gmail.com

[2]Professor CSE, SITAMS, JNTU A, Chittor A.P, India.
Email: sreekanth@sitams.org

[3]Professor CSE, MGIT, JNTU H, Hyderabad T.S. India
Email: uppi_shravani@rediffmail.com

ABSTRACT

This research Entitled "Application of Graph Theory to Computer Science in Big Network for Big Data" is a novel and innovative idea of inter-disciplinary, application of Mathematical foundations of Computer Sciences, graph theoretic approach to Computer Science Big Networks for Big data for Network Analysis regarding Security and Privacy concerns. We have formulated a mathematical graph model using Graph Mining and validated it on various computer science case studies and considerable satisfactory empirical results were obtained for security and privacy metrics. First we provide Introduction to Topics of thesis and Research Motivation in Scope of thesis. Second us Provide Literature survey in important base papers and Problem statement. Third we provide Theoretical analysis of Proposed Graph Mining Model for Big Data Network Analysis for Security. Fourth we provide Experimental work of proposed Graph Mining model. Fifth we provide Case Studies implementation of experimental model and Sixth we provide Discussion of Results. Seventh we provide Conclusion and Future work of thesis. Finally thesis ends with Few Appendixes for additional supporting background

discussion of mathematical and mining strategies used in the thesis like Graph tools utilized, light weight programming for Graphs etc.

The main research idea of Graph Mining approach for Secure Big Data network is as follows: It consists of 3 phases. In Phase 1: Big Networks data are collected using a comprehensive algorithm. In Phase 2: Integrates a various Big Network data from phase 1 to develop web mining approach using association rules, classification and clustering techniques. In Phase 3: Information generated from web mining approach can be considered as base for developing a web based application that satisfies the need of end user requirements and also satisfying needs of business industry. Thus, phase 3 produces tasks related to web application structure, web services, web architectures, web configuration management data, web classification, web communities, website navigation, and web security.

Big data has 4 V's viz. Variety, Veracity, Velocity, and Value. Building a Big Data Platform includes phases like Capturing Data, Organizing Data, Analyzing data, Acting on data and measuring the results. Big data framework technologies include: Hadoop framework like Mapping and Reducing, Pig, Hive, Spark.

Proposed Graph Mining Mathematical Models includes Algorithms like sub linear algorithms and distributed graph algorithms. Graph and Model Transformation includes Graph Transformation like Node Label Replacement Approach, Hyper edge Replacement approach, Algebraic approach, Logical approach, Theory of 2-structure, Programmed Graph Replacement approach etc.

The Case study implemented was: Continuous queries are used to monitor changes to time varying data and to provide results useful for online decision making. Typically a user desires to obtain the value of some aggregation function over distributed data items, for example, to know value of portfolio for a client; or the AVG of temperatures sensed by a set of sensors. In these queries a client specifies a coherency requirement as part of the query. We present a low-cost, scalable technique to answer continuous aggregation queries using a network of aggregators of dynamic data items. In such a network of data aggregators, each data aggregator serves a set of data items at specific coherencies. Just as various fragments of a dynamic web-page are served by one or more nodes of a content distribution network, our technique involves decomposing a client query into sub-queries and

executing sub-queries on judiciously chosen data aggregators with their individual sub-query incoherency bounds. We provide a technique for getting the optimal set of sub-queries with their incoherency bounds which satisfies client query's coherency requirement with least number of refresh messages sent from aggregators to the client. For estimating the number of refresh messages, we build a query cost model which can be used to estimate the number of messages required to satisfy the client specified incoherency bound. Performance results using real-world traces show that our cost based query planning leads to queries being executed using less than one third the number of messages required by existing schemes. Putting it all together this strategy is an improved data slicing mechanism using overlapping and prevention of membership disclosure.

Key Words: Graph Theory, Computer Science, Mathematical Models, Model Transformations, Big Data, Big Networks, Security/Privacy.

1. Introduction to Model Transformations

In this general introduction we give a general overview of graph and model transformation and a short overview of the parts and chapters of the paper. The main questions are the following:

- What is graph transformation?
- What is the algebraic approach to graph transformation?
- What is model transformation?
- How can algebraic graph transformation support model transformation?

What is Graph Transformation?

