Computer Science & Information Technology 81

David C. Wyld
Natarajan Meghanathan (Eds)

Computer Science & Information Technology

11th International Conference on Security and its Applications
(CNSA 2018) January 2~3, 2018, Zurich, Switzerland

AIRCC Publishing Corporation

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Natarajan Meghanathan,
Jackson State University, USA
E-mail: nmeghanathan@jsums.edu

ISSN: 2231 - 5403
ISBN: 978-1-921987-79-3
DOI : 10.5121/csit.2018.80101 - 10.5121/csit.2018.80107

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The 11th International Conference on Security and its Applications (CNSA 2018) was held in Zurich, Switzerland, during January 02~03, 2018. The 5th International Conference on Data Mining and Database (DMDB 2018) and The 5th International Conference on Artificial Intelligence and Applications (AIAP 2018) was collocated with The 11th International Conference on Security and its Applications (CNSA 2018). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The CNSA-2018, DMDB-2018, AIAP-2018 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically. All these efforts undertaken by the Organizing and Technical Committees led to an exciting, rich and a high quality technical conference program, which featured high-impact presentations for all attendees to enjoy, appreciate and expand their expertise in the latest developments in computer network and communications research.

In closing, CNSA-2018, DMDB-2018, AIAP-2018 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the CNSA-2018, DMDB-2018, AIAP-2018.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld
Natarajan Meghanathan

Organization

General Chair

David C. Wyld	Southeastern Louisisna University, USA
Jan Zizka	Mendel University in Brno, Czech Republic

Program Committee Members

Ali Abid D. Al-Zuky	Mustansiriyah University, Iraq
Amol D Mali	University of Wisconsin-Milwaukee, USA
Anandi G	Principal Research Scientist, Indian Institute of Science
Anita Yadav	Harcourt Butler Technical University, India
Ankit Chaudhary	Truman State University, USA
Annamalai	Prairie View A&M University, United States of America
Bing Zhou	Sam Houston State University, USA
Carmen Martinez	University of Jaen, Spain
Christophe NICOLLE	Universite de Bourgogne Franche Comte, France
Da Yan	The University of Alabama at Birmingham, USA
Daniel E. Asuquo	University of Uyo, Nigeria
Debabrata Datta	St. Xavier's College, India
Ettefagh	University of Tabriz, Iran
Fadiya Samson Oluwaseun	GIRNE AMERICAN University, Turkey
Fatih Korkmaz	Çankiri Karatekin University, Turkey
Fernando Zacarias	University of Puebla, Mexico
Francisco Macia Perez	University of Alicante, Spain
Fulvia Pennoni	University of Milano-Bicocca, Italy
Gábor Kiss	Óbuda University, Hungary
Gamini Wijayarathna	University of Kelaniya, Sri Lanka
Gammoudi Mohamed Mohsen	University of Manouba, Tunisia
Gheorghi Guzun	The University of Iowa, USA
Guilherme Gomes	Federal University of Itajubá, Brazil
Hamadouche Maamar	USDB, Algeria
Hamdi Hassen	MIRACL Laboratory, Tunisia
Hanen Idoudi	University of Manouba, Tunisia
Hao Li	Intel, China
Hao-En Chueh	Yuanpei University, Taiwan, Republic of China
Hayati Mamur	Cankiri Karatekin University, Turkey
Hung Tran Cong	Posts and Telecoms Institute of Technology, Viet Nam
Huseyin Cakir	Gazi University, Turkey
Idris Ismaila	Federal University of Technology, Nigeria
Isa Maleki	Islamic Azad University, Iran
Islam Atef	Alexandria Uniuversity, Egypt
Ismaila Idris	Universiti Teknologi Malaysia, Johor
Ivo Pierozzi Junior	Embrapa Agricultural Informatics, Brazil
Jafar Mansouri	Ferdowsi University of Mashhad, Iran

Jayan V	Centre for Development of Advanced Computing, India
Jia Zhu	South China Normal University, China
Jonice Oliveira	Universidade Federal do Rio de Janeiro (UFRJ), Brazil
Jose A.R.Vargas	University of Brasilia, Brazil
Jose Vicente Berna	University of Alicante, Spain
Jun Peng	University of Texas Rio Grande Valley, USA
Khaled Ahmed Abood Omer	University of Najran, KSA
Kosai Raoof	Le Mans Universite, Laboratoire LAUM, France
Kunwar Singh Vaisla	BT Kumaon Institute of Technology, India
Lakshmi Rajamani	Osmania University, India
Lyu Huiqiang	Zhejiang University of Technology, China
Mahesha D.M	Karnataka State Open University, India
Majid Abdollahzade	K.N.Toosi University of Technology, Iran
Manal Abdullah	King Abdulaziz University, KSA
Manish Kumar Mishra	University of Gondar, Ethiopia
Marius CIOCA	Lucian Blaga University of Sibiu, Romania
Messaoud Mezati	University of Ouargla, Algeria
Mohamed BEN AOUICHA	Faculty of Sciences, Tunisia
Mohammed Al-Mai'itah	Al-Balqa applied university, Jordan
Muhammad Asif Naeem	Auckland University of Technology, Auckland
Naren J	SASTRA University, India
Nasser Tairan	King Khalid University, Saudi Arabia
Natarajan Meghanathan	Jackson State University, USA
Neda Darvish	Islamic Azad University, Iran
Nidal M. Turab	Al-AHliyya Amman University, Jordan
Park Youngeun	Prince Sultan University in Riyadh, Saudi Arabia
Peiman Mohammad	Islamic Azad University, Iran
Pragya Shukla	Institute of Engineering and Technology, India
Rahil Hosseini	Azad University, Iran
Rahul Moriwal	Acropolis Institute of Technology & Research, India
Rajdeep Chowdhury	JIS College of Engineering, India
Ramgopal Kashyap	SISTec, India
Saad M. Darwish	Alexandria University, Egypt
Seungmin Rho	Sungkyul University, Korea
Seyyed Reza Khaze	Islamic Azad University, Iran
Siamak hoseinzadeh	Islamic Azad university, Iran
Soumaya Chaffar	Prince Sultan University, UAE
Sukant K. Mohapatra	Ericsson, USA
Susheel Kumar K	Ideal Institute of Technology Ghaziabad, India
Taruna S	Banasthali University, India
Tony Tsang	Hong Kong College of Technology, Hong Kong
Uduak Umoh	University of Uyo, Nigeria
Uttam Ghosh	Tennessee State University, USA
Victor M. Larios	University of Guadalajara, Mexico
Vivek Kumar	Gurukula Kangri Vishwavidyalaya, India
Yoo, Wook-Sung	Fairfield University, USA

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Networks & Communications Community (NCC)

Digital Signal & Image Processing Community (DSIPC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

11th International Conference on Security and its Applications (CNSA 2018)

5th International Conference on Data Mining and Database (DMDB 2018)

5th International Conference on Artificial Intelligence and Applications (AIAP-2018)

A PRACTICAL CLIENT APPLICATION BASED ON ATTRIBUTE-BASED ACCESS CONTROL FOR UNTRUSTED CLOUD STORAGE

Julian Jang-Jaccard

Computer Science and Information Technology, Institute of Natural and Mathematical Sciences (INMS), Massey University, New Zealand

ABSTRACT

One of widely used cryptographic primitives for the cloud application is Attribute Based Encryption (ABE) where users can have their own attributes and a ciphertext encrypted by an access policy. Though ABE provides many benefits, the novelty often only exists in an academic world and it is often difficult to find a practical use of ABE for a real application. In this paper, we discuss the design and implementation of a cloud storage client application which supports the concept of ABE. Our proposed client provides an effective access control mechanism where it allows different types of access policy to be defined thus allowing large datasets to be shared by multiple users. Using different access policy, each user only needs to access only a small part of the big data. The goal of our experiment is to explore the right set of strategies for developing a practical ABE-based system. Through the implementation and evaluation, we have determined the various characteristics and issues associated with developing a practical ABE-based application.

KEYWORDS

Attribute-based encryption, cloud storage service, Access control

1. INTRODUCTION

Cloud storages are widely used by organizations and people to outsource data. They have become more popular since cloud service providers offer their users cost efficient data storage with high mobility and reliability. However, the data in the cloud can be misused by cloud service providers or leaks the private sensitive data when there are inappropriate security mechanisms in place as seen in [1], [2]. Securing the data on the cloud still remains as a primary concern for many users.

One of the most promising ways to protect data on the cloud is encryption. Encryption offers additional protection as data is encrypted before they are uploaded to the cloud and decrypted after they are downloaded. However, traditional encryption technique, such as public key encryption, only allows a single user to decrypt the data which is too restrictive and ineffective in many cases. In particular in many modern applications, the data size is big and the data often

David C. Wyld et al. (Eds) : CNSA, DMDB, AIAP - 2018
pp. 01– 15, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80101

requires many different types of access policies to allow multiple users to share the data while each user only have access to a small part of it.

To improve the data sharing, cloud applications require more flexible fine-grained access control to support multiple users with different access requirements. Recent achievements in cryptographic theory such as Attribute Based Encryption (ABE) [15], [16] provide solutions to this data sharing dilemma. In ABE, in particular Ciphertext-Policy ABE (CP-ABE), users have their own attributes and a ciphertext is created with association of an access policy. Multiple users are able to decrypt the ciphertext if the users' attributes pass through the ciphertext's access structure.

Though the novelty provided by ABE scheme has been endorsed, the novelty only exists in an academic world as theory. The practical use of ABE scheme in real life applications have not been explored well. In this paper, we discuss the design and implementation details to apply an ABE scheme for a real life application. Our proposed client application sits in between the cloud application (such as DropBox, Google Drive, Microsoft OneDrive) and cloud storages (such as Amazon S3). Our client application not only provides encryption to safeguard user's sensitive data but also offer an effective access control mechanism to allow multiple users to share the data. The goal of our proposal is to provide opportunities to explore the right set of strategies for developing a practical ABE-based application. We offer the details of various algorithm implementations and issues associated with developing a practical ABE-based application.

The paper is organised as following: design considerations are presented in Section 2. Section 3 provides background material our proposal is based on. Section 4 describes the system architecture. Our implementation details, algorithms, and performance results are described in Section 5. The lessons we learnt during the proof-of-concept demonstrator is presented in Section 6. Section 7 concludes the paper along with the future work.

2. DESIGN CONSIDERATION

In this section, we discuss a number of priorities that we considered to design and implement our proposed client application.

2.1 Man-in-the-middle Attack

Most cloud storage service today provides a cloud application where a user can upload and subsequently download data using their own device at the comfort of home. More often than not, the data being transferred from user's device to the cloud storage server in not encrypted. This increases the chance for a malicious man-in-the-middle either to make an unauthorized access to user's data (i.e., breaking data confidentiality) or an unauthorized modification to the data (i.e., breaking data integrity). To provide data confidentiality and integrity, a protection mechanism must be employed such as an encryption. The secret key used for the encryption must be securely protected to prevent any potential snooping or stealth.

2.2 Dishonest Cloud Provider

Dishonest cloud provider makes an unauthorized attempt to read user's data. The cloud storage provider has all necessary tools and mechanisms to access the data that is under its full control.

Either by a malicious code implanted in the cloud server or by deliberate attempts by a dishonest cloud server administrator, the chance for data leakage increases if user's data is improperly protected. In addition, it is also possible that dishonest cloud provider makes copies of user's data. It is a common practice for cloud storage providers to make copies of original data and store them in extra storages, such as in the backup media or replicated databases. These copied datasets are then re-applied when an unexpected disaster occurs therefore user's data is recovered. However, even after user's subscription expires, these copies of the original data may still remain somewhere in the cloud server and targeted for further compromise. To prevent such misuse, data much be encrypted such a way that it does not reveal its original content other than to authorized users.

2.3 Data Sharing

One of the most widely used solutions to protect the data is encryption. In public key encryption which is one of the most widely used encryption scheme, a user encrypts data before uploading it to the cloud and decrypts the data after it is downloaded. However, such traditional encryption techniques do not support multi-party collaboration without re-distributing keys that are used for the encryption which creates a serious key management problem. In addition, as many modern applications produce big data sets that are large in size, it is often inefficient to download and upload the whole dataset.

The recent advancement of Attribute-Based Encryption (ABE) provides a solution to the issues associated with traditional encryption technique by supporting more fine-grained access control mechanism. The offer of such fine-grained access control is better suited when dealing with big data that are accessed by multiple users.

2.4 Access Policy Expression

Traditional public-key encryption requires that decryption to be done by one particular user and does not allow more complex access control policies. ABE relaxes such limitation by allowing decryption to be decided by an access policy tree. However, up until recently, the predicate expression used to define access polices were limited to be monotone which consists of AND, OR, or threshold gates. This does not allow the representation of negative constraints which raises a problem in scenarios where conflicts of interest naturally arise. For example, if Bob is not allowed to see a sensitive document shared by his colleges Alice, Sara and Kevin, it is more intuitive to define an access policy with ‘NOT Bob allowed’ than ‘allow Alice’ AND ‘allow Sara’ AND ‘allow Kevin’. The CP-ABE we apply has the strength offering the NOT gate in the predicate hence the name non monotonic.

3. BACKGROUND

Our work utilizes non-monotonic Ciphertext-Policy Attribute Based Encryption (CP-ABE) scheme originally proposed by Yamada et al. [12]. The non-monotonic scheme can be summarised as following four algorithms

- **Setup** (λ)**.** It chooses a bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^{\lambda}$ with $g \in \mathbb{G}$. Then it chooses random exponents $\alpha, \beta \in \mathbb{Z}_p$ and $H, U, V, W \in \mathbb{G}$. Set $V' = U^b$. It generates a public key **mpk** and a master secrete key **msk. mpk** = ($g, H, U, V, V', W, e(g,g)^{\alpha}$ and **msk** = (α, β)

- **Encrypt(mpk, M,** T**).** The encryption algorithm encrypts a message M under a non-monotonic access structure T over a set of attributes that are associated with a linear secret sharing scheme (L, π) using the master public key **mpk**. L is an $l \times m$ matrix. To produce a cipertext C, first it picks a random $\mathbf{s} = (s, s_1, \ldots. s_m) \in \mathbb{Z}_p^m$, computes share of s for $\pi(i)$ by λ_i= <$\mathbf{L}_i \cdot \mathbf{s}$ > for $i = 1, \ldots\ l$, and then computes C_0= M$\cdot e(g,g)^{\alpha \cdot s}$, C_1=g^s. It also computes $(C_{i,1}, C_{i,2}, C_{i,3})$ for every $i = 1, \ldots\ l$ as follows.
$$\begin{cases} C_{i,1} = W^{\lambda_i} V^{t_i}, \quad C_{i,2} = (U^{x_i} H)^{-t_i}, C_{i,3} = g^{t_i} \text{ if } \pi(i) = x_i \\ C_{i,1} = W^{\lambda_i} (V')^{t_i}, \quad C_{i,2} = (U^{x_i} H)^{-t_i}, C_{i,3} = g^{t_i}\ if\ \pi(i) = x'_i \end{cases}$$
where $t_i \in \mathbb{Z}_p$. It outputs the ciphertext $C = (C_0, C_1, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i \in [l]})$.

- **KeyGen(msk, mpk,** w**).** The key generation algorithm takes a set of attribute $w = \{x_1, \ldots.\ x_k\} \subset \mathbb{Z}_p$ and outputs a key that can identify the set (of attributes). It first chooses random r, $r_1, \ldots. r_k \in \mathbb{Z}_p$ and $r'_1, \ldots. r'_k \in \mathbb{Z}_p$ such that $r'_1 + \cdots + r'_k = r$. The private key **sk** is generated as;
$$\mathbf{sk} = \left(D_1 = g^{\alpha} W^{r}, D_2 = g^{r}, \begin{Bmatrix} k_{i,1} = V^{-r}(U^{w_i}H)^{r^i}, k_{i,2} = g^{r_i} \\ k\prime_{i,1} = (U^{bw_i}H^{b})^{r\prime^i}, k\prime_{i,2} = g^{br'_i} \end{Bmatrix}_{i \in [k]} \right).$$

- **Decrypt(mpk, C,** w**, sk)**: We assume that access structure T is satisfied y the attribute set w which means decryption is possible. Let I = $\{i |\ \pi(i) \in w'\}$. Because $w'$is authorized in the access tree T, the recipient can efficiently computer the reconstruction coefficients $\{(i, \mu_i)\}_{i \in I}$=Recon $_{L,\pi}(w')$ such that $\sum_{i \in I} \mu_i \lambda_i = s$. Then it parses $C = (C_0, C_1, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i \in [l]})$, $sk = (D_1, D_2, \{K_{i,1}, K_{i,2}, K'_{i,1}, K'_{i,2}\}_{i \in [k]})$ and computes $e(g,g)^{r \cdot \lambda_i}$ for each $i \in I$ as follows;
$$\begin{cases} e(C_{i,1}, D_2) \cdot e(C_{i,2}, K_{\tau,2}) \rightarrow e(g,W)^{r\lambda_i}\ if\ \pi(i) = x_i \\ e(C_{i,1}, D_2) \cdot \prod_{j \in [k]} (e(C_{i,3}, K'_{j,1}) \cdot e(C_{i,2}, K'_{j,2}))^{\frac{1}{x_i - w_j}} = e(g,W)^{r\lambda_i},\ \ if\ \pi(i) = x'_i \end{cases}$$
Here τ indicates the index such that w_τ= x_i. Such τ exists if $i \in I$ and $\pi(i)$ is non-negated attribute. With all that, it computes $e(C_1, D_2) \cdot \prod_{i \in I} (e(g,W)^{r\lambda_i})^{-\mu_i}$ = $e(g^s, g^{\alpha}) e(g,W)^{sr} e(g,W)^{-r \sum_{i \in I} \mu_i \lambda_i}$=$e(g,g)^{\alpha \cdot \beta}$. Finally it recovers the message by $c_0 / e(g,g)^{s\alpha}$ = M.

4. System Architecture

4.1 System Overview

Our client application is based on the model proposed in [8]. The main purpose of the client application is to provide an encryption scheme with flexible access policies. Figure 1 illustrates the overview of our proposed client application.

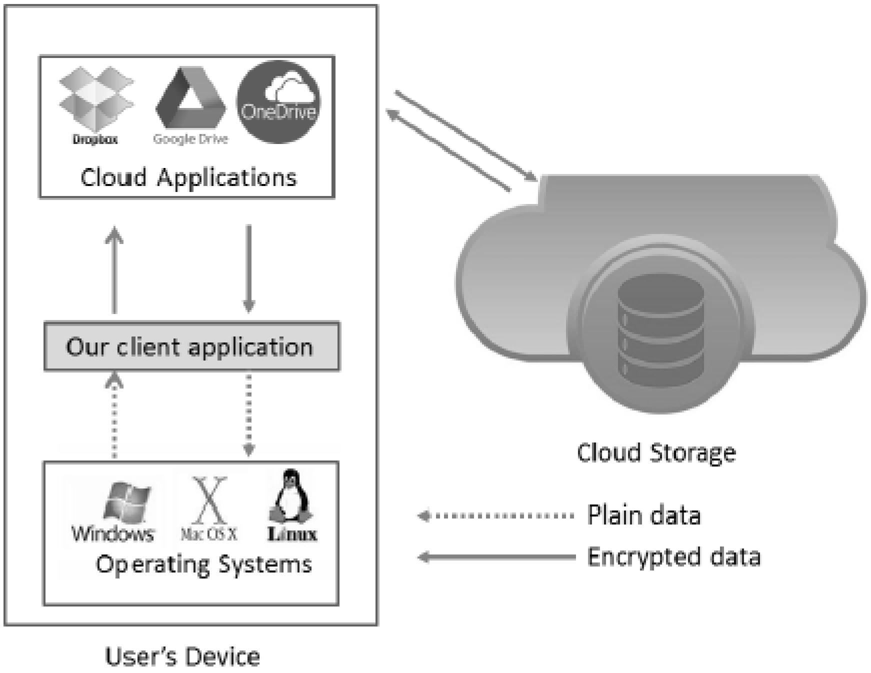


Figure 1: system overview

Our proposed client application sits between the operating system (e.g., Windows, Linux) and cloud storage application (e.g., DropBox, Google Drive, Microsoft OneDrive) installed in user's device (e.g., PCs, laptop, tables, or smartphones). Our client application provides an extra layer of protection to safeguard user's data and support attribute-based access control mechanism to support data sharing. Our client application can be seamlessly integrated with any cloud applications which we presume that it is pluggable from one application over the other.

Our application interacts with users. Here, the term users include the authorized devices, such as PCs, laptop, tables, or smartphones, laptop owned by individuals. Users can be categorised as encryptors and recipients. An encryptor refers to a type of user who owns the data and has full control of deciding whom it wants to share the data with. The encryptor can specify a set of access policies over the data to enforce access rules to control the access permission. The recipient refers to a type of user who wants to share the data with the encryptor. The recipient must provide a right set of attributes to proof that he/she satisfies the access rules imposed on the shared data to pass the permit.

Users contact administrator which can just be another user. The major role of the administrator is to generate master public and private key pairs under non-monotonic CP-ABE scheme. The administrator publishes the public keys. It also generates private keys based on user's attributes for all users. Once private keys are generated, they are distributed to corresponding users. We do not cover the details of private key distribution in this paper other than assuming that the private keys are delivered using a secure channel such as IPSec or SSL.

The cloud storage is a remote storage facility and is typically provided by cloud storage providers. The user can use cloud storage applications to store files in the cloud to share with others. Cloud storage providers may support security mechanisms to protect user's data from potential data loss and from unauthorized access. However, the access control mechanism provided by our client application does not depend on the underlying supports and does not share information regarding the encryption and decryption strategies and parameters.

4.2 System Operations

Figure 2 describes the details of the operations in terms of message exchange among various system components.

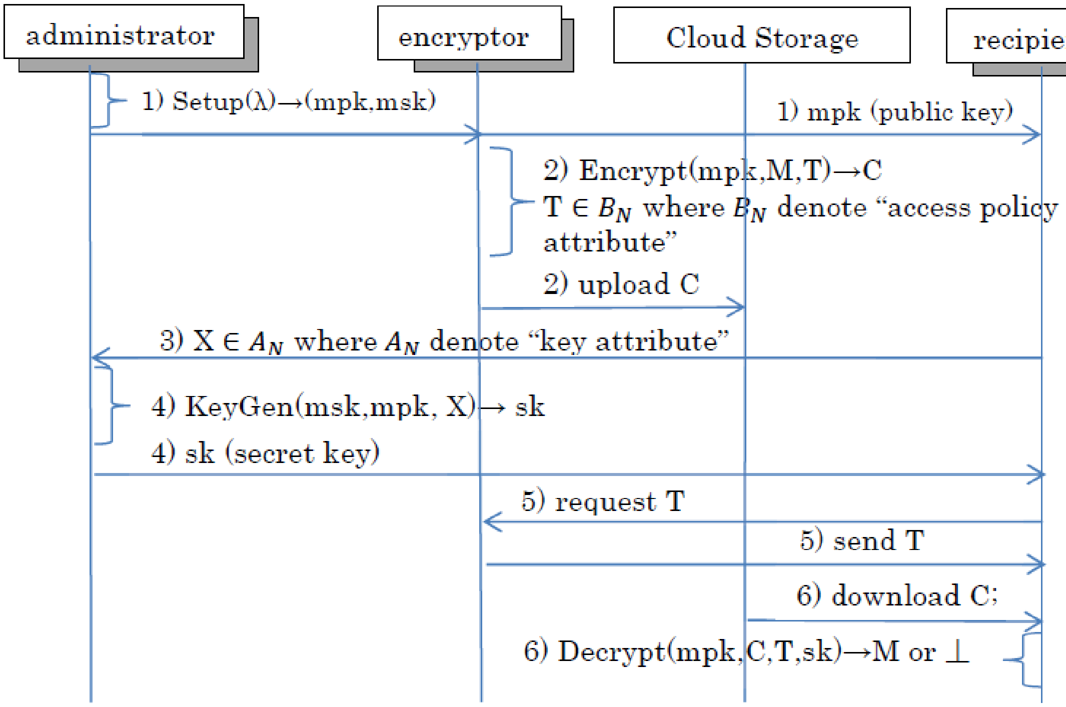


Figure 2: System operations

Note that our client application runs on the administrator, encryptor and the recipient as depicted by a gray box behind them and assists the following operations.