Graphs are important structures in mathematics, computer science and several other research and application areas. A graph consists of nodes, also called vertices; edges; and two functions assigning source and targe nodes to each edge. In fact, there are several variants of graphs, like labeled, typed, and attributed graphs, which will be considered in this paper, because they are important for different kinds of applications. Prosperities of graphs, like shortest paths, are studied within graph theory, where in general the structure of the graph is not changed. Graph transformation, in contrast, is a formal approach for structural modifications of graphs vis the application of transformation rules. A graph rule, also called production $p = (L,R)$, consist of a left-hand side graph L, a right-hand side graph R and a mechanism specifying how to replace L by R as shown schematically in Fig. 1.

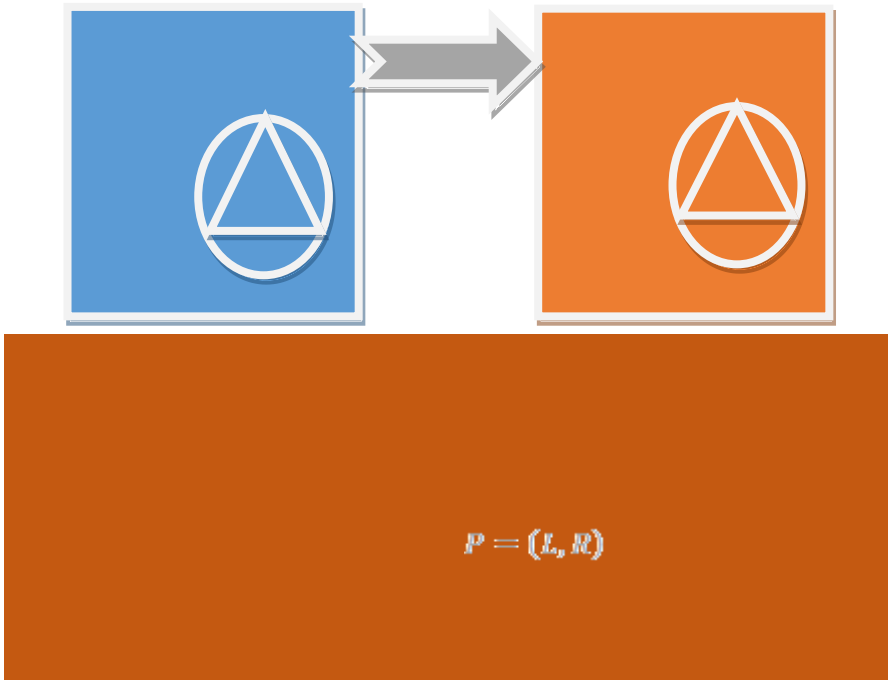

Fig. 1. Rule-based modification of graphs

The graph replacement mechanism is different in each of the following main graph transformation approaches presented in Volume 1 of the Handpaper of Graph Grammars and Computing by Graph Transformation [Roz97]:

- Node Label Replacement Approach
- Hyper edge Replacement Approach
- Algebraic Approach
- Logical Approach
- Theory of 2-Structure
- Programmed Graph Replacement Approach

In al approaches, a graph transformation system consists of a set of rules; moreover, a graph transformation system together with a distinct start graph forms a graph grammar.

What Is the Algebraic Approach to Graph Transformation?

In this paper, we present the algebraic approach of graph transformation, where a (basic) graph $G = (V, E, s, t)$ is algebra with base sets V (vertices), E (edges), and operations $s: E \to V$ (source) and $t: E \to V$ (target). Graph morphisms are special cases of algebra homomorphism$f = (f_V: V_1 \to V_2, f_E: E_1 \to E_2)$. This means that graph morphism is required to be compatible with the operations source and target. It