1) The operation here is a set up by an administrator. The client application running on the administrator takes a security parameter λ, in our case, it is a random number generated by a random number generator provided by a java cryptographic package our client application utilises. The result of the setup is a key-pair, one for a public key mpk and a corresponding private key msk.

2) In the meantime, a user acting as an encryptor wish to upload a file (M) to a cloud storage. The encryptor uses the client application running in the encrytor's device to encrypt the file. At this stage, the encryptor also defines an access policy (T) . A typical example of an access policy tree is depicted in Figure 3.

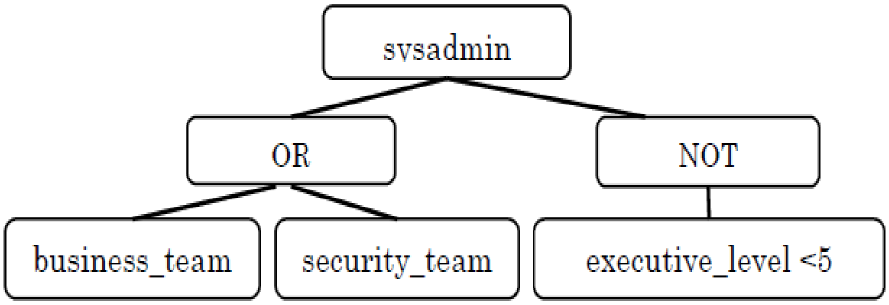


Figure 3: an example of an access policy tree

which can be defined as a simple Boolean formula;

Y = (sysadmin AND ("business_team" OR "security_team) AND
NOT (executive_level <5))

The access policy is associated with the encrypted data then uploaded to the cloud application which in turn subsequently sends the data to the cloud storage over the Internet.

3) Now let's assume that there is a recipient who wants to download a portion of the data uploaded by the encryptor to share. To proceed, the recipient first needs to send his/her attributes (e.g., job title, department, and executive level) to an administrator. Let's call the recipient's attribute X which can be defined as;

X = (job_title = sysadmin, department = security_team, executive_level = 7)

4) The administrator generates a private key sk according to the attributes then sends it to the corresponding recipient.

5) In the meantime, the recipient also sends a request to send an access policy T that is associated with the data the recipient wants to share with the encryptor.

6) Using the private key sk and access policy T received from the administrator and the encryptor, the recipient downloads the data and attempts the decryption. If the attributes associated with recipient's private key satisfies with the predicate defined in the access policy, the data is decrypted M. Nothing is returned $\perp$ if the key does not satisfy.

5. SYSTEM IMPLEMENTATION

5.1 Libraries

The CP-ABE scheme we use in our client application is a class of pairing based cryptography. It applies each encryptor's access rules as the attribute in the access policy tree. The public and private key pair is related to the attributes while the ciphertext is associated with an access policy. Our implementation is split into two packages:

cpabe: is a package that implements the CP-ABE scheme introduced by [12].

App: is a package that implements the higher level functions such as user interface, interpreting access policy and store and retrieve the data from the cloud.

The cpabe package uses java based pairing cryptography JPBC library [13]. JPBC is a java based open library that supports cryptographic methods. JPBC is a port of the Pairing-Based Cryptography Library (PBC) developed by Ben Lynn [14] to perform the mathematical operations underlying pairing-based cryptosystems directly in Java. Our current client application integrates with Aamzon S3 which represents a cloud application that runs along with our client application. We use the official Amazon Java SDK for API 1.11.58 [17] to integrate Amazon S3 into our client using Java.

5.2 Essential Algorithms

Attribute/Access Policy Parser: In CP-ABE scheme, each user's private key is associated with a set of attributes which represent their capabilities. A ciphertext is encrypted such that only users whose attributes satisfy a certain policy can decrypt. To use the scheme, we first need a scheme to define attributes and a parse an access policy which can understand them. CP-ABE scheme we implement basically support any Boolean formula, but interpreting unstructured Boolean formula and computing parameters using Linear Secret Sharing (LSS) scheme [18]. To parse an access policy, we allow encryptor to input an access policy under the following rules;

1. On a line only AND and NOT gate can be used
2. Split a line means OR gate

```
"Att1" AND "Att2" AND NOT "Att3"
"Att1" AND "Att5"
```

Theses inputs enforce user to input an access policy as a disjunctive normal form and allows App package to parse the policy more efficiently.

System initiation: the set up phase is done by administrator without the user having to take any action. Under the hood, administrator runs the cpabe package to generate a public key and a master secret key. The input function and the outputs are defined as following.

```
INPUT: cpabe.setup()
OUTPUT: master_key pub_key
```

After running cpabe.setup() function, the administrator obtains a master_key to be used to produce private keys for recipients. The public key pub_key can be shared in the Cloud or sent to any users.

Secrete Key Generation: administrator uses this function to create private keys for all recipients who wishes to share the data encrypted by an encryptor. The function keygen() integrates the input attributes (i.e., each recipient's capabilities) in the private key generation using the public key and the master secret key generated in the setup() phase. The output is a private key corresponding to a recipient. Attributes can come in a natural language but special characters cannot be included in the attributes.

```
INPUT: Cpabe.keygen(pub_key, master_key, attr)
OUTPUT: pri_key
```

Let’s assume that there are two recipients Sara and Kevin at a company and the administrator wants to produce private keys. The private keys include the both hires’ capabilities so that not all company documents can be decrypted by them. Sara’s capability is defined as her job title is a sysadmin at IT department. Her office number is at 1431 and she was hired last year - let’s say the date is denoted as yearX. In contrast, Kevin’s capability is defined as his job title is a business staff at strategy team. He has the executive level 7 and sits in the office number 2362. He was hired 5 years ago – let’s say the date is denoted as yearY. Sara’s and Kevin’s respective input parameters for the function keygen() follows.

```
Sara’s private key generation:
INPUT: Cpabe.keygen(pub_key, master_key, “Sara sysadmin it_department
office_1432 hire_yearX)
OUTPUT: sara_pri_key

Kevin’s private key generation:
INPUT: Cpabe.keygen(pub_key, master_key, “Kevin business strategy_tem
executive_level_7 office_2362 hire_yearY)
OUTPUT: Kevin_pri_key
```

The keygen() function allows some attributes are assigned a value while others a key simply “has” without further qualification. The date command can be used to help use the current time as an attribute value.

Encryption: encryptor uses this function to encrypt a file. Now assume that a staff member Bob in the company wants to encrypt a sensitive document. Bob only wish to share it with; someone who works in the sysadmin role in the security tem, or someone who is a business staff and either in the audit group or strategy team. Bob only needs the public key then can use the cpabe.enc to encrypt it with the access policy.

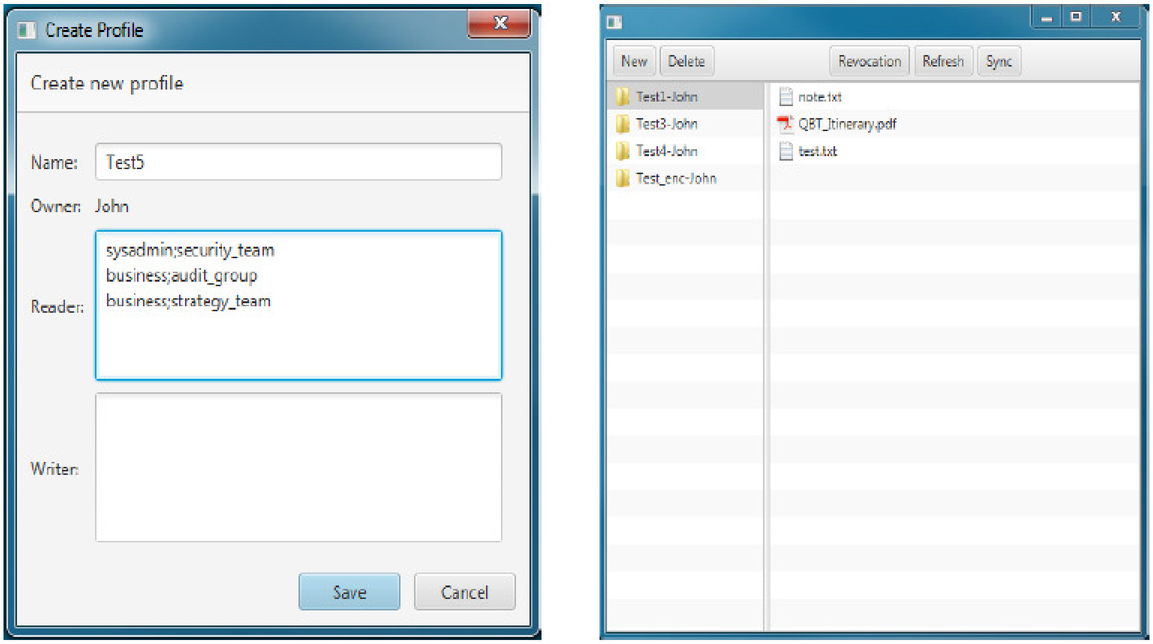
```
INPUT: Cpabe.enc(pub_key, security_report.pdf, “(sysadmin AND security_team)
OR (business AND audit_group) OR (business AND strategy_team)”
OUTPUT: security_report.pdf.cpabe
```

Decryption: the recipients use this function to decrypt the file. Kevin can successfully decrypt the encrypted security report using his private key associated with his capabilities which satisfy the access policy associated with the encrypted document. Sara won’t because the attributes of Sara’s key does not satisfy the access policy.

```
INPUT: Cpabe.dec (pub_key, kevin_priv_key, security_report.pdf.cpabe)
OUTPUT: security_report.pdf
```

5.3 User Interface

In this section, we demonstrate a set of GUI screens we developed for our client desktop demonstrator which allows the users to manage and control access control mechanism based on the non-monotonic CB-ABE.



Name: Test5

Owner: John

Reader:
sysadmin;security_team
business;audit_group
business;strategy_team

Writer:

Test1-John
Test3-John
Test4-John
Test_enc-John

note.txt
QBT_Itinerary.pdf
test.txt

(a) Setting a profile for a secure folder

(b) A secure folder for Enc/Decryption

Figure 4: client application GUI user interface

Figure 4 (a) shows the user interface screen that allows the encryptor to create a secure folder by setting an access policy that will commonly apply to the files in the folder. Reader box takes as input an access policy. For the simplicity, we use " ; " to denote AND gate and " ` " to denote NOT gate. Splitting line means OR gate. A user can create this secure folder in the Cloud (Amazon S3) by clicking the "Save" button.

Figure 4 (b) shows the screen that a list of secure folders and files in the selected folder which is synchronized with (Amazon S3). This window allows the recipients to perform an encryption/decryption operation. Encryption is performed by dragging a file in a file explorer or desktop to the right side of screen where a list of files are displayed. Then, the file is encrypted based on the access policy of the secure folder set when it created. Then, the encrypted file is uploaded and stored to the Cloud. For the decryption, a user simply clicks the file. As long as recipient's private key attributes matches with the access policy associated with the encrypted file, the file is downloaded from the Cloud and decrypted and stored the local computer.

While the processes, a public key and a corresponding recipient's private key are stored in an application folder in the local computer. Therefore, the Cloud cannot decrypt the files on their storage.

5.4 Implementation Results

We tested our implementation in the following system to measure and estimate the performance. For the elliptic curve, we use the type A (a_181_603) for the implementation.

Processor	Intel® Core™ i7-4600U CPU @ 2.10GHz 2.70 GHz
RAM	16 GB
OS	Windows 7 SP1 64-bit

For the fast encryption/decryption, we utilize the symmetric algorithm which is AES-256 to encrypt/decrypt files and hide the secret AES-128 key of each secure folder using CP-ABE scheme.

We estimate the times to execute each algorithm. The setup algorithm takes 94.8 ms on average. Also, KeyGen algorithm linearly increases for the number of attributes that a user has as shown in Table 1.

Table 1: performance of keygen depending on the number of user's attributes

# of attributes	2	4	6	8	10
KeyGen (ms)	230.2	425	616.3	816.3	1038.6

We also run encryption and decryption algorithm. The execution times are depending on both the number of attributes (non-negated attributes) and the number of attributes with not gate (negated attributes). We set a user to have five attributes but change the size of attributes for the simulations. The execution time of encryption increase almost evenly regardless of the type of attribute in an access policy. However, the execution time of decryption depends on the type of attributes in a policy. Negated attributes requires more decryption time.

# of non-negated attributes	1	2	3	4	5
Encryption (ms)	81	140.3	211.3	267.3	335.6
Decryption (ms)	38.4	60.7	82.9	102	130.4

# of negated attributes with 5 non-negated attributes	1	2	3	4
Encryption (ms)	401.6	436.4	503.8	577.8
Decryption (ms)	217	288.3	377.8	475.1

The time measures above does not depend on the size of a file since the file in the secure folder in the Cloud actually encrypted using AES-256 algorithm and CP-ABE was used only to encrypt/decrypt the AES secret key which is always 256 bits.

6. LESSONS LEARNED

ABE-based access control mechanism provides a great opportunity to better manage and share large datasets among multiple users. Though the promise is great, developing an ABE-based implementation strategy is often difficult and time consuming due to lack of language and tools support.

The time of our implementation, there were only two reference implementations of monotonic CP-ABE-scheme, one using a C language and the other with Java. If any developer wants to implement a CP-ABE in other languages such as .NET or Python, they are left with no other option but having to implement all functionalities from scratch. Currently there is no available non-monotonic CP-ABE but ours.

Another problem with current support is with platform dependency. Though Java is independent from platform in theory but in practice it is often platform dependent. For example, Java crypto API we used for our application did not guarantee the same operation on different platforms. We used AES CBC algorithm for encryption/decryption. Though the encryption successfully worked on both desktop application and Android, the decryption did not work on Android. As it turned out, the way padding was added for the AES CBC algorithms were different from the desktop application to Android. The strategy adopted for AES CBC padding for Android was not compatible with cpabe library we were using; as a result, decryption operation didn't work on the Android application.

7. RELATED WORK

Cryptography is one of the most promising ways to providing secure access control [7] [11]. The initial applications of cryptographically enforced access control [3] [6] were influential, but omitted some details which are required to practical implementations such as access policy update and key distributions. Recently, more practical cryptographically enforced access control mechanisms were proposed. They support a fine-grained access control using an advanced cryptographic primitives such as ABE and Predicate Encryption (PE).

Li et al. [9] introduced a novel implementation of Muti-Authority ABE to provide an access control on the personal health records on the cloud storage. Their scheme supports a user revocation by redistributing users' private keys and update an access policy in ciphertexts. GORAM and A-GORAM [10] are suggested to provide a secure data sharing on the untrusted cloud storage. They provide a fine-grained access control using PE and hiding even an access patterns from the server. Their systems allow updating policy by re-encrypt both access policy and their corresponding data. Wang et al. [11] introduced Sieve which can provide a fine-grained access control on the untrusted cloud storage. Sieve supports dynamic and efficient user revocation. It needs to re-encrypt policy, but reduce a burden of re-encrypting all data using key homomorphic encryption [4][5]. Garrison et al. [7] showed that role based access control (RBAC$_0$) can be cryptographically enforced. Their system uses a simple identity based encryption or a traditional RSA, but they showed that this can be extended to support more complicated access control models using advanced cryptographic primitives such as HIBE and ABE.

Although those systems well realize complicated and practical access model with advanced cryptographic primitives, supporting dynamic access control is still difficult. Even very recent work [7][11] requires both redistributing all valid users' keys and updating ciphertexts to revoke invalid users. Redistributing all users' keys needs secure communications with users and administrator and Updating ciphertexts requires that decryption and re-encryption. Therefore, they need intensive communications and computations.

Kim and Surya [8] suggested the system which provided a flexible revocation. Their system utilizes CP-ABE supports non-monotonic access structure suggested by Yamada et al. [12] and introduced a revocation algorithm without redistributing users' keys using negated attributes in access policies. However, their system is not system-wise implementation. It only implements and estimate the Yamada’s ABE schemes in C using PBC [14].

8. CONCLUSION

We proposed a client application that implements a non-monotonic CP-ABE. Our proposed client application sits in between the cloud application and cloud storages offering not only encryption and decryption processing to safeguard user’s sensitive data but also effective access control mechanism to allow multiple users sharing the data. We described the design consideration, system architecture and practical algorithms to apply an ABE-based scheme. We showed that it is possible to develop a ABE-based solution and can offer much flexible access control mechanisms for multiple users unlike public key infrastructure.

In a practical system, access policy is dynamic rather than static. Users can be added and deleted while the system is operating. Particularly, user revocation is difficult since invalid users must be successfully and immediately revoked from the system. We plan to implement a revocation mechanism [8] in our application in the near future.

ACKNOWLEDGEMENTS

The author would like to thank Jongkil Kim, CSIRO Australia, for providing the insights of the model this paper is based on and providing the implementation strategies. The author also thanks to Yao Leon Li for his assistance with the implementation and comments on the technical feedback

REFERENCES

[1] Operation Aurora. https://en.wikipedia.org/wiki/OperationAurora. Accessed: Dec. 2016.

[2] Tim Ring. Cloud computing hit by celebgate. 2015. https://www.scmagazineuk.com/cloud-computing-hit-by-celebgate/article/540448/. Accessed: Aug. 2017

[3] Selim G. Akl and Peter D. Taylor. Cryptographic solution to a problem of access control in a hierarchy. ACM Trans. Comput. Syst., 1(3):239–248, 1983.

[4] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, EUROCRYPT, volume 8441 of LNCS, pages 533–556. Springer, 2014.

[5] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan.Key homomorphic prfs and their applications. In Ran Canetti and Juan A. Garay, editors, CRYPTO, volume 8042 of LNCS, pages 410–428. Springer, 2013.

[6] E. Gudes. The design of a cryptography based secure file system. IEEE Transactions on Software Engineering, SE-6(5):411–420, Sept 1980.

[7] William C. Garrison III, Adam Shull, Steven Myers, and Adam J.Lee. On the practicality of cryptographically enforcing dynamic access control policies in the cloud. In IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016, pages 819–838. IEEE Computer Society, 2016.

[8] Jongkil Kim and Surya Nepal. A cryptographically enforced access control with a flexible user revocation on untrusted cloud storage. Data Science and Engineering, 1(3):149–160, 2016.

[9] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans. Parallel Distrib. Syst., 24(1):131–143, 2013.

[10] Matteo Maffei, Giulio Malavolta, Manuel Reinert, and Dominique Schr¨oder. Privacy and access control for outsourced personal records.In IEEE Symposium on Security and Privacy, pages 341–358. IEEE Computer Society, 2015.

[11] Frank Wang, James Mickens, Nickolai Zeldovich, and Vinod Vaikuntanathan.Sieve: Cryptographically enforced access control for user data in untrusted clouds. In NSDI, pages 611–626, Santa Clara, CA, March 2016. USENIX Association.

[12] Shota Yamada, Nuttapong Attrapadung, Goichiro Hanaoka, and Noboru Kunihiro. A framework and compact constructions for non-monotonic attribute-based encryption. In Hugo Krawczyk, editor, PKC, volume 8383 of LNCS, pages 275–292. Springer, 2014.

[13] Angelo De Caro and Vincenzo Iovino. jpbc: Java pairing based cryptography. In Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011, pages 850–855, Kerkyra, Corfu, Greece, June 28 - July 1, 2011.

[14] Lynn B (2007) On the implementation of pairing-based cryptosystems, Ph.D. thesis, Ph.D. thesis, Stanford University.

[15] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, ACM Conference on Computer and Communications Security, pages 89–98. ACM, 2006

[16] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In EUROCRYPT, pages 457–473, 2005.

[17] AWS SDK for Java 1.11.58. https://aws.amazone.com/releasenotes/2482881671102750. Accessed: Aug. 2017

[18] Amos Beimel. Secure Schemes for Secret Sharing and Key Distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1986.

AUTHOR

Julian is Associate Professor at Massey University, New Zealand. Her core research focuses are; cybersecurity (identity management, intrusion detection, trustworthy system, cloud storage, applied cryptography) and privacy protection techniques (data anonymization and Homomorphic encryption) for big data analytics. Prior to Massey, she worked at CSIRO, the premiere Australian government research agency, for 15 years. She is an active member of several database, cyber security and health data informatics research communities and has published more than 50 articles in the leading conferences and journal venues including IEEE and ACM. She obtained a BBus(Comp) from University of Western Sydney, Masters and PhD from University of Sydney, Australia.

INTENTIONAL BLANK

QUERY INVERSION TO FIND DATA PROVENANCE

Md. Salah Uddin[1], Dmitry V. Alexandrov[2], Armanur Rahman[3]

[1]National Research University Higher School of Economics (NRU HSE),
Faculty of Computer Science, School of Software Engineering,
Kochnovskiy Proezd 3,
125319, Moscow, Russian Federation
[2]National Research University Higher School of Economics (NRU HSE),
Faculty of Computer Science, School of Software Engineering,
Kochnovskiy Proezd 3,
125319, Moscow, Russian Federation;
Bauman Moscow State Technical University (Bauman MSTU),
Faculty of Engineering Business and Management,
Chair of Innovative Entrepreneurship, 2-ya Baumanskaya ul. 5,
105005, Moscow, Russian Federation
[3]BJIT Limited, Level-5, Road-2/C, Block-J, Baridhara, Dhaka-1212,
Bangladesh

ABSTRACT

Day by day data is increasing, and most of the data stored in a database after manual transformations and derivations. Scientists can facilitate data intensive applications to study and understand the behaviour of a complex system. In a data intensive application, a scientific model facilitates raw data products to produce new data products and that data is collected from various sources such as physical, geological, environmental, chemical and biological etc. Based on the generated output, it is important to have the ability of tracing an output data product back to its source values if that particular output seems to have an unexpected value. Data provenance helps scientists to investigate the origin of an unexpected value. In this paper our aim is to find a reason behind the unexpected value from a database using query inversion and we are going to propose some hypothesis to make an inverse query for complex aggregation function and multiple relationship (join, set operation) function.

KEYWORDS

Data Provenance, Structured Query Language (SQL), Query Processing, Query Inversion.

1. INTRODUCTION

In the database system domain: Data provenance, a kind of metadata, it is called lineage or pedigree which provides description of the origins of a piece of data and the process by which it arrived in a database. At present time, in different areas, such as e-science, data-warehousing etc. are required by origin of data to avoid unexpected value. To find data provenance the author has introduced different techniques such as GIS, VDL (Virtual Data Language), DB-Notes, BF05, SPG05a, SPG05b etc. But, the relation-ship between the data and its sources is very complex and

David C. Wyld et al. (Eds) : CNSA, DMDB, AIAP - 2018
pp. 17– 31, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80102

difficult to identify. So we want to use a kind of data provenance technology to automatically find out from where the unexpected data users were obtained from when users see the anomalous and suspicious data. To get better result we are going to introduce query inversion technique for some complex aggregation or multiple relationship function. Uses of the property by which some derivations can be inverted to find the input data supplied to them to derive the output data. Examples include, if an output of a database query Q applied on some source data D and given tuple is T then we want to understand which tuples in D contributed to get the output tuple T. A natural approach is to generate a new query Q0, determined by Q, D and T, such that when the query Q0 is applied to D, it generates a collection of input tuples that contributed to the output tuple T. In other words, we would like to identify the provenance by inverting the original query [10].

2. OVERVIEW OF EXISTING APPROACHES

Representing data provenance has two major approaches: annotation and inversion. Inversion approach is used to operate on an output data to find an input data. In area of data warehouses, Cui, Widom, and Wiener [27] first introduced the problem of relational database tracing data using query inversion. A disadvantage of this approach is that it cannot be used as sub-queries of normal relational queries and only partially benefit from the query optimization of the underlying Database Management System (DBMS). Another mechanism is called *where-provenance* [1]. Mainly, we use this technique for determining where annotations are propagated from. Boris Glavic et al. [28] also introduced a mechanism that is call Provenance Extension Relational Model (PERM). The PERM prototype supports provenance computation using Structured Query Language (SQL). But the disadvantage of this technique is that it does not work for correlated sub-queries.

One of the earlier definitions was given in the context of geographic information system (GIS). In GIS, data provenance is known as lineage which explicates the relationship among events and source data in generating the data product [1]. In the context of data-base systems, data provenance provides the description of how a data product is achieved through the transformation activities from its input data [2].

Annotation systems like DB-Notes [CTV05] and MONDRIAN [GKM05] is a common approach in life sciences [4] and it enables a user to annotate data item with an arbitrary number of notes which are normally propagated when annotated data is transformed.

VDL (Virtual Data Language) provides query and data definition facilities for the Chimera system [3] and it supports relational or object-oriented databases and SQL-like transformations.

The PReServ (Provenance Recording for Services) [GMM05, GJM+06a] approach uses a central provenance management service. It uses a common interface to enable different storage systems as a provenance store.

Trio is a recursive traversing lineage algorithm to achieve complete provenance of a par-ticular output tuple which introduces a new query language TriQL [5] to deal with uncer-tainty and lineage information.

3. QUERY INVERSION MECHANISM

A considerable research effort has been made by the database community to manage data provenance. Data provenance can be defined at different granularity levels such as relation or tuple. Furthermore, data provenance has been categorized based on the type of queries (e.g. why, where, how) it can satisfy. Different techniques have been proposed to generate data provenance in the context of a database system. But we are using query inversion techniques to find out data provenance from relational and complex database system easiest and fastest way.

To find specific data from databases we used some query. But we do not know behind this query at the execution time compiler need to process some tuples to generate output. Sometimes due to those tuples we get some unexpected results. We can not find which tuple is responsible for this unexpected value. But following our query inversion technique we have become able to find the problem showing the original data (see figure 1). Example, we wanted to see the sum of salary as per job category.

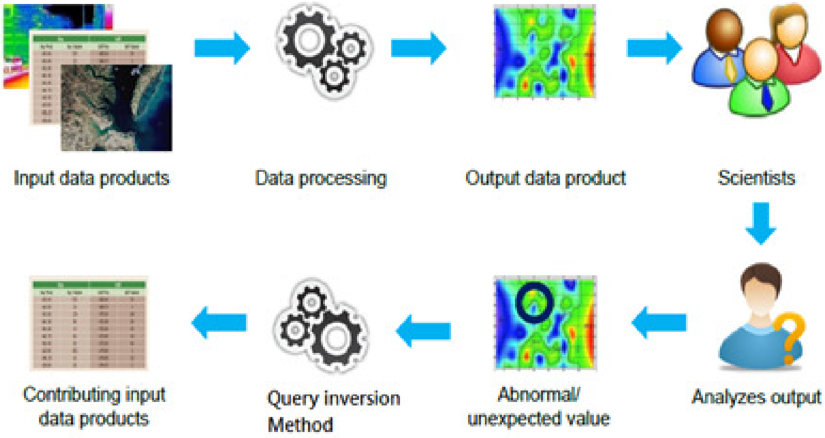
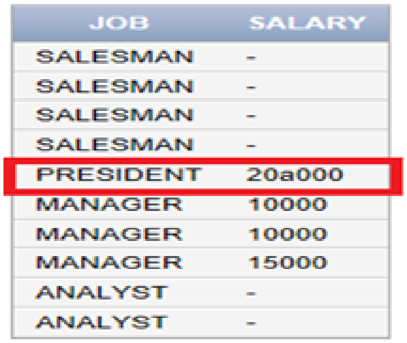


Figure 1. Data provenance technique overview.

Code: select job,sum(salary) from emp group by job

So we wrote above query to see the result, but it was showing an "**Invalid number**". Although our query is right but it was showing unexpected result because in our database there might be unwanted tuple. Due to that tuple this query is not working properly. Now we need to check all the values of the database to figure out the problem but at the present time the data is increasing and we have to process a huge amount of data daily. So it is not possible to check all the values and it is not time efficient. After analyzing the query we have developed an inverse query that is showing that unexpected tuple.

Code: select job,salary from emp where job in (select job from emp group by job) (see figure 2).



JOB	SALARY
SALESMAN	-
SALESMAN	-
SALESMAN	-
SALESMAN	-
PRESIDENT	20a000
MANAGER	10000
MANAGER	10000
MANAGER	15000
ANALYST	-
ANALYST	-

Figure 2. Showing the unexpected tuple using an inverse query.

3.1. Inverse Query hypothesis making technique

Inversion technique provides compact representation of provenance and this is the main advantage of this technique. But it is restricted to a certain class of multiple relationship queries and not universally applicable [26]. So in this section, our aim is to provide a hypothesis about possibility of developing of inverse query technique that can eliminate a limitation of multiple relationship queries.

When we need data manipulation we use some queries to get output but behind this output compiler need to process some tuples. After analyzing those tuples we have introduced some keys and extra tuples for our inverse query. Here we have described some complex aggregation and multiple relationship function in a generalized form of query and inverse query. We also have described the flow chart of the whole development procedure. In the flow chart we have to follow a life cycle to complete our inverse query (see figure 3, 4, 5). A Life cycle is a Black Arrow → Green Arrow → Red Arrow → Green Arrow.

3.1.1. Aggregation Functions

Oracle and other query languages support at least five aggregation functions such as min, max, count, sum, and avg. Aggregate functions are handled by adding parameter values to the group by list and adding the keyword condition for the aggregate column in the having clause instead of the where clause of the inverted query. For example if the general form query is:

Code: select ΔH1, f(ΔH2) from r group by ΔH1 having <predicate>

Then, the inverse query is as follows:

Code: select ΔH1, ΔH2 from r where ΔH1 in (select ΔH1 from r group by ΔH1 having <predicate>

Now, when adding keyword selections to the above keyword inverse query, any selections related to ΔH1 are added to the WHERE clause as before; however, selections relating to f(ΔH2) are added to a HAVING clause.
The relational algebraic translation of our above inverse query is:

$$\prod_{\Delta H1,\ \Delta H2,\ldots\Delta Hn}(\mho_{p\wedge contains((\Delta H1,\ \Delta H2,\ldots\ \Delta Hn),k)}(r)) \cap \prod_{\Delta H1,\ \Delta H2,\ldots\ \Delta Hn}(\mho_{p\wedge contains((\Delta H1,\ \Delta H2,\ldots\ \Delta Hn),k)>0}(r))$$

Here $\prod$=select clause, $\mho$=where clause for predicate, p=projection of attributes and r=table name.

Example of complex Aggregation function: Our generalized approach will work for all normal, complex and multiple relational aggregation functions.

Code: select sum(e.sal), count(t.deptno) from emp e, dept t

Now if we want to make an inverse query following our algorithm (see figure 3) for complex relationship, first of all we need to put *select* key then all attributes, *from* keyword and table name. After that according our algorithm, we need to put *where* keyword and first attribute without function attribute value, but here we can see that without function attribute there is no attribute, so we do not need to check the rest of the query after a table name. Our final inverse query will be as following:

Code: select e.sal, t.deptno from emp e, dept t

Following our generalized formula, it is possible to find data provenance for all simple, complex and multiple relationship aggregation functions. But for some correlated sub-queries our generalized formula does not work and it is the only limitation.

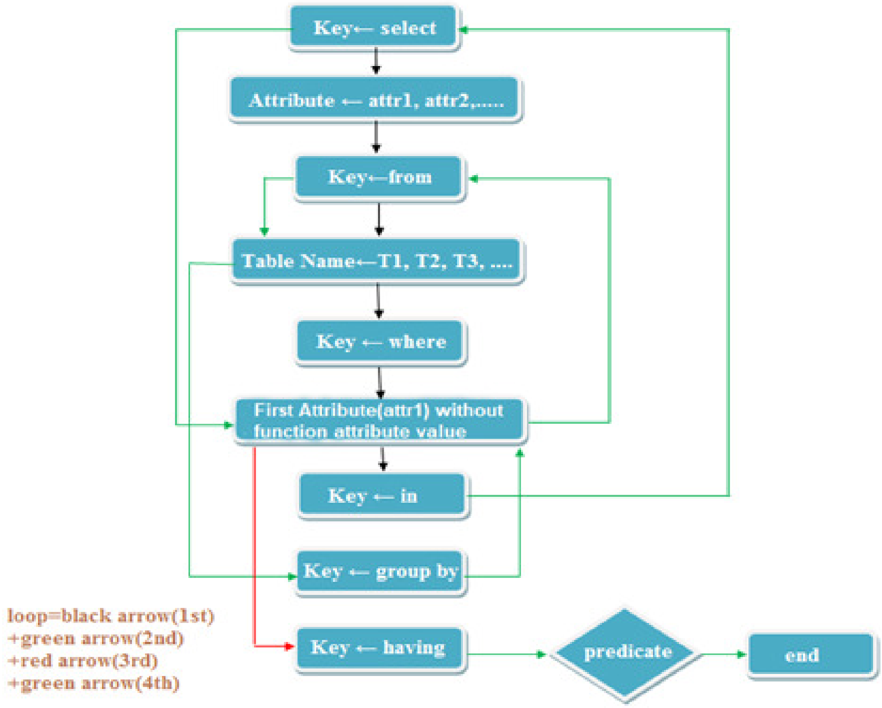


Figure 3. Inverse Query making procedure for aggregation functions.

3.1.2. Join Operation

The problem of join ordering is very restricted and at the same time it is a very complex one because it combines two or more tables in a relational database. At the compilation time if it has found any abnormal tuple it can not produce a result. So it is very difficult to find out the problem after searching multiple tables. For every tuple in the left input an output tuple must be produced for every tuple in the right input. A join operation can be implemented much more efficiently.

The approaches to handle the join operations are very similar to the ones for handling the aggregation function queries that are described in detail in the above section (3.1.1) but the difference is here we need to create a relationship between two or multiple tables. For example if the general form query is:

Code: select ΔH1, f(ΔH2) from r1 natural join r2 group by ΔH1 having <predicate>

Then, the inverse query is as follows:

Code: select ΔH1, ΔH2 from r1 natural join r2 where ΔH1 in (select ΔH1 from r1 natural join r2 group by ΔH1 having <predicate>

In this case we have to be more careful about the tuples in relationship to the different tables.

The relational algebraic translation of our above inverse query is:

$$\prod_{\Delta H1,\ \Delta H2,\ldots\Delta Hn}\left(\mho_{p\wedge contains((\Delta H1,\ \Delta H2,\ldots\ \Delta Hn),k)}(r\bowtie r)\right)\cap\prod_{\Delta H1,\ \Delta H2,\ldots\ \Delta Hn}\left(\mho_{p\wedge contains((\Delta H1,\ \Delta H2,\ldots\ \Delta Hn),k)>0}(r\bowtie r)\right)$$

Here ∏=select clause, Ʊ=where clause for predicate, p=projection of attributes, r=table name and ⋈=join operation name.

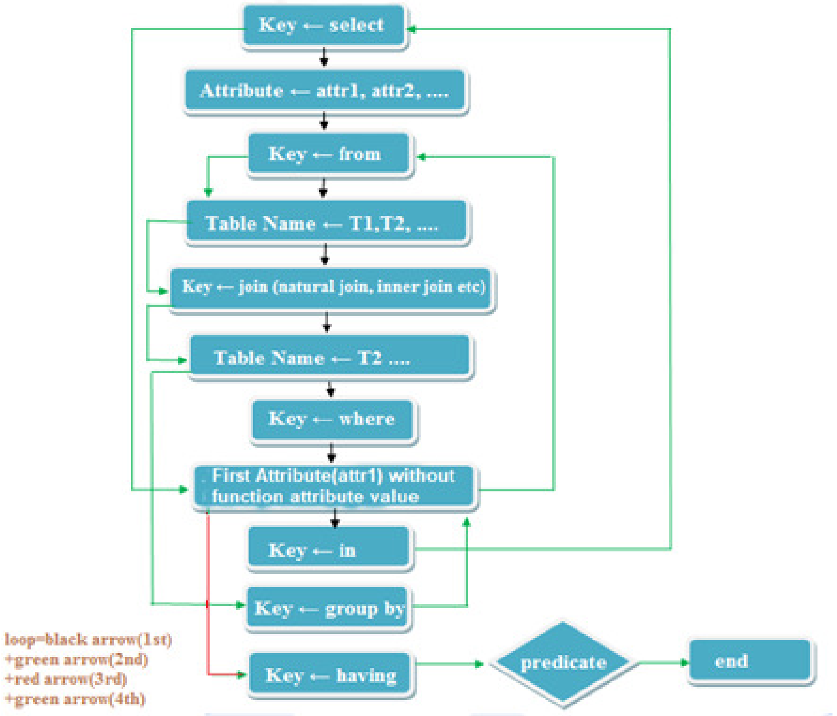


Figure 4. Inverse Query making procedure for join operation.

Example of complex Join operation: Our generalized approach will work for all normal, complex and multiple relational join operations.

Code: select count(cus.cust_first_name), sum(ord.order_total), sum(pro.quantity) from demo_customers cus natural join demo_orders ord natural join demo_order_items pro

Now if we want to make an inverse query following our algorithm (see figure 4) for multiple relationship, first of all we need to put *select* key then all attributes, *from* keyword and table name with *join* keyword. After that according our algorithm, we need to put *where* keyword and first attribute without function attribute value, but here we can see that without function attribute there is no attribute, so we do not need to check the rest of query after a table name. Our final inverse query will be as following:

Code: select cus.cust_first_name,ord.order_total,pro.quantity from demo_customers cus natural join demo_orders ord natural join demo_order_items pro

Following our generalized formula it is possible to find data provenance for all simple, complex and multiple relationship join operations. But for some correlated sub-queries our generalized formula does not work and it is the only limitation.

3.1.3. Set Operation

Set operation allows to be combined the results of multiple queries into a single result. Queries containing set operators are called compound queries. It includes union, intersect, minus operation in a database.

3.1.3.1 Intersect

A Set Intersection can be handled by individually inverting a query with respect to each keyword and then taking a join of the inverted queries on the common parameters. Let’s consider a general query:

Code: select ΔH1 from r where <predicate1> intersect select ΔH1 from r where <predicate2>

Then, the inverse query is as follows:

Code: select distinct ΔH1, ΔH2, ΔH3….. from r where <predicate1> and ΔH1 in (select ΔH1 from r where <predicate2>

The relational algebraic translation of our above inverse query is:

$$\prod_{\Delta H1,\ \Delta H2,\ldots\Delta Hn}(\sigma_{p\wedge contains((\Delta H1,\ \Delta H2,\ldots\ \Delta Hn)\cap k>0))}(r\bowtie r))\cap\prod_{\Delta H1,\ \Delta H2,\ldots\ \Delta Hn}(\sigma_{p\wedge contains((\Delta H1,\ \Delta H2,\ldots\ \Delta Hn),k)>0}(r\bowtie r))$$

3.1.3.2 Union

Handling the Union clause is complex mainly due to these reasons:

1. Each subquery involved in the Union may contain some of the keywords.

2. Each subquery in the Union may contain only a subset of the overall query parameters.

The approaches to handle the Union clause are very similar to the ones for handling the intersect queries and are described in detail in section (3.1.3.3). Let’s consider a general query:

Code: select ΔH1 from r where <predicate1> union select ΔH1 from r where <predicate2>

Then, the inverse query is as follows:

Code: select distinct ΔH1, ΔH2, ΔH3….. from r where <predicate1> or ΔH1 in (select ΔH1 from r where <predicate2>

The relational algebraic translation of our above inverse query is:

$$\prod_{\Delta H1,\ \Delta H2,\ldots\Delta Hn}(\sigma_{p\wedge contains((\Delta H1,\ \Delta H2,\ldots\ \Delta Hn)\cup k>0))}(r\bowtie r))\cap\prod_{\Delta H1,\ \Delta H2,\ldots\ \Delta Hn}(\sigma_{p\wedge contains((\Delta H1,\ \Delta H2,\ldots\ \Delta Hn),k)>0}(r\bowtie r))$$

3.1.3.3 Minus/Except

The Minus/Except clause/operator is used to combine two SELECT statements and returns rows from the first SELECT statement that are not returned by the second SELECT statement. This means except/minus returns only rows, which are not available in the second SELECT statement.

Let’s consider a general query:

Code: select ΔH1 from r where <predicate1> minus select ΔH1 from r where <predicate2>

Then, the inverse query is as follows:

Code: select distinct ΔH1, ΔH2, ΔH3….. from r where <predicate1> and ΔH1 not in (select ΔH1 from r where <predicate2>

The relational algebraic translation of our above inverse query is:

$$\prod_{\Delta H1,\ \Delta H2,\ldots\Delta Hn}(\sigma_{p\wedge contains((\Delta H1,\ \Delta H2,\ldots\ \Delta Hn)\cap k>0))}(r\bowtie r))\ \neg\cap\ \prod_{\Delta H1,\ \Delta H2,\ldots\ \Delta Hn}(\sigma_{p\wedge contains((\Delta H1,\ \Delta H2,\ldots\ \Delta Hn),k)>0}(r\bowtie r))$$

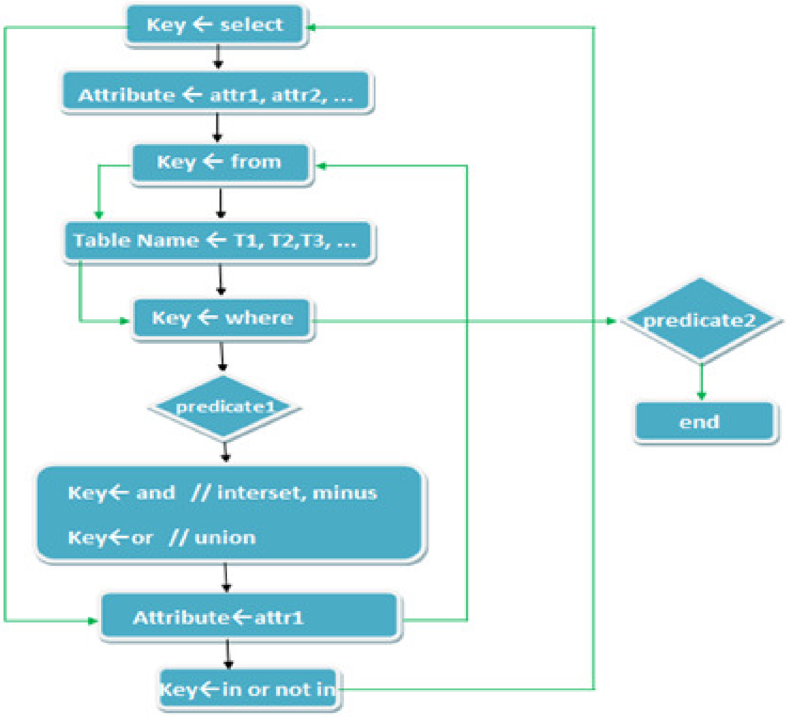


Figure 5. Inverse Query making procedure for set operation.

Example of complex Set operation: Our generalized approach will work for all normal, complex and multiple relational set operations.

Code: select course_id from section where semester='Fall' and year=2009 intersect select course_id from section where semester='Spring' and year=2010 union select course_id from teaches where semester='Fall' and year=2009

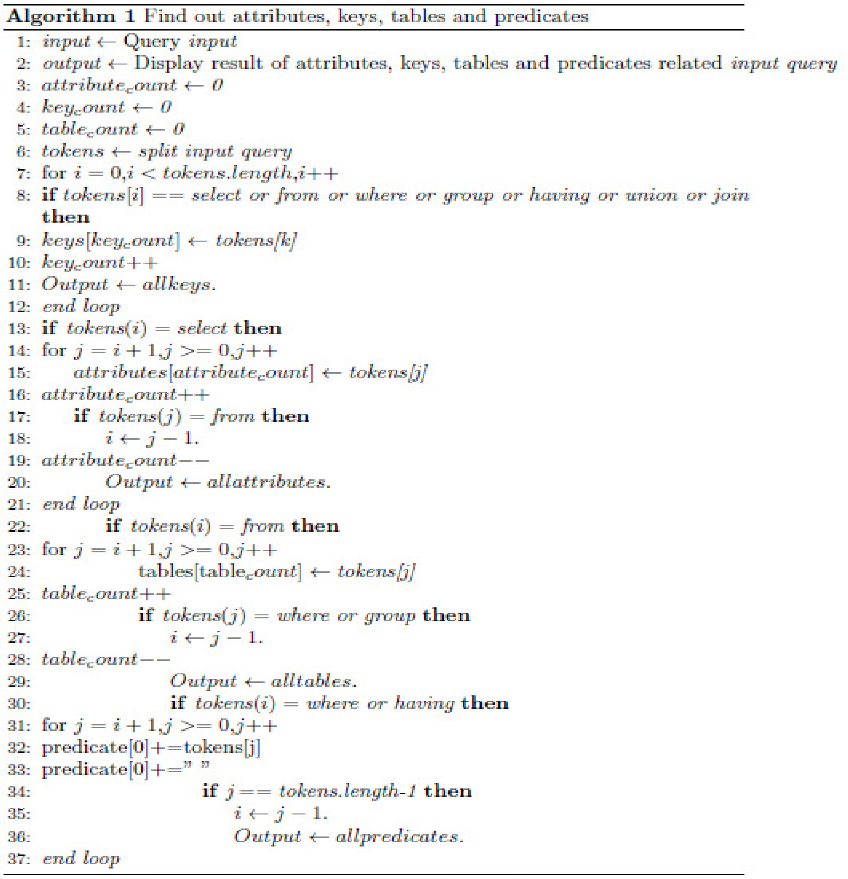
From above query we can see that there are two set operations: intersect and union. So following our algorithm (see figure 5), the inverse query will be:

Code: select course_id,semester,year from section where semester='Fall' and year=2009 and course_id in (select course_id from section where semester='Spring' and year=2010) or course_id in (select course_id from teaches where semester='Fall' and year=2009)

Following our generalized formula it is possible to find data provenance for all simple, complex and multiple relationship set operations. But for some correlated sub-queries our generalized formula does not work and it is the only limitation.

3.2. Prototype Development Algorithm:

First of all we have separated all keys, tables, attributes and predicates from the main query then we have checked the query to find it,s aggregation, set operation or join operation. Finally our prototype dynamically set those values following the sequence of the previous flow chart (see figure 3, 4, 5).



```
Algorithm 1 Find out attributes, keys, tables and predicates
input ← Query input
output ← Display result of attributes, keys, tables and predicates related input query
$attribute_count \leftarrow 0$
$key_count \leftarrow 0$
$table_count \leftarrow 0$
tokens ← split input query
for $i = 0, i < tokens.length, i{+}{+}$
if $tokens[i] ==$ select or from or where or group or having or union or join
then
$keys[key_count] \leftarrow tokens[k]$
$key_count{+}{+}$
Output ← allkeys.
end loop
if $tokens(i) =$ select then
for $j = i + 1, j >= 0, j{+}{+}$
    $attributes[attribute_count] \leftarrow tokens[j]$
$attribute_count{+}{+}$
    if $tokens(j) =$ from then
        $i \leftarrow j - 1.$
$attribute_count{-}{-}$
        Output ← allattributes.
end loop
        if $tokens(i) =$ from then
for $j = i + 1, j >= 0, j{+}{+}$
            $tables[table_count] \leftarrow tokens[j]$
$table_count{+}{+}$
            if $tokens(j) =$ where or group then
                $i \leftarrow j - 1.$
$table_count{-}{-}$
                Output ← alltables.
                if $tokens(i) =$ where or having then
for $j = i + 1, j >= 0, j{+}{+}$
predicate[0]+=tokens[j]
predicate[0]+=" "
                    if $j ==$ tokens.length-1 then
                        $i \leftarrow j - 1.$
                        Output ← allpredicates.
end loop
```

Figure 6. Algorithm for separating keys, attributes, tables and predicate.