is important to note that graphs and graph morphisms define a category Graphs, such that categorical constructions and results are applicable in the algebraic approach of graph transformation. In fact, an important concept is the gluing construction of graphs, which corresponds to the pushout construction in the category Graphs**.** Pushout are unique up to isomorphism and have useful composition and decomposition properties. The main conceptual idea of gluing is the following: Given graphs G_1 and G_2 with common intersection G_0, the gluing G_3 of G_1 and G_2 along G_0, written $G_3 = G_1 + G_0$, G_2 is given by the union G_3 of G_1 and G_2 and shown in the gluing diagram in A production $P = (L \leftarrow K \rightarrow R)$ in the algebraic approach is given not only by left-and right-hand side graphs L and$R$, but, in addition, by a gluing graph K and (injective) graph morphisms from K to L and R. Given a context graph D with morphism$K \rightarrow D$ a direct graph transformation from a graph G to a graph H via a production p, written $G \Longrightarrow H$ via P, is given by two gluing (pushout) diagrams as gluing of R and D along K. In other words, L is replaced by R, while the context D remains unchanged. This definition of direct graph transformation is elegant, because it is well defined (up to isomorphism) and symmetric. However, it leaves open how to apply a production P to a given host graph G and how to calculate the host graph H. In order to apply a production $P = (L \hookleftarrow K \hookrightarrow R)$ to a graph$G$, we first have to find an occurrence of L in G, given by graph morphism $m: L \mapsto G$. is called match morphism. Then, we have to construct D and H in such a way that (1) and (2) become gluing (pushout) diagrams in Fig.1.3 is a pushout diagram. This means, given a production $P = (L \hookleftarrow K \mapsto R)$ and a match $m: L \mapsto G$ satisfying the gluing condition, we obtain in a first step the context graph D and gluing (push out) diagram (1) and in a second step diagram (2) by gluing (push out) construction. The first step corresponds to the deletion of $L \backslash K$ from G and the second step to the addition of $R \backslash K$ leading to H, written $G \Longrightarrow H$ via p and$m$. This algebraic approach is called double push out (DPO) approach, because a direct transformation consists of two push outs ion the category Graphs (see Fig.1.3). An important variant of the algebraic approach is the single push out (SPO) approach, where a direct transformation is defined by a single pushout in the category Graphs of graphs and partial graph morphism. In this paper, we mainly present the algebraic DPO approach of graph transformation. Moreover, we allow replacing the category of graphs by a suitable axiomatic category (see M-adhesive categories in Chap.4). This leads to the concept of M-adhesive transformation systems in the algebraic approach, which can be specialized to transformation systems for different kinds of graphs, petri nets, and other kinds of high-level replacement systems.

What is Model Transformation?

Model-driven software development (MDD) has been used successfully within the last two decades for the generation of software system. Especially, UML diagrams [UML,15] re useful for modeling different views of systems on an abstract level independently of specific implementations. In this case, models are UML diagrams, but in general models can be any kind of visual or textual artefacts. This culminates in the well-known slogan "Everything is a Model" stated in [Bez05]. Model transformation means defining transformations between (different) models. It plays a central role in MDD and several other applications. Model transformations in MDD are especially used to refactor models, to translate them to intermediate models, and to generate code. According to [CH06], we distinguish between endogenous and exogenous transformations. Endogenous transformations take place within one languages. Moreover, model-to-model transformations are usually distinguished from model-to-text transformations. Typical examples of model-to-model transformations are the transformation S2P from statecharts to Petri nets in [EEPT06] and CD2RDBM from class diagram to relational database models in this paper. Important properties for most kinds of model transformations are type consistency, termination, syntactical and semantic correctness, completeness, functional behavior and information preservation. We will discuss this topic in the next subsection and in Part III of this paper.

How Can Algebraic Graph Transformation Support Model Transformation?

In [CH06], an overview of various model transformation approaches is given following object-oriented, rule-based, constraint-based and imperative concepts. In the following, we show how algebraic graph transformation can support the definition and analysis of rule-based model transformation [Tae 10]. Especially for visual models, grapoh transformation is a natural choice for manipulating their underlying graph structures. The double pushout (DPO) approach introduced above can be interpreted as a kind of in-place transformation, where the source graph is transformed step by step into the target graph. Using the DPO approach for typed graphs-with different type graphs for source and target domain-allows us to ensure type consistency by construction [EEPT06]. The rich theory of the DPO approach provides support for the verification of other properties of model transformations discussed above [EE08]. Even better support for the verification of these properties is given by the ripple graph grammar (TGG) approach [KS06, EEE'07] presented in Chap.3 and Part III of this paper. A triple graph consists of a source graph, a target graph, and a correspondence graph. The last one is mapped to the source and the target graph is order to establish a correspondence between elements of these graphs. The TGG approach is closely related to the DPO approach, in the way that graphs are replaced by triple graphs and TGG rules are usually nondeleting. The main additional idea is the following: From each TGG rule, a forward and a backward rule can

derived automatically, which allows us to construct type-consistent any syntactically correct forward and backward transformations between the source model and target model domains.