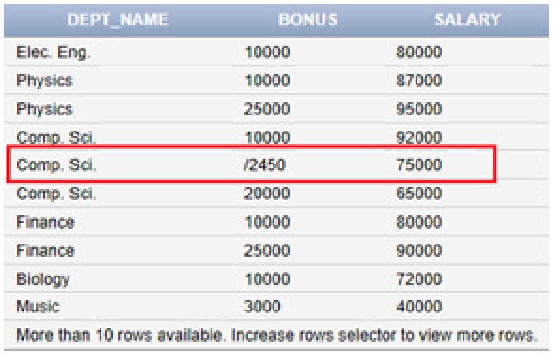
4. EXPERIMENTAL RESULT

We have developed a prototype that is providing an inverse query for any given query. After that we have checked this inverse query is right or wrong in Oracle Database XE 11.2. We have written a query to get total summation of salary and bonus information the specific department name category field.

Query: select dept_name,sum(bonus+salary) from instructor group by dept_name
Output: Invalid Number

After that, now we want to know how to get the above results which tuples are working and which one is responsible for an unexpected result. Then we use our inverse query:

Inverse Query: select dept_name,bonus,salary from instructor where dept_name in(select dept_name from instructor group by dept_name)



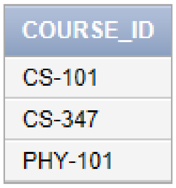
DEPT_NAME	BONUS	SALARY
Elec. Eng.	10000	80000
Physics	10000	87000
Physics	25000	95000
Comp. Sci.	10000	92000
Comp. Sci.	/2450	75000
Comp. Sci.	20000	65000
Finance	10000	80000
Finance	25000	90000
Biology	10000	72000
Music	3000	40000

More than 10 rows available. Increase rows selector to view more rows.

Figure 7. Output of inverse query for above aggregation function.

From our inverse query output **(see figure 7)** we can see that there is a tuple which contains an unexpected value. Due to this value our main query provides an unexpected result.
We tested for set operation and join query, we got accurate result to find data provenience.

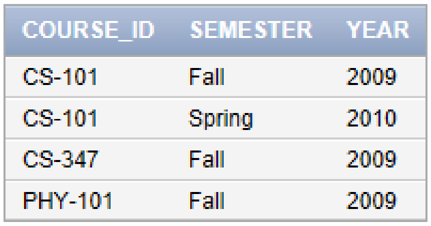
Query: select course_id from section where semester='Fall' and year=2009 intersect select course_id from section where semester='Spring' and year=2010 union select course_id from teaches where semester='Fall' and year=2009 minus select course_id from takes where semester='Fall' and year=2010 (see figure 8).



COURSE_ID
CS-101
CS-347
PHY-101

Figure 8. Output of above intersect query.

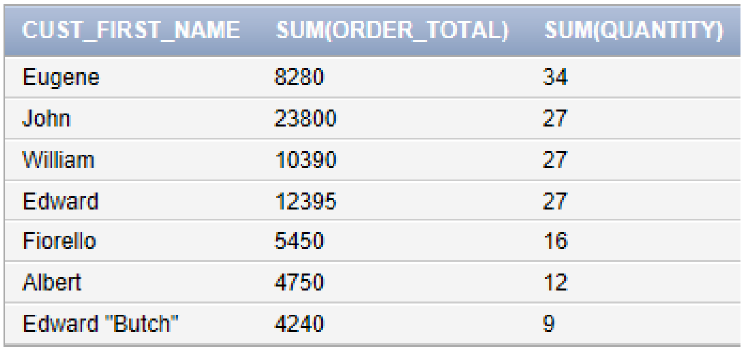
Inverse query: select course_id,semester,year from section where semester='Fall' and year=2009 and course_id in (select course_id from section where semester='Spring' and year=2010) or course_id in (select course_id from teaches where semester='Fall' and year=2009) and course_id not in (select course_id from takes where semester='Fall' and year=2010) (see figure 9).



COURSE_ID	SEMESTER	YEAR
CS-101	Fall	2009
CS-101	Spring	2010
CS-347	Fall	2009
PHY-101	Fall	2009

Figure 9. Output of above intersect inverse query.

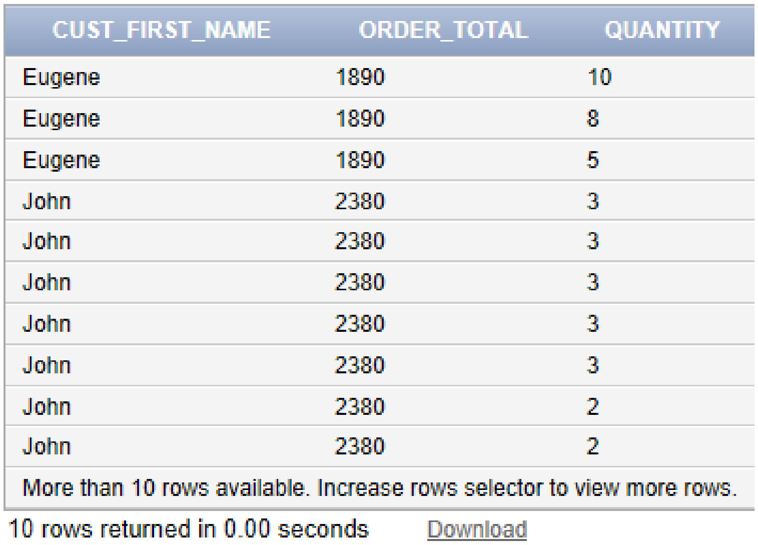
Query: select cust_first_name,sum(order_total),sum(quantity) from demo_customers natural join demo_orders natural join demo_order_items group by cust_first_name (see figure 10).



CUST_FIRST_NAME	SUM(ORDER_TOTAL)	SUM(QUANTITY)
Eugene	8280	34
John	23800	27
William	10390	27
Edward	12395	27
Fiorello	5450	16
Albert	4750	12
Edward "Butch"	4240	9

Figure 10. Output of above join query.

Inverse Query: select cust_first_name,order_total,quantity from demo_customers natural join demo_orders natural join demo_order_items where cust_first_name in(select cust_first_name from demo_customers natural join demo_orders natural join demo_order_items group by cust_first_name) (see figure 11).



CUST_FIRST_NAME	ORDER_TOTAL	QUANTITY
Eugene	1890	10
Eugene	1890	8
Eugene	1890	5
John	2380	3
John	2380	3
John	2380	3
John	2380	3
John	2380	3
John	2380	2
John	2380	2

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds Download

Figure 11. Output of above join inverse query.

Finally, we have become able to find the data provenance **(see figure 2, 7, 8, 9, 10, 11)** using our inverse query. Now if any user gets an abnormal or unexpected value, or they want to check behind their query, which tuples work, they can easily check by creating an inverse query.

4.1. Performance Evaluation:

To check performance of our new algorithm all experiments are performed on Intel core i3 machine with 4 GB ram and the size of our test database 10MB, 100MB and 500MB. For testing we have used three types of SQL query such as normal (Q1), complex (Q2) and multiple relationship (Q3) of four tables. To get execution time for each query we wrote “set statistics time on" before our query and at the end of our query we also added "set statistics time off". We evaluated execution time of each query for aggregation function (see table 1), join operation (see table 2) and set operation (see table 3). Analyzing the experimental results of aggregation function (see table 1), we can see that in most of cases there is a little time execution difference between the main query and the inverse one. If we compare small (10MB) and large (500MB) datasets, the execution time of our inverse queries is not increasing too much.

Table 1. The Execution time for each query of Aggregation function.

Query	**10MB**		**100MB**		**500MB**	
	Query	**Inverse query**	**Query**	**Inverse query**	**Query**	**Inverse query**
Q1	35 ms	36 ms	69 ms	76 ms	77 ms	94 ms
Q2	66 ms	69 ms	149 ms	157 ms	158 ms	169 ms
Q3	19 ms	321 ms	29 ms	575 ms	31 ms	581 ms

Table 2. The Execution time for each query of Join operation.

Query	**10MB**		**100MB**		**500MB**	
	Query	**Inverse query**	**Query**	**Inverse query**	**Query**	**Inverse query**
Q1	49 ms	70559 ms	69 ms	109513 ms	77 ms	250124 ms
Q2	52 ms	88015 ms	81 ms	191092 ms	98 ms	398029 ms
Q3	55 ms	90939 ms	92 ms	220306 ms	123 ms	502196 ms

Table 3. The Execution time for each query of Set operation.

Query	**10MB**		**100MB**		**500MB**	
	Query	**Inverse query**	**Query**	**Inverse query**	**Query**	**Inverse query**
Q1	3 ms	9598 ms	8 ms	15598 ms	11 ms	18400 ms
Q2	2 ms	8598 ms	9 ms	15101 ms	12 ms	18139 ms
Q3	3 ms	9321 ms	14 ms	17004 ms	19 ms	21409 ms

Execution time changes rather high from main query to inverse query for join and set operations. And if we compare the execution time of our inverse query for small (10MB) and large (500MB) datasets, the difference will be higher due to in this case a processor needs to process a huge dataset to provide the results.

5. Conclusion and Future work

Nowadays, provenance of data products is a widely-studied topic that attracts much attention of researchers. In this paper the main focus was to provide a guideline to find data provenance for unexpected values. To solve this problem we used an inverse query mechanism. Therefore, we showed that can easily find data provenance with the use of inverse queries. We proposed the generalized forms that work for all types of normal, complex and multiple relationship queries except nested query. We presented also the execution times of our inverse queries for small and large datasets. Finally, we found that the execution time is not high for large datasets comparing to small datasets, and in most of the cases our solution is rather fast. Thus this technique can be used in different applications. For example, we generate business reports using different applications. If we suspect any errors in these reports, we have to check the related source datasets manually. Now this problem has been solved by using the proposed technique, as we can easily find the possible error in rather efficient way.

Since the proposed technique does not work for sub-queries, but only works for normal, complex and multiple relationship functions, the sub-queries go in focus of future research. Thus we plan to solve the sub-queries problem by improving the technique and develop a web prototype so that any user could generate the inverse query related his or her main query.

ACKNOWLEDGEMENTS

We would like to thank all the colleagues at the School of Software Engineering of NRU HSE for their feedback and useful recommendations that contributed to bringing this paper to its final form.

REFERENCES

[1] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. In Proceedings of the International Conference on Database Theory, pages 316–330, 2001. (Cited on pages 2 and 21.)

[2] D. P. Lanter. Design of a lineage-based meta-data base for GIS.Cartography and Geographic Information Scence, 18(4):255–261, 1991.(Cited on pages 2 and 29.)

[3] Ian T. Foster, Jens-S. V¨ockler, Michael Wilde, and Yong Zhao. Chimera: A Vir-tual Data System for Representing, Querying, and Automating Data Derivation. In SSDBM'02: Proceedings of the 14th International Conference on Scientific and Statistical Database Management, pages 37–46, Washington, DC, USA, 2002. IEEE Computer Society.

[4] Laura Chiticariu, Wang-Chiew Tan, and Gaurav Vijayvargiya. DBNotes: a post-it system for relational databases based on provenance. In SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pages 942–944, New York, NY, USA, 2005. ACM Press.

[5] O. Benjelloun, A. D. Sarma, A. Halevy, and J. Widom. ULDBs: Databases with uncertainty and lineage. In Proceedings of the International Conference on Very Large Data Bases, pages 953–964, 2006. (Cited on page 22.)

[6] P. Agrawal, O. Benjelloun, A. D. Sarma, C. Hayworth, S. Nabar,T. Sugihara, and J. Widom. Trio: A system for data, uncertainty, and lineage. In Proceedings of the International Conference on Very Large DataBases, pages 1151–1154, 2006. (Cited on page 22.)

[7] M. K. Anand, S. Bowers, T. McPhillips, and B. Ludäscher. Efficient provenance storage over nested data collections. In Proceedings of the International Confe-rence on Extending Database Technology: Advances in Database Technology, pages 958–969. ACM, 2009. (Cited on page 20.)

[8] S. Davidson, , S. C. Boulakia, A. Eyal, B. Ludäscher, T. M. McPhillips, S. Bowers, M. K. Anand, and J. Freire. Provenance in scientific workflow systems. IEEE Data Engineering Bulletin, 30(4):44–50, 2007.(Cited on pages 15 and 17.)

[9] U. Park and J. Heidemann. Provenance in sensornet republishing. In Provenance and Annotation of Data and Processes, volume 5272 of LNCS, pages 280–292. Springer, 2008. (Cited on pages 26, 28, 37, 38, 111, 122,and 264.)

[10] M. R. Huq, P. M. G. Apers, and A. Wombacher. An Inference-based Framework to Manage Data Provenance in Geoscience Applications. Accepted in IEEE Transactions on Geoscience and Remote Sensing, Earlyaccess article DOI: 10.1109/TGRS.2013.2247769, IEEE Geoscience and Remote Sensing Society, 2013.

[11] C. Beeri, A. Eyal, S. Kamenkovich, and T. Milo. Querying business processes. In VLDB, 2006.

[12] O. Benjelloun, A. D. Sarma, A. Y. Halevy, and J. Widom. ULDBs: Databases with uncertainty and lineage. In VLDB, 2006.

[13] Grigoris Karvounarakis, Zachary G. Ives and Val Tannen: Querying Data Prove-nance. SIGMOD'10, June 6–11, 2010, Indianapolis, Indiana, USA.

[14] Boris Glavic, Klaus Dittrich: Data Provenance: A Categorization of Existing Ap-proaches. SNF Swiss National Science Foundation: NFS SESAM

[15] Qi Yang. Computation of chain queries in distributed database systems.In Proc. of the ACM SIGMOD Conf. on Management of Data, pages 348-355

[16] L. Becker and R. H. G uting. Rule-based optimization and query processing in an extensible geometric database system. ACM Trans. on Database Systems (to appear)

[17] P. Bernstein, E. Wong, C. Reeve, and J. Rothnie. Query processing in a system for distributed databases (sdd-1). ACM Trans. on Database Systems, 6(4):603-625

[18] Parag Agrawal, Omar Benjelloun, Anish Das Sarma, Chris Hayworth, Shubha Nabar, Tomoe Sugihara, and JenniferWidom. An Introduction to ULDBs and the Trio System. IEEE Data Engineering Bulletin, 29(1):5–16, 2006.

[19] Yingwei Cui, Jennifer Widom, and Janet L. Wiener. Tracing the lineage of view data in a warehousing environment. ACM Trans. Database Syst., 25(2):179–227, 2000.

[20] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A survey of data prove-nance in e-science. SIGMOD Rec., 34(3):31–36, 2005.

[21] Paul Groth, Sheng Jiang, Simon Miles, Steve Munroe, Victor Tan, Sofia Tsasa-kou, and Luc Moreau. An Architecture for Provenance Systems — Executive Summary.Technical report, University of Southampton, February 2006.

[22] Ian T. Foster, Jens-S. V¨ockler, Michael Wilde, and Yong Zhao. Chimera: A Vir-tual Data System for Representing, Querying, and Automating Data Derivation. In SSDBM '02: Proceedings of the 14th International Conference on Scientific and Statistical Database Management, pages 37–46, Washington, DC, USA, 2002. IEEE Computer Society.

[23] Dennis P. Groth. Information Provenance and the Knowledge Rediscovery Prob-lem. In IV, pages 345–351. IEEE Computer Society, 2004.

[24] P. Yue, Z. Sun, J. Gong, L. Di, and X. Lu. A provenance framework for web geo-processing workflows. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, pages 3811–3814. IEEE, 2011. (Cited on page 29.)

[25] M. R. Huq, A. Wombacher, A. Mileo. Data Provenance Inference in Logic Pro-gramming: Reducing Effort of Instance-driven Debugging.Technical Report TR-CTIT-13-11, Centre for Telematics and Information Technology, University of Twente, 2013.

[26] Bhagwat, L. Chiticariu, W. C. Tan, and G.Vijayvargiya, "An Annotation Management System for Relational Databases," in VLDB, 2004, pp. 900-911.

[27] Y. Cui, J. Widom, and J. Wiener. Tracing the Lineage of View Data in a Warehousing Environment. ACM Transactions on Database Systems (TODS), 25(2):179–227, 2000.

[28] Perm: Processing Provenance and Data on the same Data Model through Query Rewriting –Boris Glavic, Gustavo Alonso — 2009 — In ICDE ’09: Proceedings of the 25th International Conference on Data Engineering

AUTHORS

Md Salah Uddin is Master’s student of the program “System and Software Engineering” at NRU HSE, Moscow, Russian Federation.
Research interests: Cloud Computing and Security, Big Data, Solid State Drivers, Databases and Neural networks.

Dmitry V. Alexandrov is Professor of the School of Software Engineering at NRU HSE, Moscow, Russian Federation; Professor of the Chair of Innovative Entrepreneurship at Bauman MSTU, Moscow, Russian Federation.
Research interests: Artificial Intelligence, Multi Agent Systems, Data Analysis, Databases, Mobile Applications Development

Armanur Rahman is Junior Software Engineer at BJIT Limited, Dhaka, Bangladesh. He has awarded his bachelor degree at East West University, Bangladesh.
Research interests: Data Mining, Databases and Neural networks.

INTENTIONAL BLANK

AUTOMATING AUTOMATION: MASTER MENTORING PROCESS

Martin Ciupa, Nicole Tedesco and Mostafa Ghobadi

calvIO Inc., Webster, NY, USA

ABSTRACT

This paper describes an innovative computer system framework supporting robot automation programming, by a gesture controlled "show and tell" process, whereby human experts describe their goals of a process to be learned, and demonstrate via positional sensors(such as Microsoft Kinect or Leap Motion) the actions necessary to achieve those goals. These inputs are collected and optimized by the robot mentoring process, interpreted back to the human expert for confirmation feedback and/or subsequent fine tuning. This framework, is a pedagogical model, modelled on a professional human process needed to mentor others. We have applied this concept in an innovative way to mentor robots with gesture control and machine learning.

KEYWORDS

AI, Automation; Robotics; Deep Learning; Really Useful Machine Learning.

1. INTRODUCTION

As technology advances and novel tools in AI and IoT emerge, paradigm shifts in industrial settings becomes more and more imperative. Accelerating the manufacturing process has been supported via different practices such as rapid prototyping technology as well as facilitated setup and intuitive programming interface for conventional manufacturing process. Offline programming of the robots within an intuitive 3D modeling and simulation environment has attracted lots of attentions in the recent decade as a solution to speed up the "order-to-delivery" process [1]. In this regard, this paper proposes an organized procedure for a smart master mentoring process aiming at "automating the automation" using AI tools such as Deep Learning and rule extraction methods such as Really Useful Machine Learning (RUML^{SM}) [2] a patent pending process [3].

Heuristic motion training approaches such as robot learning by demonstration and interactive human robot interface for path planning is one of the research areas that has properly been investigated before[4], [5].To this end, different conventional tools such as Deep Learning [6], [7],reinforcement learning [8], and hidden Markov model[9], and probabilistic segmentation of movement primitives [10]can be used. Moreover, heuristic mathematical models and methodologies in probabilistic filtering and inference such as feedback-based information roadmap[11], filtering in presence of partial observation [12],[13],and adaptive uncertainty propagation for coupled multidisciplinary systems [14]have the potential to improve the robustness of the proposed framework against the uncertainties in dynamical environment.

Such motion training approaches can potentially profit the automation industry. However, a holistic paradigm shift in automation industry in order to expedite the order-to-delivery process is

David C. Wyld et al. (Eds) : CNSA, DMDB, AIAP - 2018
pp. 33– 40, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80103

a chain yet with several lost rings that need to be addressed. The new paradigm must provide solutions for different domains, applications, goals and tasks. It also needs to provide the rationalization behind the provided solutions such that one can evaluate its the efficiency and effectiveness of the solutions. Such a rationalization can extend the applicability of the gained expertise into new domains by extrapolating the rationale to similar tasks in a new domain. The master mentoring process not only provides a step-by-step training framework from beginner to expert stages but also incorporates new approaches such as learning by demonstration and smart motion planning to facilitate the programming of the robots. Furthermore, it utilizes RUML^{SM} Learning [3] to rationalize the acquired knowledge. The proposed training framework, is a pedagogical model, modelled on a professional human process needed to mentor others. We have applied this concept in an innovative way to mentor robots with gesture control and machine learning.

2. METHOD

Figure 1 illustrates show & tell process for a master mentoring robot training process; the description is of the major embodiments in the figure. First an abstract / generic description of the stages is described, then an example is given as to its application to the "Show and Tell Master Mentoring Robot Training Process" specifics.

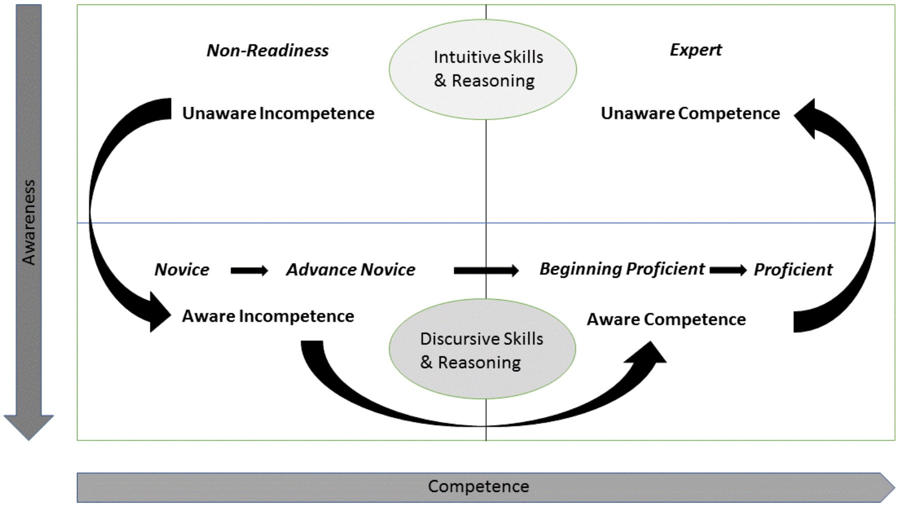


Figure 1. Master Mentoring Process is a Show & Tell Process proposed in this paper to automate the automation and to train the robots.

See Figure 2 for illustration example of non-readiness stage, where we are at an unaware incompetence situation. The human expert user (hereafter referred to as the Trainer) specifies the domain of the learning application and the goals of the process being trained. For example: Domain - Six Axis Robot Pick and Place; Goal -To Pick from Conveyor a Specified Tool and Place it into a Storage Container. The Robot Mentoring system has software to recognize the match of the Domain to an Application Library set. In addition, the specification of the Goal first seeks to determine if a matching Application definition pre-exists to be used or sets up a new Application definition based on either a) Drop down Menu driven by the context of the Domain/Application specification and Goals, or b) Natural Language Processing (NLP) parsing of the Domain/Application specification and Goals.

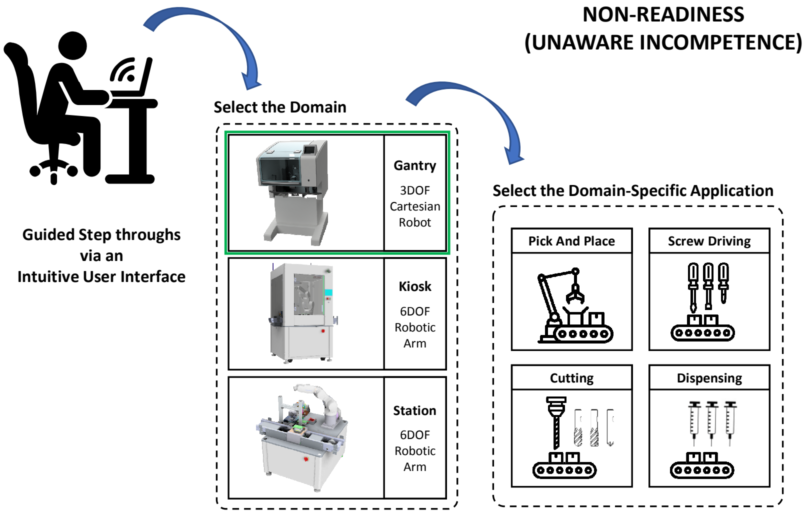


Figure 2. An intuitive user interface is provided to select the domain, application and finally define the goals.

Figure 3 for illustration example for novice stage, where we are at an aware incompetence situation. Basic gesture learning of objects and actions is useful for process tasks that have many alternative options for physical movement, e.g., the movement of a Robotic Arm (kinematics) for point A to point B in a 3D space. Motion capture technology, such as provided by Position Sensors, Proprioception (Stereoscopic and Depth detection by visual and/or ultrasonic) Sensors, can be used. The Trainer performs a motion, this is captured in a 3D Simulation software package. The Simulation is played back to the Trainer for fine tuning. For object assignment tasks the Motion capture technology can be used to point at Objects, and assign them labels as appropriate to the Application specification, whereas an application is referred to the main task of the automation goal such as Pick and Place, Screw driving, Cutting, Dispensing, etc.For instance, one can consider pointing at a Storage container for a Place process to deliver a Specified tool to in a Pick and Place Application specification.

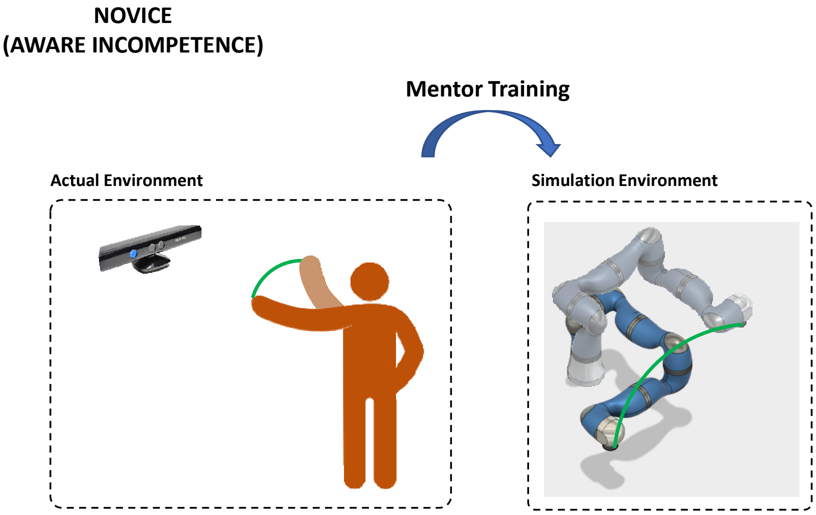


Figure 3. Experts demonstrate via a motion capture sensor to the robot how to travel on a trajectory to perform a specific task.

Figure 4 for illustrates an example of advanced novice stage, where we are still at aware incompetence but more practical scenarios can be investigated. Setting out 3D simulation and Digital Twin is used to provide a realistic simulation environment. Having defined the Domain and Goal for an Application specification as mentioned in the two prior stages above, and having developed a 3D Model/Simulation of the process, the concept of a Digital Twin is available. The Digital Twin is whereby design work can be conducted within the Simulation environment, and for this to map to the deliverable design in the Physical environment (i.e., the Robot physically executing the process in a test/development or production scenario). Changes in the Production scenario would be captured and changes made the Simulation environment and vice versa. The advantage of this is twofold: a) Simulation environments are based on physical models, and may be imperfect in comparison to the actual physical environment, and the capture of anomalies and divergence will help the simulation to improve; b) Once a test/development environment is deployed, changes will likely be ongoing as version upgrades to the environment and/or objects occur, causing maintenance of the process and Application specification.

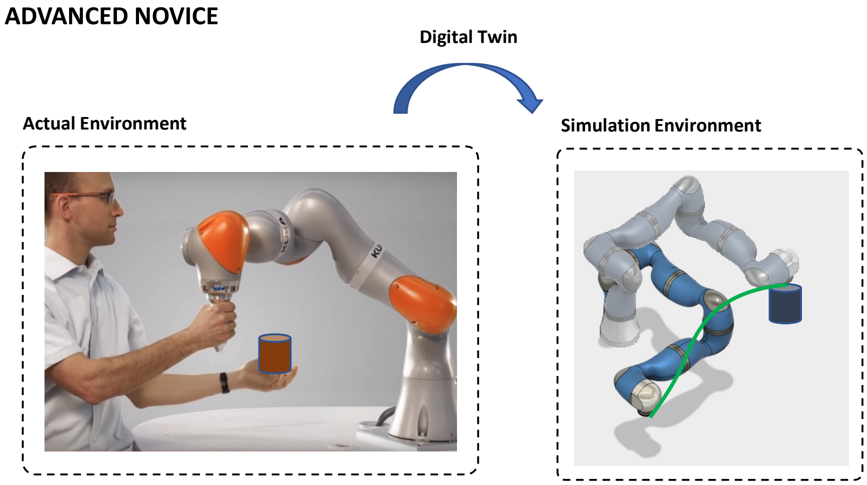


Figure 4. Digital twining as a tool to online monitoring of peripheral changes is employed to have a realistic simulation environment.

See Figure 5 for illustration example of beginning proficiency stage, where we are at an aware competence situation. Learning of possible schemas and stratagems with Twin Space (Design and Tests): In this case multiple scenarios can be tested to optimize the performance of the system. E.g., performance in terms of speed, wear and tear, power usage and precision may be set up as metrics for the characterization of the simulation and a multi-attribute utility analysis (a data science weighted selection matrix). Optimum solutions can be searched for, once the three prior stages are achieved.

See Figure 6 for illustration example of proficient stage, where we are still at an aware competence situation. However, we can start learning more sophisticated automation scenarios. As the knowledge base grows, the AI Planning and Optimization of Schemas/Stratagems (Reconfiguration and Test) can contribute more and more. Once many instances of optimization in the prior stage are undertaken, and captured in a structured data set, this data set can be subject to AI Machine Learning/Deep Learning to recognize the optimum schemas/stratagems. Simple articulation of the schemas/stratagems can be provided (e.g., schema X1 was selected).

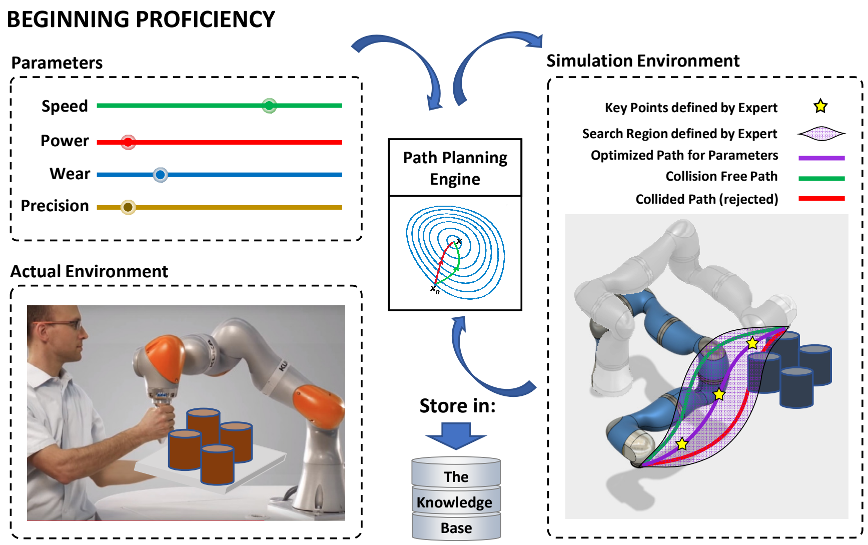


Figure 5. Multiple scenarios for different applications and tasks are collected and optimized through the digital twin equipped simulation environment and the results are stored as the knowledge base.

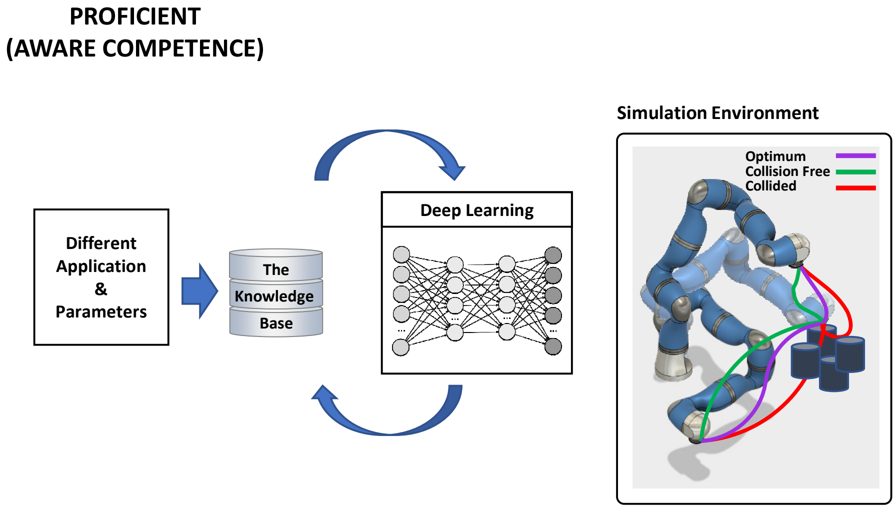


Figure 6. For different applications and tasks, the acquired knowledge base can be transferred into an actual AI framework using Deep Learning.

See Figure 7 for illustration example of expert stage, where we enter an unaware competence situation. AI Rationalization of Learned Schemas/Stratagems (Compliance and Explanation) is the eventual stage of the proposed work. It is anticipated that in many cases, there is in practice a need for automated system to be able to rationalize why it has selected a schema. E.g., to ensure there is compliance to a regulated requirement for process to conform to guidelines (not all processes, even apparently efficient ones may comply.) Deep Learning systems at the time of this application are recognized as having a weakness in this area [15]. Innovative “rule-extraction” systems that interrogate the Deep Learning systems and produce best fit rules that the system is performing against are a means for such compliance checking. The authors have filed a provisional patent application in this area [3].

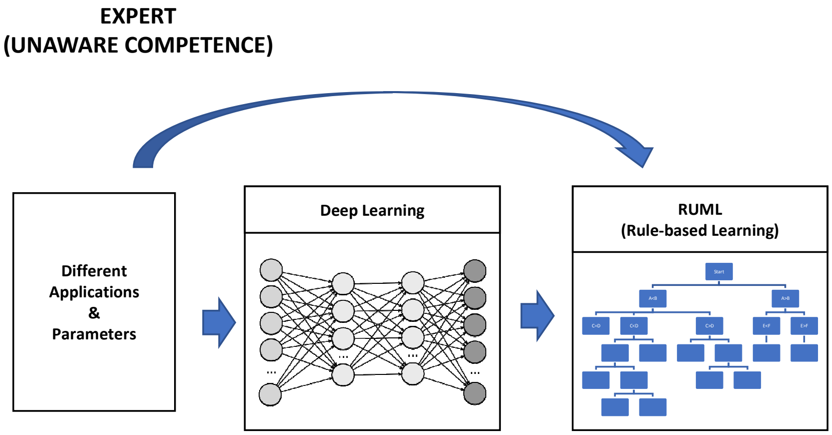


Figure 7. Gradually the rationalization of the relationship between the solution provided for different applications and tasks is obtained using a Really Useful Machine Learning.

3. CONCLUSION

An innovative computer system framework is presented that supports robot automation programming by a gesture controlled “show and tell” process, whereby human experts describe their goals of a process to be learned, and demonstrate via positional sensors the actions necessary to achieve those goals. These inputs are collected and optimized by the robot mentoring system, interpreted back to the human expert for confirmation feedback and/or subsequent fine tuning. This framework, is a pedagogical model, modelled on a professional human process needed to mentor others. We have applied this concept in an innovative way to mentor robots with gesture control and machine learning.

A six-stage training process from beginner to expert is provided by example, where the level of the expertise will transform from an unaware incompetence to unaware competence, then to aware competence, and eventually end up at an unaware competence where the rationalization of the trained skills is obtained. The proposed framework provides a holistic solution to automate the automation step-by-step using AI tools. The application of such a training framework can not only facilitate the manufacturing process but can also significantly accelerate the order-to-delivery process in industrial automation.

REFERENCES

[1] Z. Pan, J. Polden, N. Larkin, S. Van Duin and J. Norrish, "Recent progress on programming methods for industrial robots," Robotics and Computer-Integrated Manufacturing, vol. 28, pp. 87-94, 4 2012.

[2] M. Ciupa, "Is AI in Jeopardy? The Need to Under Promise and Over Deliver - The Case for Really Useful Machine Learning," in Computer Science & Information Technology (CS & IT), 2017.

[3] M. Ciupa, Hybrid Machine Learning Design Based on a Bottom-Up/Top-Down Methodology, US 62/476,068, United States Patent and Trademark Office, 2017.

[4] S. M. Khansari-Zadeh and O. Khatib, "Learning potential functions from human demonstrations with encapsulated dynamic and compliant behaviors," Autonomous Robots, vol. 41, pp. 45-69, 1 2017.

[5] S. Chernova and A. L. Thomaz, "Robot Learning from Human Teachers," Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 8, pp. 1-121, 4 2014.

[6] A. Murali, A. Garg, S. Krishnan, F. T. Pokorny, P. Abbeel, T. Darrell and K. Goldberg, "TSC-DL: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with Deep Learning," in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016.

[7] T. Zhang, Z. McCarthy, O. Jow, D. Lee, K. Goldberg and P. Abbeel, "Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation," 10 2017.

[8] J. Rey, K. Kronander, F. Farshidian, J. Buchli and A. Billard, "Learning motions from demonstrations and rewards with time-invariant dynamical systems based policies," Autonomous Robots, 5 2017.

[9] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell and A. Billard, "Learning and Reproduction of Gestures by Imitation," IEEE Robotics & Automation Magazine, vol. 17, pp. 44-54, 6 2010.

[10] R. Lioutikov, G. Neumann, G. Maeda and J. Peters, "Learning movement primitive libraries through probabilistic segmentation," The International Journal of Robotics Research, vol. 36, pp. 879-894, 7 2017.

[11] A.-a. Agha-mohammadi, S. Chakravorty and N. M. Amato, "FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," The International Journal of Robotics Research, vol. 33, pp. 268-304, 2 2014.

[12] M. Imani and U. M. Braga-Neto, "Maximum-Likelihood Adaptive Filter for Partially Observed Boolean Dynamical Systems," IEEE Transactions on Signal Processing, vol. 65, pp. 359-371, 1 2017.

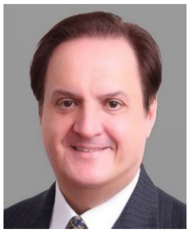
[13] M. Imani and U. M. Braga-Neto, "Particle filters for partially-observed Boolean dynamical systems," Automatica, vol. 87, pp. 238-250, 1 2018.

[14] S. F. Ghoreishi and D. L. Allaire, "Adaptive Uncertainty Propagation for Coupled Multidisciplinary Systems," AIAA Journal, vol. 55, pp. 3940-3950, 11 2017.

[15] Will Knight, The Dark Secret at the Heart of AI - No one really knows how the most advanced algorithms do what they do. That could be a problem., MIT Technology Review, 2017.

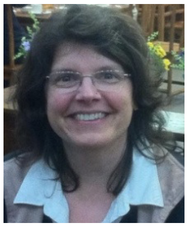
AUTHORS

Martin Ciupa



Martin Ciupa is the CTO of calvIO Inc., a company (associated with the Calvary Robotics group of companies) focused on simplifying the cybernetic interaction between man and machine in the industrial setting. Martin has had a career in both technology, general management and commercial roles at senior levels in North America, Europe and Asia. He has an academic background in Physics and Cybernetics. He has applied AI and Machine learning systems to applications in decision support for Telco, Manufacturing and Financial services sectors and published technical articles in Software, Robotics, AI and related disciplines.

Nicole Tedesco



Nicole Tedesco is the Chief Architect at calvIO Inc, a company (associated with the Calvary Robotics group of companies) focused on simplifying the cybernetic interaction between man and machine in the industrial setting. Nicole has many years of experience in software architecture software engineering, business design, business analysis, financial analysis, regulatory analysis, project management, data mining, social networks, math, physics, editor, curriculum designer, UML, ITIL, IASA, etc. She has worked in large MNC’s and smaller entrepreneurial outfits.

Mostafa Ghobadi

Mostafa Ghobadi is currently working for calvIO Inc. as a Senior Robotics Engineer. He received his BS and MS degrees in mechanical engineering from Isfahan University of Technology, Isfahan, Iran. He has recently graduated with Ph.D. degree of Mechanical Engineering from University at Buffalo SUNY, NY, USA. His current research interests include control and robotics, stochastic filtering and sensor fusion, mathematical modeling and system identification, human–robot interaction, Software Development and AI.

FRAMEWORK FOR ANALYZING TWITTER TO DETECT COMMUNITY SUSPICIOUS CRIME ACTIVITY

Safaa.S Al Dhanhani

Khalifa University, Abu Dhabi, United Arab Emirates

ABSTRACT

This research work discusses how an integrated open source intelligence framework can help the law enforcements and government entities who are investigating crimes based on statistical and graph analysis on Twitter data. The solution supports a real-time and off-line analysis of the tweets collections. The framework employs tools that support big data processing capabilities, to collect, process and analyze a huge amount of data. The outline solution supports content and textual based analysis, helping the investigators to dig into a person and the community linked to that person based on a tweet. Our solution supports an investigative processes composed of the following phases (i) find suspicious tweets and individuals based on hash tags analysis (ii) classify the user profile based on Twitter features (iii) identify influencers in the FOAF networks of the senders (iiii) analyze these influencers' background and history to find hints of past or current criminal activity.

KEYWORDS

Twitter, analysis, crime, detection

1. OVERVIEW

Twitter is a microblogging service for sharing messages restricted to 140 characters [1]. Twitter provides some features for users to communicate with each other: 1. Posts: Twitter provides to the users posting features in which they can post messages on their personal page. 2. Hashtags: Twitter users can use hashtag which is written on the format of: #hashtag, which is relating the tweet to a certain topic. Hashtags are searchable in Twitter; the users can search by a hashtag and retrieve the tweets related to that hashtag. Mentions: Mentions is written on format of "@username", users can use mention to refer a message to another user. Replays: Replay is another interaction way that is used to replay messages. Favorites: Users can favorite a tweet, Twitter provides the count of the favorite related to a tweet as an information. Retweet: Twitter has feature of retweets, which means that user can repost a tweet of someone else, the original writer will always appear. Retweeted tweets provide the number of retweets of that tweet as information.

David C. Wyld et al. (Eds) : CNSA, DMDB, AIAP - 2018
pp. 41– 60, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80104

2. LITERATURE REVIEW

Many researchers have done a lot of work in analyzing Twitter for event detections and event predictions, most researchers have used Twitter features to conduct the analysis, such as: retweets and mentions. Most prediction techniques of events have been derived using textual analysis or sentimental analysis. However, it has been shown that predicting of events can be done without textual analysis or sentimental analysis with successful results, they are several ways that have been used for this purpose: users' communication analysis [3, 4, 5], account analysis[6, 7, 8], prediction of events based on sentiments analysis [9, 10, 11]. Also, it has been showing that Twitter can help on predicting crime location using linear regression [12]. Furthermore, crime analysis used node analysis for crime detection and understanding key players of terrorist on the social network[13, 14, 15]. This section is exploring the related work on crime analysis using social media and particularly in Twitter.

2.1 Related Work on Framework for Crime Detection

Christopher C. Yang and Tobun D. Ng have proposed a framework solution for crime related weblog including links and contents analysis and visualizations[5]. The main pillars of this framework: 1. extraction of community specific topics 2. Specifying the relationship between the bloggers in the social network 3. Content and sentimental analysis 4. Visualization of different level of abstractions. They also introduced some searching techniques, independent and dependent neighborhood graph, such as top N documents from seed set, and HITS algorithms for ranking algorithms. The authors were against NLP because of the difficulty of processing the text on the blogs since it is not always written in a proper language structure. They have introduced some visualization methodologies such as filtering the nodes and arcs which are less relevant to the searched target, having a fisheye view to get view picture of the relationship on the network and having the ability to dig down to other relationships without changing the structure.C. Christopher and D. N. Tobin covered significant criteria of crime identifications, but ignoring language processing will reduce the accuracy and increase the false positive results. The researchers have built a framework that only operates on blogs and did not show the way of verifying results. Also, it did not highlight the elimination techniques of false positive accounts. In addition, the authors did not suggest the technologies for implementing this framework.

2.1.1 Twitter User Account Analysis

Zhang and Paxson have developed a method to identify the automated Twitter account based on the behavior of the tweets on a periodic manner [6]. The study has declared that the organic Tweets represented on randomly distribution patterns; however, automated Tweet forming a structured pattern. The reason why it is formed that way, because usually they are running on a scheduler that tweets every minute or every day [6]. During the study, they have captured a small number of false positive results 2 out of 1000, and this was for a few accounts that update on a regular time basis; for example, students who update timeline every day after their classes. Another example, Dr. Phil updates his timeline every day before the show. Unlike the false positive, false negative has a higher probability and that was because of some reasons such as hybrid behavioral account which mask themselves with organic posts. Another reason is that some automated posts are based on RSS feed, not schedulers, so it is created with the feed updates. This evaluation can evade if the automated account has mutated its behavior according to know organic account. According to the analyzed data, 14% of the public account were posting discernible automation [6]. While 24% were generated by automated bots, and 15% of the tweets were generated from the automated source [6]. The authors have also included verified accounts analysis, it has been found that most verified accounts belonged to celebrities and popular companies. They are many verified account that failed on the test such as popular brands, TV shows fans, political figures, news, non-profit organizations and government organizations. According to the finding most users are not verified accounts, 40% were verified users with the sample used. Surprisingly, the verified users seem to be automated account with compared to non-verified accounts, 6.9%, 16% consequently [6]. The researchers found that tweet source has clearly identify the account type, it is found that people used mostly phones for tweets while automated accounts used API to post tweets. The research has highlighted interesting findings of automated account based on the trained data, but the study did not include content or sentiment analysis to increase result accuracy. In addition, the reseach did not emphasize the age of the accounts and number of followers, as profile indication of automated or non automated accounts.

Klatsch framework which analyzes the users' behavior in Twitter feeds using graph analysis[7].The designed model represented as events and memes where actors have the role of creating the events, who represent the users whereby the memes represent the information in a needed level of detail. Each of these are related to a unit of weight and edges in the network structure. An example explaining this; event can be a tweet which is created by an actor who can be a user who posted, or retweeted a tweet. The intended solution aimed to identify organic and non-organic tweets, based on topologies of the Twitter communication network. In [7], the researchers have highlighted interesting topologies pattern of non-organic tweets. However, they did not provide how they utilize the sentiment analysis with the graph analysis. In addition, they did not cover user classification and evaluation of the tweet accuracy, which increases the false positive and false negative as well.