Historical Notes

Historically, graph grammars and transformations were first studied as "web grammars by Pfalz and Rosenfeld [PR69] in order to define rule-based image recognition. Part [Pra71] used pair graph grammars for string-to-graph translations, similar to the concept of the triple graph grammar approach. The historical roots of the algebraic approach were presented by Ehrig, Pfender, and Schneider [EPS73]. The first introduction to the DPO approach-including the well-known Local Church-Rosser Theorem-was presented by Ehrig and Rosen in [ER76. Ehr79]. The first paper on graph grammars was published by Nag [Nag79] with its main focus on the Chomsky hierarchy, implementation and applications. The concept of graph transformation has at least three different historical roots:

1. From Chomsky grammars on strings to graph grammars,
2. From term rewriting to graph rewriting.
3. From textual description to visual modeling.

Motivated by these roots, the concept of "Computing by Graph Transformation" was developed as a basic paradigm in the ESPRIT Basic Research Actions COMPUGRAPH and ALLIGRAPH, and continued in the TMR Networks GETGRATS and SEGRAVIS in the period 1990-2006. The state of the art of graph transformation and their applications of 15 years ago is documented in there volumes of the "Handpaper of Graph Grammars and Computing by Graph Transformation" [RoZ97, EEKR99], where [RoZ97] includes an introduction to the algebraic SPO and DPO approaches. A first detailed part of the theory of the DPO approach was published in the EATCS Monographs in TCS [EEPT06], while the newer developments are presented in this paper. We present its main concepts, based on the exytended theory of M-adhesive transformation systems [EGH10], including results for parallelism, concurrency and amalgamation [EGH'14]; results for systems with nested application conditions concerning embedding, critical pairs and local confluence[EGH^+12]: characterizations of constructions based on the notion of finitary M-adhesive categories [GBEG14]; concurrency based on permutation equivalence [HCE14]; and model transformation and model synchronization based on triple graophy grammars [HEGO14, HEO^+13]

2. Validation on case study

EXISTING SYSTEM

First, many existing clustering algorithms (e.g., k- means) requires the calculation of the "centroids". But there is no notion of"centroids"in our setting where each attribute forms a data point in the clustering space. Second, k-medoid method is very robust to the existence of outliers (i.e., data points that are very far away from the rest of data points). Third, the order in which the data points are examined does not affect the clusters computed from the k-medoid method.

DISADVANTAGES

1. Existing anonymization algorithms can be used for column generalization, e.g., Mondrian. The algorithms can be applied on the sub table containing only attributes in one column to ensure the anonymity requirement.
2. Existing data analysis (e.g., query answering) methods can be easily used on the sliced data.
3. Existing privacy measures for membership disclosure protection include differential privacy and presence.

1.2.2 PROPOSED SYSTEM

We present a novel technique called slicing, which partitions the data both horizontally and vertically. We show that slicing preserves better data utility than generalization and can be used for membership disclosure protection. Another important advantage of slicing is that it can handle high-dimensional data. We show how slicing can be used for attribute disclosure protection and develop an efficient algorithm for computing the sliced data that obey the ℓ-diversity requirement. Our workload experiments confirm that slicing preserves better utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute.

ADVANTAGES:

1. We introduce a novel data anonymization technique called slicing to improve the current state of the art.
2. We show that slicing can be effectively used for preventing attribute disclosure, based on the privacy requirement of ℓ-diversity.
3. We develop an efficient algorithm for computing the sliced table that satisfies ℓ-diversity. Our algorithm partitions attributes into columns, applies column generalization, and partitions tuples into buckets. Attributes that are highly-correlated are in the same column.
4. We conduct extensive workload experiments. Our results confirm that slicing preserves much better data utility than generalization. In workloads involving the sensitive attribute, slicing is also more effective than bucketization. In some classification experiments, slicing shows better performance than using the original

data (which may overfit the model). Our experiments also show the limitations of bucketization in membership disclosure protection and slicing remedies these limitations.