2.1.2 Detection of Events Based on Twitter Communication

In [3], the authors have proposed a system composed of two processes: online and offline processes. The online processes contain ranking and clustering, which intended to detect events, and online searching for analytics. The clustering model designed to group the similar tweets according to their geo location, and their relevant timeframe with respect to their importance. On the other hand, offline processes retrieve the data from the API crawler then extract the CDE related tweet according to classifiers, then index the data and store it in the database. The researchers have used several analytic techniques to extract event related to tweet such as: 1. linear regression to predict friends' location when the information related to their location does not exist whereas tagged or mentioned by another user with geo location, 2. ranking techniques by using Twitter feature which analyzes the content according to crime or disaster related words such as kill or accident hashtag,3.user profile analysis using Twitter user feature by Twitter profile information and deciding whether the user are considered as credible or not. The analysis includes number of Twitter attributes such as: number of tweets, the age of the account and account verification information. 4. Usage feature is used to define the most important tweet explicitly and implicitly by having the number of retweets and favorites as an indicator. The research has identified key values to identify the CDE; however, it did not demonstrate how URLs can distinguish whether it is related to an event or not. In addition, it did not show how to correlate the hashtag to an event, they only related these parameters to geo locations, which is not always enabled in users' account. Moreover, detecting the friend's location from linear regression as they have mentioned may got error because location may differ each time, for example, whenever I met with my friends we always try to choose a new location!

Unlike TEDAS approach, in [4] the researchers have found an interesting pattern with respect to communication in Twitter during the world cup 2011 event, using non-textual data. They have considered the newly event, and the influencers of the event via users' interactions, as posts for newly events, and retweets or mentions for the influencers. In addition, they have address a methodology to extract event from a raw number of tweets and retweets that occurred during these events using linear classifier. The authors have found that people are less social during the event; whereas they are busy with mentions, retweet and replies after the event, while they are posting about the event during the event. The authors in [4]have contributed in successful results the pattern of events related to world cup; however, the mechanisim fails to identify the creditable sources of the event to get more accurate results. In addition, this methodology fails to identify the event impact in the society, good or bad.

2.1.3 Predicting Crime Location Using Linear Regression

In [12], the author has used Twitter specific linguistic analysis and statistical analysis based on topics which are related to crimes, the study has examined sample of crimes in Chicago city, which contains the crime type with geo location records. The researcher used this as a historical data, then examining the geo location tagged tweets which are related to a crime. He has compared the traditional linear regression output with Twitter feature analytic for prediction. It has been shown that Twitter data has improved the results of the prediction; however, the author did not provide prediction of the crime that may happen on the location with time, so that the police get ready to prevent the crime from happening. Moreover, the current solution does not provide network analysis of the accounts, to identify the criminal. In addition, it is not studying account analysis, and trending topics, to have more accurate account and tweet credibility.

2.1.4 Detecting Crimes Based on Nodes Analysis

R. YK Lau, M. Kamal H and M. I. Pramanik, [13] have proposed a framework for detecting crimes based on criminal network patterns, the framework has used structure analysis based on centrality measures and network mapping. Each of these measures concluding a role or a dependency of criminal in the network. The researchers used a dataset from official site of the Los Angeles County Sheffi's Department, the data collected was holding records from 2004 to 2005. The researchers have studied several crime types and they used some attributes of the crimes for evaluation, each record was related to an individual criminal. In [13], they have only studied the relationship between the criminals on the social network, but it did not looked at other factors like weapons, locations, and organizations.

In [14], the researchers have used similar algorithms used in [13]; however, they have also included Page Rank and eigenvector algorithms in the analysis of social network. The researchers used leaked data from data theft service of Nigerian advanced fee fraud scammer, then they have searched for Facebook accounts related to criminal people, getting their profiles. Using that technique, the researchers have linked criminal profile with their friends building the social network for analysis. The study has found that key members of criminals have high rank of centrality and well-connected members. H. Sarvari, E. Abozinadah, A. Mbaziira and D. Mccoy have found groups based on Facebook communities but they have not validate the communication of the email addresses, to accuratly identify the relationship between the parties and the strength of the relationships.

Ala in 2012, proposed graph algorithms to find the financial manager on a decentralized terrorist network. It's found out that financial manager is the most important node in the network, it's the most operative and have an active relationship with other nodes in the network [15]. The researchers have used the subset of categories used in NATO (North Atlantic Treaty Organization) model, AIntP-3 data model. Ala has successfully defined the financial manager in the case study analyzed, finding that financial manager playing key player in the terrorist network. Ala has successfully identified the financial manager as a key player on the terrorist network; however, the study did not show how to detect the terrorist network in the diffused social network. The study did not show how to map nodes or actors to entity from data collection from social media, it is only explaining how graph analysis can be applied as in NATO with data collected from news. It did not demonstrate extraction of NATO categories from the social media.

2.1.5 Prediction Based on Twitter Data Analysis

There are many researches used Twitter data to predict events such as elections, plays, or crimes' locations. They have used different aspect of Twitter data, such as Twitter communication which was explained on section 2.4, or content analysis of the text, which will be explained on the following subsection. Others they have studied historical data collected from non-social media source and map it with Twitter to predict crime location intensity.

2.1.6 Prediction Based on Sentiment Analysis

On the crime pattern detection using online social media [10], it addresses two main domains: geo based analysis of the tweets with respect to selected cities, and intensity analysis of the cities by applying sentiment analysis to the collected tweets. They are several tools used for sentiment analysis: subjective analysis which focuses on the opinion and ignores the facts, for this methodology "Bayes" and cut based classification were used. Another methodology that does polarity analysis of the subjective sentence. This classification composed of two classes binary and multiclass classification. The binary classification is composed of positive and negative whereas the multiclass classification is composed of five categories: strong positive, positive, neutral, negative, strong negative. Dictionary based ANEW was used as well, it's providing a set of normalizing emotional rating for English words. Recursive Neural Tensor Network RNTN sentiment methodology used and it provides misclassifications: very positive, positive, very negative, and negative. The result of sentiment analysis with respect to the geolocation provides the trend of crimes that happened at that location. Raja's research has address good sentiment analysis technique but the paper doesn't provide solution to identify suspicious people that on the area or prediction techniques to identify crimes that may happened on the location.

In 2011 A. Bermingham and A. F. Smeaton studied prediction of election results using volume based and sentiment analysis in Twitter [11]. The study was designed for the Irish general election which took place on the 25th February 2011. The data were collected for the main parties of the election. Finna Fail (FF), Green Party, Labour, Fine Gael (FG) and Sinn Fein (SF). The collection was based on hashtags of the names and abbreviation. The evaluation for the parties used different sample of time set. According to the study, the sentiment analysis has shown some positive results compared with the traditional volume based analysis, but also introduces failure. The researchers had failure in using sentiment analysis for prediction of the election, and find that volume based was more successful, concluding that was due to the popularity of the parties will make them get more polls.

They did not address the problem behind the problem of sentiment analysis was not accurate and they could not distinguish between from sentiment analysis between people preferences and people reaction to news. The authors did not analyze the credibility of the tweets and the sources which increased their errors.

Ramteke, Shah, Godhia, Aadil have proposed a model to analyze election results using Twitter sentiment analysis [9]. The researchers have used public Twitter API to collect tweets, then they manually label the data using hashtag clustering. The authors used VADER (valence Aware Dictionary and Sentiment Reasoner) used for social media text, and it returns polarity of the sentence based on emotions. The researchers have collected data for training of two days, passing keywords: Democrats, Republicans, and the names of the candidates. They have compared several text-based classifiers for sentiment analysis: SVM linear kernel, SVM rbf kernel, SVM-liblinear, Naïve Bayes – multinomial NB, after the training they have concluded that SVM with linear kernel was the most accurate and they have selected it for entity classifier. According to the analysis used, Ratio = |p|/T where p is the total number of positive tweets for a candidate and T is the total of the tweets that is related to the candidate.

The research has addressed a good interpretation of sentiment analysis selection but it did not address the prediction of the winner. Also, the framework fails to provide automation analysis of the tweets, which require extraction of tweet body from the JSON to CSV for analysis, which is not feasible since data collection of Twitter is huge. Also, it does not provide features to compare the new and old data for analysis.

3. PRELIMINARIES

This section explains the concepts of network centrality and point out three types of centrality measurements, moreover, it is explaining the way to calculate each of these measurements.

3.1 Network Centrality

It has been found that network centrality analysis help on identifying many factors that may affect the network such as fast distribution of information, stopping evil nodes, and protecting the network from cracking[16]. According to [17]centrality is defined as: a way to formalize intuitive notions about the distinction between: the importance, and the internals and externals. They are three measures that highlight the importance of the node in the network: (i) betweenness,(ii) closeness, (ii) degree centrality

3.1.1 Degree Centrality

Degree centrality is the sum of the connection from a node to a node in the network, an example of this is authority, hub and PageRank algorithms [15]. The node with more number of links got a higher importance on the network.

Equation 1[16]

$$C_D = d(n_i) = X_{i+} = \sum_j X_{ij}$$

The following equation is calculating the degree of node n_i where $n_i \in$ network of nodes, which is the sum of the X edges between n_i and n_j

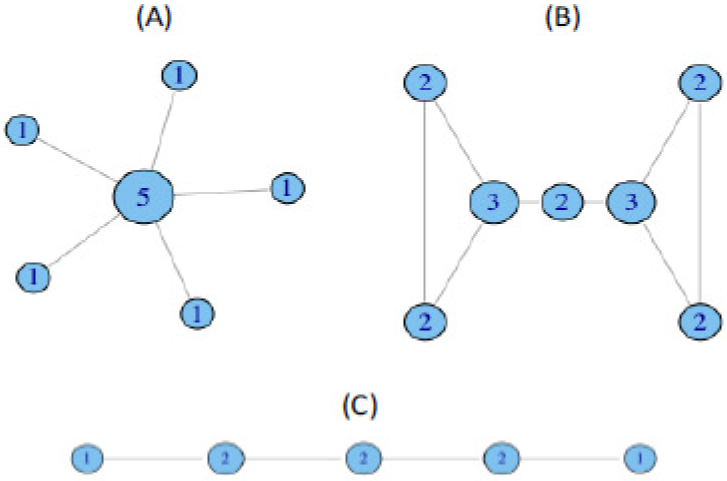


Figure 1 Degree Centrality

Figure1 showing different topologies with different values of degree, and it is observed that start topology has the highest degree of centrality which is Figure A. The more edges connected to the node the higher degree of centrality. High degree indicates importance of the node in the network, but if all node only connected to one node, which made this node high degree, but the graph is not well connected. An explanation of the above, if a node with five connections as shown in Figure A, has degree of five while the other nodes have degree of one because they all have one connection only, it made the graph low connectivity, because if a node was damaged the graph will be disjointed graph. For this reason, calculating the graph connectivity giving greater value to evaluate the graph, Freeman's formula and centrality degree variance used for this purpose.

Equation 2[15]

$$S_D^2 = \left[\sum_{i=1}^{g} (C_D(n_i) - \overline{C}_d)^2 \right] / g$$

Equation 3[16]

$$C_D = \frac{\sum_{i=1}^{g} \left[C_D(n^*) - C_D(n_i) \right]}{[(g-1)(g-2)]}$$

Equation 2 has calculation of the variance between each node centrality, whereas equation 3 is Freeman's formula, which has range from 0, to 1. Degree variance calculated as following: the difference between degree of centrality and the mean square divided by number of nodes. Freeman's formula is calculated as following: the summation of difference between the maximum degree of centrality and the degree centrality of each node on the network divided by sample size.

3.1.2 Closeness centrality

Closeness centrality is value of the distance of one node to other nodes in the network, it is calculated as the inverse of the summation of the distances between a node to other nodes in the network.

Equation 4[16]

$$C_c(n_i) = \left[\sum_{j=1}^{g} d(n_i, n_j) \right]^{-1}$$

Closeness measures how much nodes are reachable, and the possibility that nodes to be connected to each other. The higher closeness between nodes means the nodes are well connected, however, the lower closeness between the nodes means they may some disconnects between the nodes in the network, or the distances between the nodes are big.

3.1.3 Betweenness centrality

Betweenness centrality presenting the number of frequency that a node is between other nodes in the geodesic paths. Equation 5is calculating the betweenness, where g_{jk} is the number of geodesics between jk, and g_{jk} (*ni*) is the number that actor *i* is on network [16]. Equation 6 is the normalized version.

Equation 5[16]

$$C_B(n_i) = \sum_{j<k} g_{jk}(n_i) / g_{jk}$$

Equation 6 [16]

$$C'_B(n_i) = C_B(n_i) / [(g-1)(g-2)/2]$$

Betweenness measures the nodes that all shortest paths cross from them, which indicates an important node, if this node attacked it may damage the network.

4. FRAMEWORK

This section describes the framework approach of the solution provided, including the methodology and the tools used, how they are utilized to achieve their purpose.

4.1 Introduction

This research proposes a framework to analyze and extract intelligence from social media contents such as Twitter posts. The framework will be using powerful tools designed for the big data analytics such ELK (elasticsearch, logstash, kibana) cluster, and Neo4j graph associations and relationship diagram. The framework will include key features for data visualization, discovery, exploration, and analysis. This solution built to facilitate data search and filtering, moreover, it's suggesting methodologies to detect the crime activity and predict criminals based on Twitter data. The methodology used in this study begun with having predefined people accounts or hashtags that are related to crime, and originated from locations in the UAE. The collected data related to historical data of criminals who were arrested due to Twitter activity. They are several techniques applied to identify the people who may related to these communities: sentiment analysis, account classification techniques and graph analysis were applied to detect and help to predict the suspicious communities. The results of this analysis were validated with another data source and tool with successful outcomes.

4.2 Framework Structure

The framework consists of four main components; data collection, storage, and search and visualization. The framework includes two types of analysis, statistical and network analysis. The reason why we have considered network analysis is, statistical analysis can capture significant information about the users, the tweet information, most and least important aspects with respect to Twitter attributes; however, statistics cannot define the relationship between users and possible relations which are important for investigator or analyst to find, such as: the relationship between the people and define the community of suspicious communities.

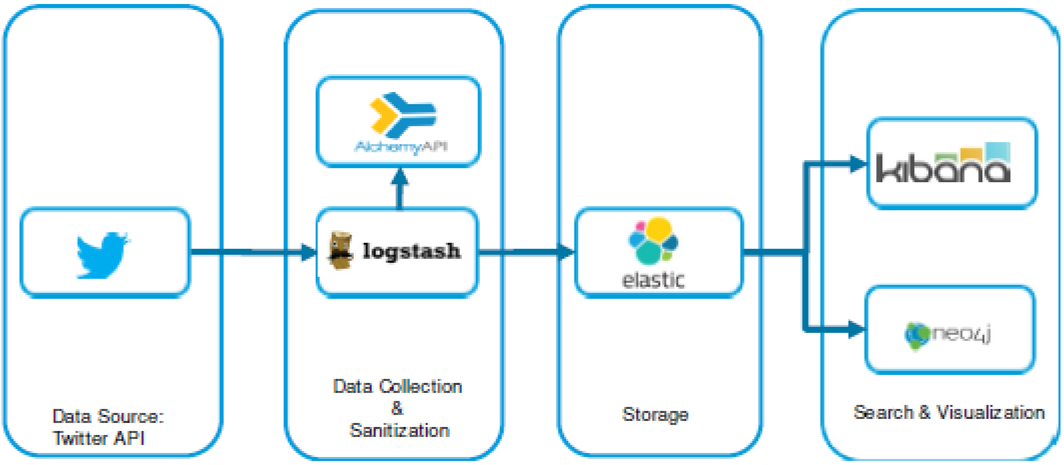


Figure 2 Framework

4.2.1 Tweets Collection

Twitter Developer API used to collect tweets with certain settings Twitter API allows developers to pass keywords of interest to limit the collection of interest. This API collects only 10 percent of random tweets, the process of collection started on Dec 14, 2016 to Feb 24, 2017, with collection of 68,438 hits. The keywords used are hashtags of UAE activist and hashtag of Daesh “داعش”. and some related keywords in Arabic, and geo locations in the UAE, adopted from [3] getting related keywords, and locations, by using content feature of Twitter. The collection was random during the day, nights and mornings.

The first few weeks of collection it was rarely to find tweets with geo location, when the collection was based on keywords of hashtags and usernames only. Then, geo location passed into the keyword fields and started collect more tweets with geo location information. Even though the locations specified on the keywords was based on the UAE, there were some tweets collected from other locations on the map. It seems that Twitter used keywords with OR conditions, to get more results related to at least one of the keyword.

4.2.2 Logstash

Logstash is an open source solution works as an engine that can collect data on a real time pipeline capabilities [18]. It can normalize, unify and sanitize the data into other formats. These capabilities enable Logstash to do log and events collection, the beauty of Logstash lie on the supported input and output plugins which facilitate the data ingestion and data analysis. The collection is established by configurable inputs, which can be different type of data such as network streaming events, or some files. The configuration allows the data to be filtered, to eliminate unnecessary data. Finally, output configuration to pass the data into other files to store.

4.2.3 Sentiment analysis

The sentiment analysis is used to detect the polarity of the text emotion, positive, negative and neural. The positive means that the person is happy, whereas negative indicates that the person is unhappy or angry, when the text detected neither positive or negative criteria was succeeded, it is considered as neural. Having this methodology is not to find out the happy criminals on Twitter, combination of hashtag analysis with sentiment analysis can obtain the supporter, or welling to be, or is belongs to suspicious community. Sentiment analysis has shown successful results on predicting elections in other researches [9][10][11]. In this research sentiment analysis with hashtag analysis can filter the community of interest in crime activity, this facilitate for the investigator the way of extracting the community as first step. The technique used for sentiment analysis is Alchemy API from IBM.

4.2.4 Elasticsearch

Elasticsearch is an open source engine, to organize data and make it accessible for search, and support queries to search with aggregation with near real time processing [18]. Elasticsearch supports system distribution which can utilize shards and replicas, via routing and rebalancing of data and processing. It is integrated with Logstash as ELK cluster for analytics of data collection and log parsing.

4.2.5 Kibana

Kibana is an open source analytics and visualization platform intended for ELK cluster, to visualize Elasticsearch queries for the end user. It supports real time visualization of queries with aggregation and with near real time latency. Moreover, Kibana supports different types of charts, and tables used to build dashboard easily [18].

4.2.6 Neo4j

Neo4j is NoSQL graph database that is implemented using java and Scala, it implements property of graph model and storage as well. Moreover, Neo4j provides database characteristics including ACID transactions, clustering, runtime failover

4.2.7 Neo4j Schema

The designed model represented the user as an actor who has the role of performing multiple actions such as tweets, retweets, tags, and mentions, the related actions are linked to one or more tweet, Figure 3 is illustrating the relationships

.

- Posts: A user posts a tweet
- Mentions: A tweet mentions one or more user
- Replys to: A tweet replays to a tweet
- Retweets: A tweet retweets another tweet
- Tags: A tweet tags one or more hashtag

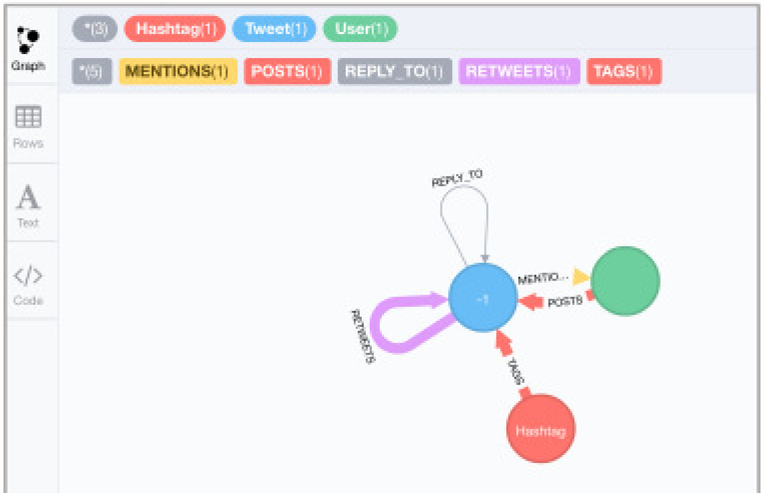
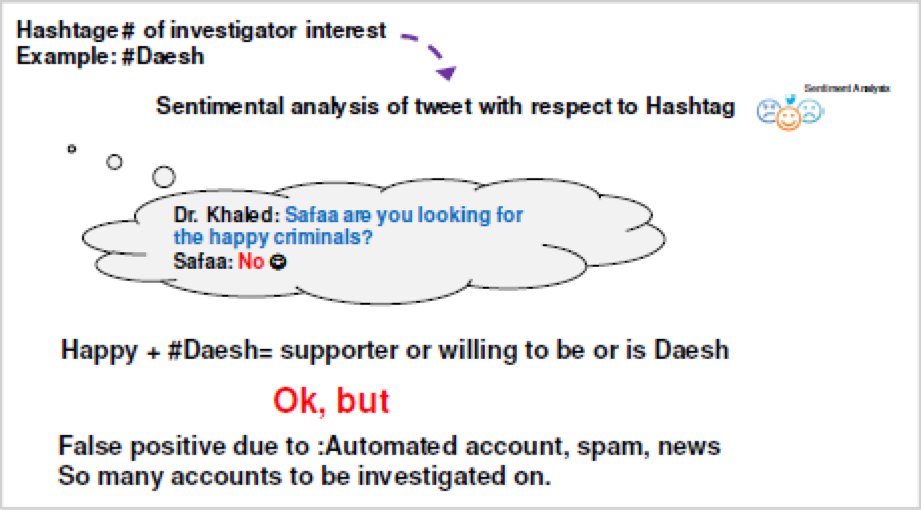


Figure 3 Neo4j Twitter Graph

5. ANALYSIS

This section explains the analysis used in the study, statistical analysis and network analysis. The Statistical analysis provides indication of the account type identification and the activity level as a first level of examination while network analysis adding more refined analysis about the relationship between the main player of the Twitter network which will be explained later.

Figure 4 illustrates an overview of what investigator will be concerned about, as the first point of interest will be searching for people who are interested in a topic like Daesh, and sorting them down by their impression about this hashtag, using the sentimental analysis, having this can give the investigator some sort of solution, but there are still some challenges that may face investigators such as false positive and false negative account investigation, therefore, we have introduce some statistical analysis which will help reducing this problem. The second point of interest will be identifying the relationship between people and predicting the people who may cause crime by analyzing Twitter network, which will be explain in the next section, network analysis.



Hashtage# of investigator interest
Example: #Daesh

Sentimental analysis of tweet with respect to Hashtag

Sentiment Analysis

Dr. Khaled: Safaa are you looking for the happy criminals?
Safaa: No ☹

Happy + #Daesh= supporter or willing to be or is Daesh

Ok, but

False positive due to :Automated account, spam, news
So many accounts to be investigated on.

Figure 4 Analysis Approach

5.1 Statistical Analysis

The statistical analysis based on statistical methods applied to some of Twitter features or data, which will interest investigator or analyst to consider. We have adopted some criteria that was applied in the previous research areas[3],[6], [8]. Most researchers have used Twitter features for to conduct statistical results that can be obtain predictions. In this section will well be exploring multiple inputs that can support examining user's accounts, having multiple methods can increase the confident level on the prediction.

5.1.1 User classification analysis using activity level inspection

Investigator must have clear profile information about the person to be investigated on, user classification is a key factor for analyzing personals of suspicious activity, and helping the agent to better known false positive account for elimination from the suspicious domain. We have adopted Zhang and Paxson methodology in classifying the automated accounts and non-automated accounts. Automated accounts are known for tweets generated from another source that a person does, such as: APIs, RSS feed. In the experiment, we have inspected the level of activity of two accounts were from top twenty-five accounts tweeting with hashtag Daesh in Arabic, and with positive sentiment results "11119aass", and "justasender", Figure 5 is showing how is the behavior of account. "justasender" From the figures, it is clearly shown that patterns should be eliminated from the analysis due to automation detection.

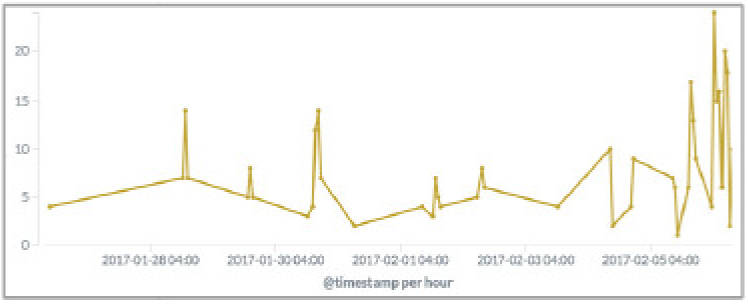


Figure 5 Justasender Account

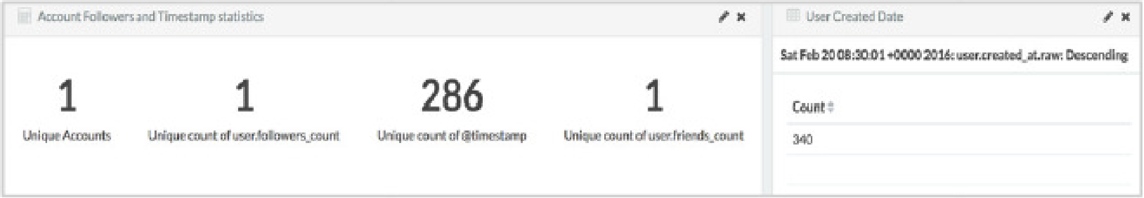
5.1.2 User classification analysis using account verification attribute

Another area that used to analyze user profile, is validated accounts and non-validated accounts, similar results are showing with the research conducted by [6]. Lower number of verified accounts compared with non-verified accounts, with percentage of 2.04%, 97.96 consequently.

During the user classification analysis, while sorting the verified accounts, it was clear that most verified accounts belongs to news or organizations, aa [6] they found in their result, an example; for top twenty-five accounts, "Sharjah24" which is an account for news and, "uaefa" account for UAE football association official Twitter account.

5.1.3 User classification analysis using account age and history

Adopting from [3], user feature such as: age of the account give the investigator degree of credibility, in how much trust should consider this account has valid information. In addition, having the number of follwers, and friends, giving an idea how much this account is popular and can be an influencer account as well.



Account Followers and Timestamp statistics

1 Unique Accounts

1 Unique count of user.followers_count

286 Unique count of @timestamp

1 Unique count of user.friends_count

User Created Date

Sat Feb 20 08:30:01 +0000 2016: user.created_at.raw: Descending

Count
340

Figure 6 User profile information

Figure 6 is showing how the investigator can get the account information such as: number of the followers, the tweets, and friends of this account, moreover, the account creation date.

5.1.4 User classification analysis using tweet source inspection

Tweet sources evaluation have contributed on classifying the accounts, it has been shown that tweets which are generated by a third-party application or an API and having a higher proportion automated account in [6], [8], also with the sample used for this research it is found that people are using more phones, tablets or integrated API with other social media application like Facebook or Instagram to post tweets, organizations or other automated accounts use APIs to generate their tweets.

5.1.5 User classification analysis using account mentions and posts behavior

As [8] found comparing the communication of the tweets, finding that people are posting tweets with mentions and replays more that organization. When comparing the top twenty-five active posting account and top mentioned accoutns, it shows that top active posting account belong to organization or news, however, top twenty-five mention screens belong to people who may have more influence in Twitter network

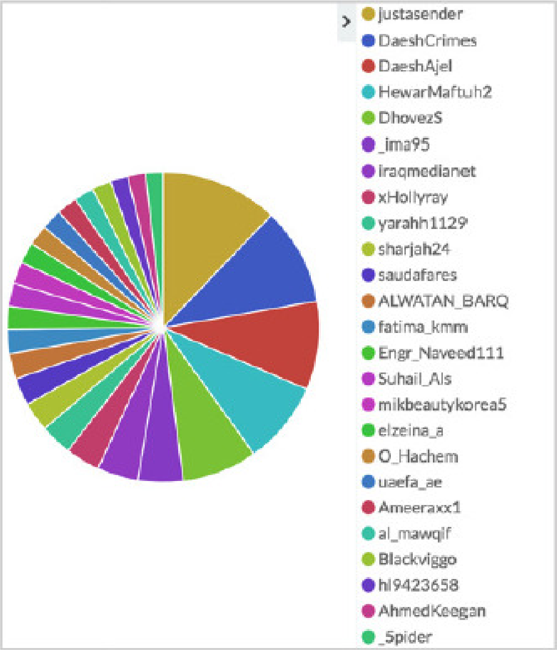


Figure 7 Top 15 Active account by number of posts

5.1.6 User classification analysis using sentiment analysis

When applying sentiment analysis of the total volume of the tweets generated with verified and non verified accounts, it displays that people have more positive and negative, that neural, on the other hand, organizations and automated accounts have more neural with 15 percent in deference, this also was observed in [7]. Usage feature of Twitter has significant impact on analyzing the tweets importance evaluation, adopted from [2], having number of favorites and retweets shows the invistigator how much enflunce has this tweet on the network.

5.1.7 Network Analysis

Each entity in the network represented by a node and each action initiating a relationship between the nodes, section 4.2.7 explains how the schema built, and relationship implemented in the framework. We have considered the results found in the literature for studying the network analysis of Twitter network, most researchers have applied centrality measures to study the behavior of the malicious nodes. Section 3 explains the concepts of network centrality and how it is calculated. The network analysis can predict the suspicious nodes by calculating the centrality of the bad nodes and the neighbor nodes, this can predict the direct and indirect communication between the nodes. The results found that influencers of the network have the most centrality in the network compared with other nodes.

5.1.8 Malicious Node Analysis

While statistical analysis can give indication of the nodes that needed to be examined, network analysis can provide the relationships between each node. For this research, we have identified malicious node as people who were arrested due to crime activity using Twitter network, UAE activist who were arrested in 2012-2013, belonging to terrorist groups like Ekhwan, Eslah Group

[19]. Most these accounts are inactive accounts, or suspended account, during the study, we have found some active accounts.

5.1.9 Predicting the Suspicious Node in Twitter Network Based on Communication Centrality

Nodes who are influencers in the network; are the ones that concern us, malicious nodes that were explained previously found with high degree of centrality with communication relationships, which made them influencer in the network. We have found that each of these accounts may communicate to other similar nodes properties like, having high replys / mentions centrality, but also, they may communicate with other accounts which are having less communication centrality, but they also may communicate with unknown other high degree centrality, which indicate indirect communication with nodes that did not seem to be suspicious in the beginning.

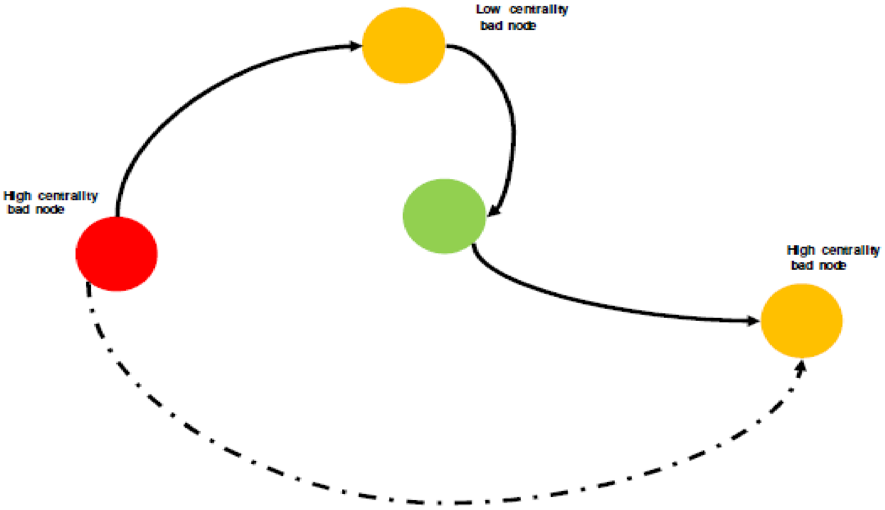


Figure 8 Network analysis to detect bad nodes

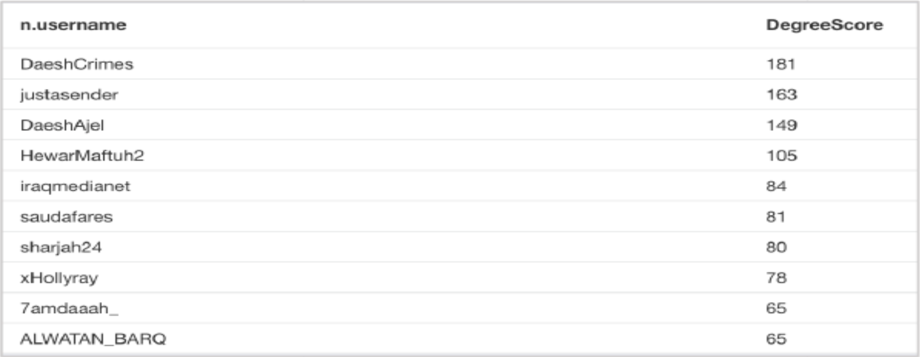
Figure 7 explains the above, the dotted line showing the indirect communication, red node is malicious node, orange are suspicious nodes, green looks like the legitimate node. The next subsections describe the methods to recover these nodes. We have considered replies, mentions, posts, as type of communication which are more relevant to people communication with each other in the Twitter network.

5.1.10 User Classification Analysis Using Account Mentions and Posts Degree Centrality.

Calculating the nodes centrality in the network is key starting point to identify the influencers of the network, in this section, we have examined top ten largest centrality of the network based on mention relationship and post relationship, the results are shown in the below tables. Comparing the results with statistical analysis, both are showing the same results, section 5.1.5 pie charts are presenting the same results, of count.

Neo4j Query: match(n:User)-[r:MENTIONS]-(m:Tweet) return n.username, count(r) as DegreeScore order by DegreeScoredesc limit 10;

Neo4j Query: match(n:User)-[r:POSTS]-(m:Tweet) return n.username, count(r) as DegreeScore order by DegreeScoredesc limit 10;



n.username	DegreeScore
DaeshCrimes	181
justasender	163
DaeshAjel	149
HewarMaftuh2	105
iraqmedianet	84
saudafares	81
sharjah24	80
xHollyray	78
7amdaaah_	65
ALWATAN_BARQ	65

Figure 9 Degree centrality of posts relationship

5.1.11 Shortest Path Between Users with Respect to Mention Relationship (Betweeness)

Degree centrality calculates how much the node active in the network, in term of the number of relationship the nodes have with other nodes, this cannot define the relationships, and the paths between each node, while shortest path algorithms can calculate the possible paths between the nodes, and the type of relationship needed to establish that relationship. The below query calculates shortest path between Node A, and Node B with the mention relationship, the returned values are the paths that connects these nodes together, which are the tweets between them. Figure 10 illustrates the shortest path query results.

Neo4j Query: MATCH p=allShortestPaths((u:User{username: "A"})-[:MENTIONS*0..10]-(u2:User {username: "B"})) RETURN p

The same approach can be applied with different types of relations like: reply, post, retweet, only the type of relationship specified on the query changed.

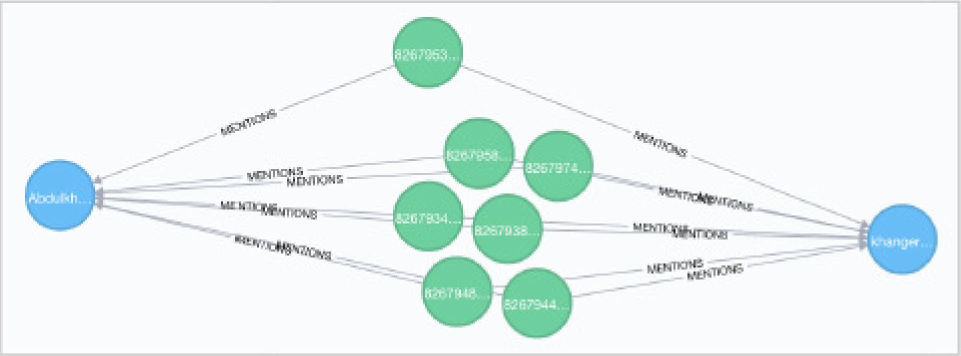


Figure 10 Shortest Paths Between Two Users by Mention Relationship

5.1.12 Shortest Path Between Users with Exhausted Search for All Paths Between Users (Betweeness)

While the pervious method resulting valuable information, the investigator may not be sure the kind of relationship that connects two or more nodes together, and need more like searching for reachability between the users. Shortest path with exhausted search in all path find all relationship between nodes in the Twitter network. This concept finds all relationship between suspicious nodes in Twitter network.

Neo4j Query: MATCH (cs:User { username:"A" }),(ms:User { username:"B" }), p = shortestPath((cs)-[*] (ms))WITH pWHERE length(p)>1 RETURN p;

5.1.13 Shortest Path between Users with Indirect Communication (Betweeness).

Shortest path between users who have indirect communication, this reveals indirect communication between nodes which can be suspicious.

Neo4j Query: MATCH (cs:User { username:"A" }),(ms:User { username:"B" }),(vs:User {username:"C"}), p = shortestPath((cs)-[*]-(vs)) WITH p WHERE length(p)> 1 RETURN p

6. VALIDATION

Validation is very important step in any experiment, for this research we have used Matego, as another tool of verification. Matego is a tool that is used for security reconnaissance that gather information about a target, and it can relate the data together. During the study, we have found a suspicious user account that was communicating with a known activist that was arrested in 2013, we have inspected the suspicious user account to validate the result, using his email address, that we have found it posted also in Twitter. We have found the user was communicating via email to community that belongs to Daesh. While searching the emails that were found in the communications, we have found that one of the emails posting in public blog some media related to Daesh.

7. SYSTEM DESIGN & IMPLEMENTATION

This section demonstrates the design of building the framework solution, they are several components to build this framework to execute each functionality of the framework. The main functionality to of this system:

- The system should provide data collection
- The system should provide data parsing and indexing
- The system should provide data storage
- The system should provide visualization
- The system should provide analysis

7.1.1 Elasticsearchstack

Starting with the collection which is retrieved from Twitter API, the developer edition [20], then indexing and parsing by Logstash, Logstash, is parsing the JSON files using Logstash template, moreover, all tweets are sent to sentiment analyzer by AlchemyAPI. The output from AlchemyAPI then sent to Logstash again to be inserted into the tweet information. Then it's indexed to Elasticsearch and stored. After that the data are ready to be analyzed by Kibana which provides statistical views of the data. For this solution, the cluster installed in on one machine having all the nodes installed with the following specifications:

- Elasticsearch-2.3.5
- Kibana-4.5.3-darwin-x64
- Logstash-2.3.0
- Python 2.7
- Py2neo 2.0.9
- Neo4j-community-3.1.0-rc1
- Alchemyapisdk
- Neo4j Cluster

Neo4j clustered with Elasticsearch to get the collected data to be pushed for Neo4j for node analysis. The pushed data is JSON format, it's also labeled and built into schema to build the relationship between the tweets and the users. Python script is used to pull the data from Elasticsearch, and display it on neo4j, it uses Elasticsearch library and py2nev.2.0.9. Logstash Configuration

7.1.2 Visualization Using Kibana

Kibana allow building different dashboard and uses different type of graph such as pie charts, line charts and bar charts. In addition, it also supports some aggregation functions and statistical functions and prediction graphs. For this project, most the graph used are statistical graph with the use of aggregations such as unique counts, or sum. Moreover, time series graph to provide some prediction of the hashtags based on time. The following graphs are sample of dashboard developed to analyze the overall collection such as the count of the hits, and language detected, users' analysis, content shared such as URLs and hashtags, moreover, map plotting the geo location of the tweets with count.

7.1.3 Visualization Using Neo4j

Neo4j provides different level of abstraction for the searched query the following query provides 1000 nodes that have post relationship. Neo4j provides deeper level when double click into a topology to find the related nodes: tweets and users and the relationship between them; which help observation without a key search.

8. CONCLUSION

This research we demonstrate how social media can detect suspicious crime activity using Twitter data for analytics. This solution is built to facilitate data search, filtering and suggest methodologies to detect crime activity for investigators and intelligence entities. The paper proposing a framework that utilizes big data capability tools, to process and analyze the data from Twitter. This framework provides two ways of analytics, statistical and network analysis of Twitter feature, sentimental analysis also provided to increase the quality of the data to be inspected for investigation. We are exploring how these combined methods can perform real case investigation with certain results, using another way for validation such as Maltego.

9. Future Work

For future work, we are looking for including weighted graph with the sentiment analysis result combined, also the timestamp of the tweet, to enhance the searching result and pattern inspection, which will also increase the accuracy and make the job more efficient. Moreover, including other social media applications, to build better profile of criminals and influencers. Having more techniques of prediction and include different ways of predictions to increase accuracy, such as friend of friend, location of friends, location of the users. Having more analytic searching methods, like searching by certain personality. In addition, including more case studies by continuing searching and investigation, to have more trained data for machine learning.

ACKNOWLEDGEMENT

I would like to express my deepest sense of gratitude to.

- Dr.Damiani for his continuous supervision, guidance, encouragement and support throughout the project.
- Dr. Khaled Salahfor his guidance and exceptional support.
- Dr.Nawaf Al Moosa for his guidance and exceptional support.
- Mr. Benjamin Hirsch for his supervision, assistance and valuable technical support throughout the project
- Mr. Abdulrahman Al Remethi for his assistance and valuable technical support throughout the project.

I would like to take this opportunity to thank my colleagues at the university and the staff in the Information Security Department for their support.

REFERENCES

[1] “Twitter,” Sep 2014. [Online]. Available: http://ezproxy.kustar.ac.ae/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=ers&AN=87323158&site=eds-live. [Accessed 25 Jan 2017].

[2] “Law enforcement uses social media in investigations infographic - LexisNexis,” 2017. [Online]. Available: http://www.lexisnexis.com/risk/insights/law-enforcement-social-media-infographic.aspx.. [Accessed 28 Jan 2017].

[3] R. Li, K. H. Lei, R. Khadiwala and K. C.-C. Chang, “TEDAS: A Twitter-based Event Detection and Analysis System,” in 2012 IEEE 28th International Conference on Data Engineering, 2012.

[4] F. Chierichetti, J. Kleinberg and . R. Kumar, “Event Detection via Communication Pattern Analysis,” in AAAI Publications, Eighth International AAAI Conference on Weblogs and Social Media , 2014.

[5] C. Christopher and D. N. Tobin , “Terrorism and Crime Related Weblog Social Network,” Link Content analysis and information visualization, Vols. 1-4244-1330-3/07/ (2007), pp. 55-58, 2007.

[6] C. M. Zhang and V. Paxson, “Detecting and analyzing automated activity on twitter,” in In Proceedings of the 12th international conference on Passive and active measurement (PAM'11), Berlin, Heidelberg, 2011.

[7] J. Ratkiewicz, M. Meiss, B. Goncalves, D. Conover, F. Flammini and F. Menczer, "Detecting and Tracking Political Abuse in Social Media," in International AAAI Conference on Web and Social Media Fifth International AAAI Conference on Weblogs and Social Media, North America, 2011.

[8] L. D. Silva and E. Riloff, "User Type Classificatin of Tweets with Implications for Even Recognition," in 2014 ACL Joint Workshop on Social Dynamics and Personal Attributes in Social Media, Baltimore, MD, USA, 2014.

[9] J. Ramteke, S. Shah, G. Darshan and A. Shaikh, "Election result prediction using Twitter sentiment analysis," in Inventive Computation Technologies (ICICT), 2017.

[10] R. A. Bolla, Crime Pattern Detection Using Online Social Media, 2014.

[11] A. Bermingham and A. F. Smeaton, " On using Twitter to monitor political sentiment and predict election results," in n: Sentiment Analysis where AI meets Psychology (SAAIP) Workshop at the International Joint Conference for Natural Language Processing (IJCNLP), Chiang Mai, Thailand, 2011.

[12] M. Gerber, "Predicting Crime using Twitter and Kernel Density Estimation," Decision Support Systems (Elsevier) , vol. 61, no. http://dx.doi.org/10.1016/j.dss.2014.02.003, pp. 115-125 , 2014.

[13] R. YK Lau, M. Kamal H and M. I. Pramanik, Automatic Crime Detector: A Framework for Criminal Pattern Detection in Big Data Era., 2016.

[14] H. Sarvari, E. Abozinadah, A. Mbaziira and D. Mccoy, "Constructing and analyzing criminal networks.," in Security and Privacy Workshops (SPW), 2014.

[15] L. Kaati, A. Rezine and A. Berzinji, ""Detecting key players in terrorist networks.", 2012 European. I," in Intelligence and Security Informatics Conference (EISIC), 2012.

[16] C. Mascolo, Social and Technological Network Analysis Lecture 3: Centrality Measures, University of Cambridge.

[17] N. T. Inc., "Introduction to Graph Databases," 25 Mar 2017. [Online]. Available: https://neo4j.com/online_training/graphdatabases/?aliId=U2FmYWEgQWxkaGFuaGFuaS9zYWZhY S 5hbGRoYW5oYW5pQGt1c3Rhci5hYy5hZQ%3D%3D. [Accessed 2017].

[18] elasticsearch, "Logstash Introduction," 2017. [Online]. Available: https://www.elastic.co/guide/en/logstash/current/introduction.html#introduction.

[19] A. Khoori, "Man jailed for spreading rumors that harmed UAE on social media," 2017. [Online]. Available:http://www.thenational.ae/uae/man-jailed-for-spreading-rumours-that-harmed-uae-onsocial-media. [Accessed 28 Jan 2017].

[20] Twitter, 2017. [Online]. Available: https://dev.twitter.com/. [Accessed 2017].

[21] Chaffey, "Global socla media reach 2016," 2016. [Online]. Available:http://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-mediaresearch/attachment/screen-shot-2016-02-15-at-11-54-57. [Accessed 25 January 2017].

[22] "Arab Social Media Report," in Arab social media influencers summit, Dubai, 2015.

[23] A. Howard, A. Duffy, D. Freelon, M. Hussain, M. Mazaid and W. Mari, "Opening Closed Regimes: What Was the Role of Social Media During Arab Spring?".

[24] "A Facebook crime every 40 minutes: From killings to grooming as 12,300 cases are linked to the site," 2017. [Online]. Available: http://www.dailymail.co.uk/news/article-2154624/A-Facebook-crime-40-minutes-12-300-cases-linked-site.html. [Accessed 26 26 2017].

[25] M. Nati, "9 Violent Crimes Posted to Social Media - ODDEE", Oddee," 2017. [Online]. Available: http://www.oddee.com/item_99740.aspx. [Accessed 27 Jan 2017].

[26] J. Gifford, "Loggy," 2016. [Online]. Available: https://www.loggly.com/blog/scaling-elasticsearch-formulti-tenant-multi-cluster/. [Accessed 2017].

PREDICTIVE DETECTION OF KNOWN SECURITY CRITICALITIES IN CYBER PHYSICAL SYSTEMS WITH UNOBSERVABLE VARIABLES

Alessio Coletta[1,2]

[1]Security and Trust Unit, Bruno Kessler Foundation, Trento, Italy
[2]Department of Information Engineering and Computer Science,
University of Trento, Italy

ABSTRACT

A large number of existing Cyber Physical Systems (CPS) in production environments, also employed in critical infrastructures, are severely vulnerable to cyber threats but cannot be modified due to strict availability requirements and nearly impossible change management. Monitoring solutions are increasingly proving to be very effective in such scenarios. Since CPS are typically designed for a precise purpose, their behaviour is predictable to a good extent and often well known, both from the process and the cyber perspective. This work presents a cyber security monitor capable of leveraging such knowledge to detect illicit activities. It uses a formal language to specify critical conditions and an SMT-based engine to detect them through network traffic and log analysis. The framework is predictive, i.e. it recognises if the system is approaching a critical state before reaching it. An important novelty of the approach is the capability of dealing with unobservable variables, making the framework much more feasible in real cases. This work presents the formal framework and first experimental results validating the feasibility of the approach.