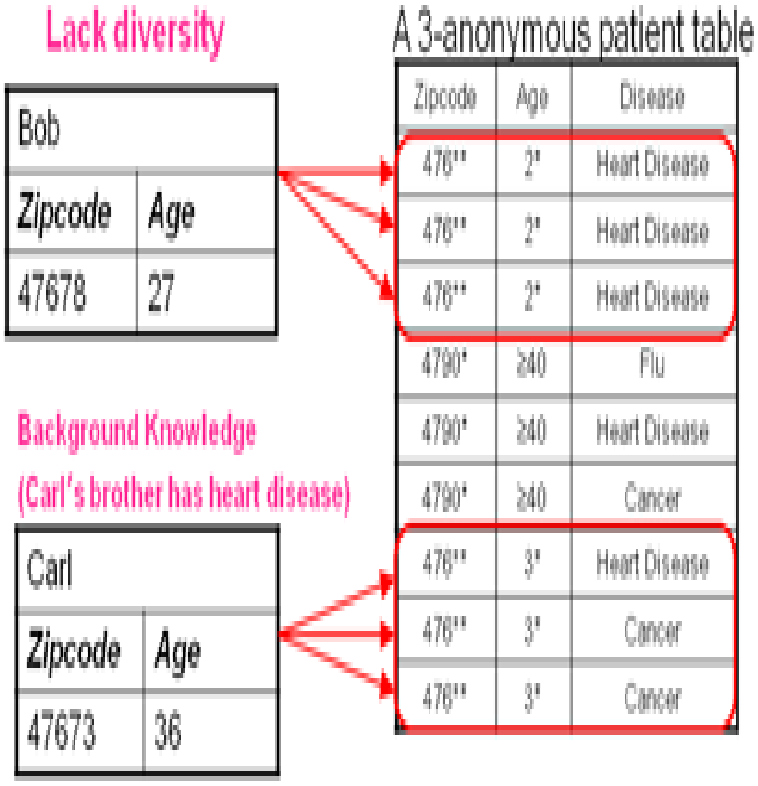

Fig 2. System Architecture

3. CONCLUSIONS AND FUTURE ENHANCEMENT

This work motivates several directions for future research.

First, we consider slicing where each attribute is in exactly one column. An extension is the notion of overlapping slicing, which duplicates an attribute in more than one column. These releases more attribute correlations. For example, in Table,

one could choose to include the Disease attribute also in the first column. That is, the two columns are {Age, Sex, and Disease} and {Zipcode, Disease}. This could provide better data utility, but the privacy implications need to be carefully studied and understood. It is interesting to study the tradeoff between privacy and utility.

Second, we plan to study membership disclosure protection in more details. Our experiments show that random grouping is not very effective. We plan to design more effective tuple grouping algorithms.

Third, slicing is a promising technique for handling high-dimensional data. By partitioning attributes into columns, we protect privacy by breaking the association of uncorrelated attributes and preserve data utility by preserving the association between highly-correlated attributes. For example, slicing can be used for anonymizing transaction databases, which has been studied recently.

Finally, while a number of anonymization techniques have been designed, it remains an open problem on how to use the anonymized data. In our experiments, we randomly generate the associations between column values of a bucket. This may lose data utility. Another direction to design data mining tasks using the anonymized data computed by various anonymization techniques.

REFERENCES:

[1]. Geirino Mazzola, Gerard Milmeister, Jody Weissmann, "Comprehensive Mathematics for Computer Scientists", Spriger-Verlag Berlin Heidelberg 2006, ISBN: 3-540-36873-6.

[2]. Kai-Tai Fang Hong Knong, P.R. China Runze Li,"Design and Modeling for Computer Experiments", Taylor & Francis Group, 2006.

[3]. Narsing Deo, N. Millican, Orlando. Narsingh Deo, "Graph Theory with Applications to Engineering and Computer Science", 2014, PP.7-8.

[4]. Jonathan D.H.Smith, "On the challenges of Mathematics", 2014, P.4.

[5]. Pooja Mohan, Manpreet Singh "Formal Models for Context Aware Computing", 2013, PP. 53-58.

[6]. Jun Zhao, Osman Yagan, Virgil Gligor, "Random Intersection Graphs and Their Applications in Security, Wireless Communication, and Social Networks", the paper summarizes some of the results in our work [40]-[54].

[7]. Arie van Deursen, Ali Mesbah, Alex Neder of, "Crawl-Based Analysis of Web Applications: : Prospects and Challenges", 2014,.1-10.

[8]. Ibrahim AbakerTargio Hashem, IbrarYaqoob, Nor BadrulAnuar, Salimah Mokhtar, Abdullah Gani, Samee Ullah Khan, "The rise of big data on cloud computing: Review and open research issues", 2015, PP.98-115.

[9]. Mehdi Bahrami, Mukesh Singhal, "The Role of Cloud Computing Architecture in Big Data", Springer 2015, PP. 275-295.

[10]. Ivan Rodero, Marish Parashar, Omer F. Rana, Toan Petri, "Incentivizing Resource Sharing in Social Clouds", January 2012.

[11]. Markus Leitner, Oskar-Morgenstern-Platz1, "Layered Graph Models and Exact Algorithms for the Generalized Hop-Constrained Minimum Spanning Tree Problem", January 2015.

[12] .Eugenio M. Fedriani, Angel F. Tenorio, Methods Cuantitativos, "Fundamental Products and Autonomous Sets: An Algorithmic Approach", 2015, PP.1.9-18.

[13]. Albert Atserias, Armin Biere, Samuel Buss, Antonina Kolokolova, Jakob Nordstrom, KaremSakallah,"Theoretical Foundations of Applied SAT Solving (14w5101)", January 2014.

[14]. Shin-Shin Kao, Hsiu-Chunj Pan, "A Further Study on the 4-Ordered Property of Some Chordal Ring Networks", 2015.

[15]. N.M.M.Yusop, M.K.Hasan, M, Rahmat "Comparison New Algorithm Modified Euler in Ordinary Differential Equation Using Scilab Programming", August 2015.

[16]. N.M.Singhi, M.K.Srinivasan, "Some problems in combinatorics".

[17]. S.Sekar, M.Nalini,"Analysis of the Non linear Singular systems using a domain Decomposition Method", Pearson tech group, 2009, PP.1-4.

[18]. Bernhard Thalheim, Klaus-Dieter, Schewe Andreas, Prinz Bruno Buchberger "Texts & Monographs in Symbolic Computation", ISSN: 0943-853X, ISSN: 2197-8409 (Electronic), Springer International Publishing Switzerland, 2015, DOI: 10.1007/978-3- 319-17112-8.

[19]. Ronald L, Graham, Donald E.Knuth, Oren Patashnik, "Concrete Mathematics", Published by Addison-Wesley, 1989, ISBN: 0-201-55802-51.

[20]. R.M.R. Lewis, "A Guide to Graph Coloring Algorithms and Applications", Springer International Publishing Switzerland 2016, ISBN: 978-3-319-25728-0, ISBN: 978-3-319-25730-3 (ePaper), DOI: 10.1007/978-3-319-25730-3.

[21]. Nathalie Japkowicz, "Big Data Analysis: New Algorithms for a New Society", Springer International Publishing Switzerland 2016, ISSN: 197-6503, ISSN2197-6511 (electronic), ISBN: 978-3-319-26987-0, ISBN: 978-3-319-26989-4 (ePaper), DOI: 10.1007/978-3-319-26989-4.

[22]. Deze Zeng, Lin Gu, Song Guo, "Cloud Networking for Big Data", Springer International Publishing Switzerland 2015, ISSN: 2366-1186, ISSN: 2366-1445 (electronic), ISBN: 978-3-319-24718-2, ISBN: 978-3-319-24720-5 (ePaper), DOI: 10.1007/978-3-319-24720-5.

[23]. MarttiLehto, Pekka Neittaanmaki, "Cyber Security: Analytics, Technology and Automation", Springer International Publishing Switzerland 2015, ISSN:

2213-8986, ISSN: 2213-8994 (electronic), ISBN: 978-3-319-18301-5, ISBN: 978-3-319-18302-2 (ePaper), DOI: 10.1007/978-3-319-18302-2.

[24]. Dark Web, "Exploring and Data Mining the Dark Side of the Web", Springer Science Business Media, LLC, 2012, ISSN: 1571-0270, ISBN: 978-1-4614-1556-5 e-ISBN: 978- 1-4614-1557-2, DOI: 10.1007/978-1-4614-1557-2.

[25]. Niall Adams, Nicholas Heard, "Data Analysis for Network Cyber-Security", Published by Imperial College 2014, ISBN: 978-1-78326-374-5.

[26]. K.Erciyes," Distributed Graph Algorithms for Computer Networks", Springer London Heidelberg New York, ISSN: 1617-7975, ISBN: 978-1-4471-5172, ISBN: 978-1-4471- 5172-2, ISBN: 978-1-4471-5173-9 (ePaper), DOI: 10.1007/978-1-4471-5173-9.