KEYWORDS

Security Monitoring, Detection and Prevention Systems, Critical Infrastructures, Cyber Physical Systems, SMT.

1. INTRODUCTION

Cyber Physical Systems (CPS) are composed by networked ICT devices that support the operation of physical entities. In this work we use CPS as general term that includes Industrial Control Systems (ICS), building automation systems, and the Internet of Things used for control and automation. The progressive use of ICT technology exposed CPS to vulnerabilities and threats typical of the ICT world [1]–[3]. Cyber Physical Systems present many specific differences from standard ICT systems [4] that make general ICT security solutions seldom effective for CPS. However, such peculiarities can also lead to better tailored solutions.

CPS are typically designed for a specific purpose in a predetermined production environment. As a consequence, the behaviour of their physical process is predictable to a good extent and often well documented. Fortunately, this predictability reflects on the cyber counterpart, thus it is possible to leverage such knowledge of the CPS to specify known critical conditions that combine

David C. Wyld et al. (Eds) : CNSA, DMDB, AIAP - 2018
pp. 61– 77, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80105

cyber and process aspects for a greater expressiveness and effectiveness. However, to the author's best knowledge, specification-based security monitoring approaches appear less mature than other approaches like anomaly-based techniques. This work presents a contribution in this regard.

It is necessary to observe the CPS current state to detect if it has reached a critical state. However, the assumption that every parameter of the CPS can always be observed is too strong and unfeasible in real cases. The main novelty of our approach is the ability to handle unobservable variables. Moreover, the framework is also capable of computing, if necessary, the piece of missing information required for a more accurate result. Such information is provided to security operators as a guide for finding a refinement of the CPS state. The refinement can feed the monitor back, leading to more precise detection.

Present paper improves our previous works [5] and present the same ability to predict whether the CPS is evolving towards some critical states, monitoring the changes in time of a notion of proximity from critical conditions. However, unobservable variables complicate the formal definition and the actual computation of the proximity, as explained in the following sections.
As previous works, the framework does not need a full model of the CPS, which is very hard to achieve in real cases. It is based on passive observations of the CPS through the analysis of network traffic and logs, to be more suitable for the industrial sector where change management and shutdowns are nearly impossible in practice, especially when employed in critical infrastructures. The framework presents an expressive specification language and is agnostic to observation methods and attack models, thus it is suitable for detecting possible 0-days attacks.

Section 2 shows a very simple example to explain the main idea behind the cyber security monitoring framework. Section 3 describes related works and approaches in literature and on the market. Section 4 defines our proposed framework, while Section 5 presents our first working prototype and our first experimental results that validates the approach.

2. A MOTIVATING EXAMPLE

This section presents an example of CPS which is overly simplified but still capable of explaining the kind of anomalies our framework is able to detect, its capability of handling unobservable aspects, and its notion of predictiveness. The following sections refer to this example to ease the explanation.

Figure 1 shows a simplified building automation system controlling the temperature of two rooms. The CPS is made of the following components ($i = 1,2$):

- a thermometer in each room that measures the temperature T_i
- an external thermometer for the outdoor temperature O
- a radiator R_i in each room that can be switched on/off
- a main water heater H that can be switched on/off.

Each room i has a setpoint S_i representing the desired target temperature of that room. Sensors (the thermometers) and actuators (the radiators and the heater) are wired to Programmable Logic Controllers (PLC). Each PLC is connected to the same TCP/IP-based Process Control Network (PCN). The Main Controller (MC), connected to the PCN, is able to send read and write commands to the PLCs. In this example we assume the Modbus is used [2], [6], a very widely used control protocol with no security mechanisms for authentication/authorisation.

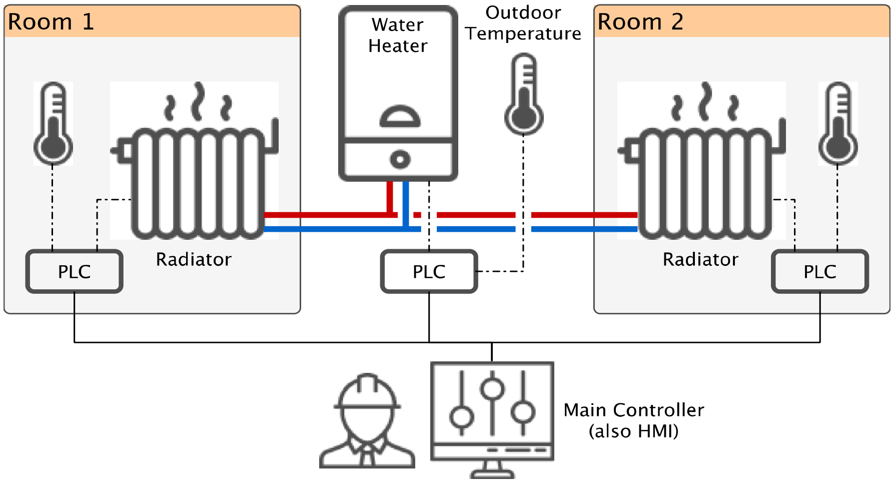


Figure 1. Two rooms building automation example.

In this example the main controller provides a Human Machine Interface (HMI) that allows an operator to visualise the current process parameters and to manually operate the system. An operator uses the HMI component of MC to: check temperatures and the on/off status of the radiators and of the heater; set the desired temperature of the rooms (setpoints); turn on/off the radiators and the heater. Besides manual human operations, our example CPS is automatically operated through a set of rules implemented in the main controller MC and listed in Table 1.

Table 1. Example of automatic operation rules.

every 500 ms	→	read T_i, S_i, R_i, H, O
$T_i < S_i$ and $O < S_i$	→	write $R_i =$ **true** and $H =$ **true**
$T_i > S_i$	→	write $R_i =$ **false**
$T_1 > S_1$ and $T_2 > S_2$	→	write $H =$ **false**

In this example the main controller is the only device that is supposed to send read and write Modbus commands to the PLCs. Suppose that an attacker (e.g. a malware) compromises the main controller and sends malicious Modbus commands to the PLCs from it. Such illicit commands would have exactly the same network signature and the same payload as the licit ones. Thus, neither signature-based IDS (e.g. Snort [7], [8], Suricata [9]) nor basic Modbus deep packet inspection tools (e.g. Wireshark [10]–[12], Bro [13]) can detect such anomalies. Indeed, the only way to detect them is to understand that such Modbus commands do not conform with the expected behaviour of the system.

Assume that the attacker sends read commands to the PLCs to gather and exfiltrate process information. Read commands and their responses are identical to the licit ones, however a network activity showing an unexpected read frequency can be considered illicit, expressed by the following *critical condition*

$$|F - f_{expected}| > \epsilon \qquad (\phi_1)$$

where F is the number of read commands per seconds *observed* from network capture, $f_{expected} = 3/0.5$ corresponds to 3 PLCs and 500 ms fromTable 1, and ϵ is a tolerance constant.

Similarly, suppose the attacker sends a Modbus write command $H =$ **false** to turn off the water heater to prevent room 1 or room 2 from reaching the desired temperature. Although this command is identical to the licit ones, it might be possible to detect such illicit activity when the presence of the write command does not conform with the expected behaviour from Table 1, expressed by the critical condition

$$\neg H \wedge ((T_1 < S_1 \wedge O < S_1) \vee (T_2 < S_2 \wedge O < S_2)) \qquad (\phi_2)$$

Intuitively, critical condition(ϕ_2) holds if a turn off command is sent when the heater should be on according to Table 1.

Notice that the critical condition (ϕ_1) purely addresses the cyber layer of the CPS, while (ϕ_2) mixes cyber and process aspects allowing for a greater expressiveness and effectiveness of the approach.

In this example a sensor might stop working, e.g. the outdoor thermometer, and the corresponding value might become unobservable. The novelty of our framework is the capability of handling both *observable* and *non-observable* variables, improving its range of applicability. Moreover, condition (ϕ_2) might not be critical if a human operator intentionally operates the CPS manually, e.g. for maintenance. A more accurate critical condition can be defined as

$$\neg M \wedge \neg H \wedge ((T_1 < S_1 \wedge O < S_1) \vee (T_2 < S_2 \wedge O < S_2)) \qquad (\phi_3)$$

where the boolean value M represents the manual and intentional operation of the human operator. The assumption that a monitoring tool is aware of human intentions is unreasonable in practical cases. Thus, M must be treated as unobservable, yet is necessary for a better accuracy.

If the example CPS is in a state where the critical condition (ϕ_1) is not satisfied, a notion of *proximity* from (ϕ_1) can be defined, representing how far the measured read command frequency is from its expected value. Monitoring how the proximity value changes in time enables to monitor if the CPS is approaching the critical condition (ϕ_1).

3. Related Work

One of the main source of vulnerability for CPS is the lack of security mechanisms in communication protocols, like authentication, authorisation, and confidentiality [2], [3]. Literature presents several secured version of control protocol, e.g. [14]–[16]. However, these security approaches rely on the possibility to redesign and replace at least some parts of the system, while for many industrial control systems downtimes and change management is not practical or affordable due to the high costs and risks related to any possible change. For this reason, redesign is often not an option and legacy components are often present. Passive and unobtrusive security measures are crucial for such CPS.

Intrusion Detection Systems (IDS) have been widely used in ICT security with good results. Signature-based IDS, like Snort [7], [8], are able to express *bad* IP packet that can be detected. Since cyber attacks are combinations of different licit-like actions and communications, signature-based IDS usually fall short in detecting complex attacks.

The *Anomaly-based* intrusion detection approach has proved effective for CPS cyber security [17]–[22]. [23] classifies anomaly-based IDS in two main categories:

i. *unattended techniques*, leveraging statistical models or machine learning to create a baseline representing licit behaviours that are compared with the run-time observations

ii. *specification-based techniques*, for which a human ICS expert precisely defines what is licit or anomalous in a specification language, and the detection tool compares the state of the monitored system against such specifications.

The absence of human effort is a good advantage of the unattended techniques, but they suffer from high false positive rates which requires human effort to spot false alarms. Our work focuses on the specification-based approach, with the advantage that false positive rates are extremely low or even zero when enough knowledge of the system is available. The main drawback is the effort required to define the known critical conditions. However, CPS typically shows predictable and repeatable behaviours over time. Moreover, the design phase of a critical infrastructure is detailed and documented, providing valuable knowledge to be modelled. Nonetheless, some approaches to automatically derive specifications from the monitored system have proved effective, e.g. [24]. For this reason, specification-based techniques seem to be a good approach for developing security monitors for CPS.

Security monitoring has gained relevance in the Security Operation Centres (SOC) of big organizations and in the DevOps sector. Wide spread frameworks includes Splunk [25], [26], Elasticsearch-Logstash-Kibana (ELK) [27]–[30], Grafana [31], and LogRythm [32]. Such tools continuously collect log events and time series data (e.g. cpu load, memory consumption, etc.). Security operators can customise visualisation dashboards of such information to spot anomalous vs. normal behaviours in a graphical way. Moreover, security operators can define custom alarms specifying queries on the collected data and events, for instance to detect known indicator of compromise (IoC). The possibility to define alarms is somehow similar to our notion of critical condition described in this paper. Unlike our proposed framework, such tools allow queries only on observable data and do not offer a notion of proximity / proximity range from criticality.

Nai et al. [33]–[35] developed a specification-based Intrusion Detection and Prevention System methodology specific for SCADA systems that is not based on specific attack models and can detect 0-day attacks. The methodology allows combining the knowledge of the physical process with the cyber behaviour to monitor, and is further extended in [5] with a greater expressiveness and more effective computation methods. Our present work further improves the same approach. The novelty of this work consists in (i) dealing with unobservable aspects of the system for a greater expressiveness and feasibility in real cases (ii) using real-time knowledge refinements from human operators (iii) guiding the operator to express better refinements (iv) computing proximity and proximity ranges from criticality even in presence of unobservable variables.

4. THE MONITORING FRAMEWORK

The proposed monitoring framework passively runs in parallel with the monitored CPS. It continuously observes the current state of the CPS and checks it against conditions that are a-priori known to be illicit or anomalous, hereafter called *critical conditions*. Figure 2depicts the main structure of the framework.

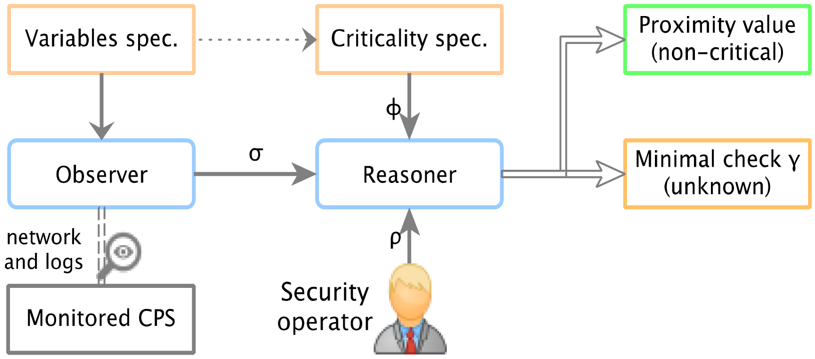


Figure 2. Structure of the real-time monitoring framework.

The first step is to identify the aspects of the CPS, called *variables*, that are necessary to express the critical conditions. In real cases, the assumption that it is always possible to retrieve the value of all the variables is too strict and unfeasible. Thus, a variable can be *observable* or *unobservable*, either temporarily or permanently. Unobservable variables complicate the framework but allow for a greater expressiveness and practical feasibility. There are three main cases in which a variable is considered unobservable:

i. a variable bound to the value of a malfunctioning sensor that cannot provide its value;

ii. a variable bound to a parameter of the CPS which is required to express the critical condition but that can never be observed by design, e.g. the temperature of a gas in a point where no thermometer has been installed;

iii. any aspect of the monitored system that is inherently unobservable, e.g. the intention of a human operator that acts without specifying his actions in advance.

The monitoring framework is composed by two main components: the *observer* and the *reasoner*. The former continuously analyses network traffic and logs generated by the monitored CPS, in order to retrieve the value of the observable variables. The latter checks the current state of the CPS against the set of known critical conditions.

The input to the observer consists of the *specification of variables*, which enumerates the variables of interest and their properties. Precisely it defines for each variable:

i. the *name*, used as an identifier in the specification of critical conditions

ii. the *type*: boolean, integer, or real

iii. an optional *range constraint*, i.e. lower and upper bounds

iv. an optional *observation method*: how the observer captures the value of the variable through network or log analysis. When the method fails or is not provided, the variable is considered unobservable.

This paper is agnostic w.r.t. observation methods, so any method can be used. The only constraint is that the observation needs to be nearly real-time and the result must be a boolean or numeric value. For instance, in our prototype the observer uses deep packet inspection against network traffic captured in real-time.

Our threat model assumes the integrity of the observed values: if an attacker takes the complete control of the network it might compromise the effectiveness and correctness of our monitoring framework. However, this assumption is typical of security monitoring solutions cited in Section 3. In real cases, such approach is still valid provided that a sufficient large number of variables are observable and effective critical conditions are specified. In this way the likelihood that an attacker is able to compromise enough values to make the monitor ineffective is low.

Iteratively the reasoner gets as input the values from the observer (σ) and a set of critical conditions (ϕ), and checks if each condition is met with the current observations. If the critical condition only contains observable variables, the reasoner is always able to tell whether the CPS has reached the criticality or not. In presence of unobservable variables it might be impossible to discriminate whether the CPS is in a critical state only from observations.

The reasoner is also able to take as input some further information about the CPS state in form of a logical assertion, hereafter called *refinement* and denoted by ρ. Refinements are typically provided by human operators to give the monitor additional information about unobservable variable.

Each critical condition is associated to a function that represents a notion of *proximity*. When the reasoner is able to determine that CPS is currently not in a critical condition, it computes the proximity from that condition.

When the reasoner is unable to determine whether the current state satisfies a critical condition, it computes the minimal condition of unobservable variables that is necessary to determine that the system state is not critical (γ inFigure 2). The minimal condition γ is hereafter called *assisted* *check*, because it helps security operators figure out the missing unobservable information. In other words, the assisted check can guide operators to provide better knowledge refinements.

4.1 Specification of Variables and Critical Conditions

Let $\mathcal{V}$ denote the set of variables, whose type can be boolean, integer, or real, and let $range(v)$ denote the range constraint of v defined in the variable specification. Boolean variables range on the set $\{0,1\}$, with both the boolean and the numeric meaning, in order to be able to use boolean and numeric variables in the same arithmetic expressions. As a consequence, all variables in $\mathcal{V}$ range on $\mathbb{R}$.

Definition 1. Let $V \subseteq \mathcal{V}$ be a subset of variables. A *partial assignment* (or simply an assignment) is a function $a: V \to \mathbb{R}$ that maps variables to their value such that $a(v) \in range(v)$ for each variable $v \in V$. The notation $dom(a)$ denotes its domain V.

Definition 2. A *state* of the monitored CPS (or simply a state) is an assignment s such that $dom(s) = \mathcal{V}$, i.e. a total assignment on variables. The set of all possible system states is denoted by $\mathcal{S}$. Given a partial assignment a, we define

$$\mathcal{S}(a) = \{s \in \mathcal{S} \mid \forall v \in dom(a): s(v) = a(v)\}$$

as the set of states that satisfy the assignment a.

Our framework uses a partial assignment c to represent the current observations that the observer passes to the reasoner. If all variable are observable, c is total, i.e. c is a CPS state. In case of unobservable variables, c represents only the observable portion of the current state of the CPS, and from the perspective of the reasoner the CPS could be in any state of $\mathcal{S}(c)$.

Definition 3. A *state formula* is defined by the grammar:

$$\phi ::= a_1 v_1 + \cdots + a_n v_n \bowtie b \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi$$

where $v_i \in \mathcal{V}$, $a_i, b \in \mathbb{R}$, $\bowtie \in \{<, \leq, >, \geq, =, \neq\}$. The set of variables occurring in a formula ϕ is denoted by $var(\phi)$.

A state formula is a boolean combination of linear inequalities of variables and expresses a property of the CPS state, where both observable and unobservable variables may occur in a formula. We use the standard interpretation of formulae over assignments.

Definition 4. Given a partial assignment c and a formula ϕ such that $var(\phi) \subseteq dom(c)$, the assignment *csatisfies* (or *models*) the formula ϕ, denoted by $c \vDash \phi$, when recursively:

$$\begin{array}{rlcrlc} c \vDash \sum_i a_i v_i \bowtie b & \text{iff} & \sum_i a_i c(v_i) \bowtie b & c \vDash \neg\phi & \text{iff} & c \nvDash \phi \\ c \vDash \phi_1 \wedge \phi_2 & \text{iff} & c \vDash \phi_1 \text{ and } c \vDash \phi_2 & c \vDash \phi_1 \vee \phi_2 & \text{iff} & c \vDash \phi_1 \text{ or } c \vDash \phi_2 \end{array}$$

The set of states satisfying a formula ϕ is denoted by $\mathcal{S}(\phi)$.

Our framework uses state formulae to define the known critical conditions of the monitored CPS. Moreover, at each iteration the observer passes the receiver the formula

$$\sigma := \bigwedge_{v \in V} v = c(v)$$

representing the current observation, where V is the set of observed variables and $c(v)$ is their observed value.

Example. The simple example described in Section 2 uses the following variables:

- *T_1, T_2, S_1, S_2, O: real variables for internal temperatures, desired temperatures (setpoints), and outdoor temperature*

- *H: boolean variable corresponding to the on/off status of the heater*
- *M: boolean unobservable variable corresponding to the fact that the operator has manually and intentionally operated the CPS through the HMI.*

Variables T_i, S_i, O, H are observed through the analysis of the Modbus traffic on the process network.

4.2 Criticality Detection

Error! Reference source not found.depicts the behaviour of the reasoner at each iteration. For each critical formula ϕ, the reasoner uses the formula σ from the observer and an optional refinement ρ of the CPS state (if provided) to discriminate whether the monitored CPS has reached ϕ. To this aim, a formula is defined as

$$\kappa \coloneqq \sigma \wedge \rho$$

that represents all the information about the CPS state s available to the reasoner, i.e. that $s \in \mathcal{S}(\kappa)$.

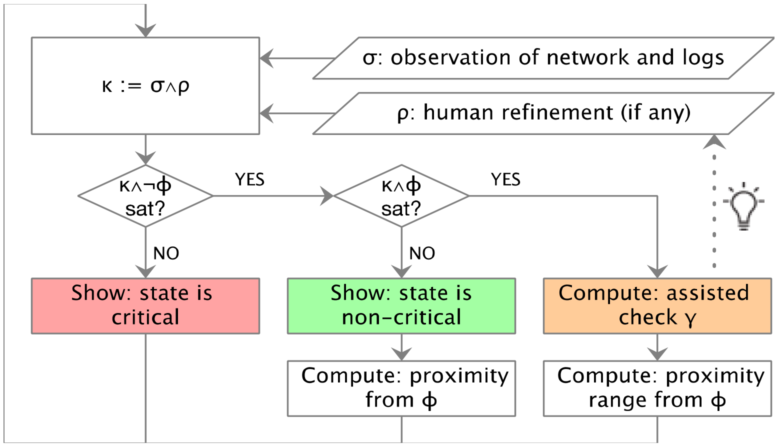

κ := σ∧ρ
σ: observation of network and logs
ρ: human refinement (if any)
κ∧¬ϕ sat?
YES
κ∧ϕ sat?
YES
NO
NO
Show: state is critical
Show: state is non-critical
Compute: assisted check γ
Compute: proximity from ϕ
Compute: proximity range from ϕ

Figure 3. Reasoner flow chart given criticalityϕ.

To discriminate if the CPS is currently in a critical state the reasoner checks whether the formulae $\kappa \wedge \phi$ and $\kappa \wedge \neg\phi$ are *satisfiable* using an SMT solver. Three cases are possible:

i. The system *is in a critical state*, regardless unobservable values, or equivalently $\mathcal{S}(\kappa) \cap \mathcal{S}(\phi)^C = \emptyset$. Similarly, this is equivalent to checking whether the formula

$$\kappa \wedge \neg\phi \text{ is unsatisfiable.} \tag{1}$$

ii. The system *is not in a critical state* regardless unobservable values, or equivalently $\mathcal{S}(\kappa) \cap \mathcal{S}(\phi) = \emptyset$. This is equivalent to checking whether formula

$$\kappa \wedge \phi \text{ is unsatisfiable.} \tag{2}$$

iii. If both formulae in (1) and (2) are satisfiable, then $\mathcal{S}(\kappa) \cap \mathcal{S}(\phi) \neq \emptyset$ and $\mathcal{S}(\kappa) \backslash \mathcal{S}(\phi) \neq \emptyset$. In other words, it is not possible to establish from κ whether the actual CPS state is critical, because this depends on some unobservable values not in κ.

The last case means that κ does not contain enough information to discriminate the criticality of the CPS state. Since the observation σ does not contain information about unobservables by definition, the only way to obtain a more precise result is to provide a more informative refinementρ.

In practical cases it can be hard for a human operator to understand which piece of information is missing. To this aim, the reasoner is able to calculate a minimal condition, hereafter denoted by γ, that is sufficient to guarantee the non criticality of the current CPS state given κ and ϕ, i.e. such that $\kappa \wedge \gamma \wedge \phi$ is not satisfiable. Our monitoring solution shows γ to a human operator, who can try to manually check the CPS in order to verify if γ holds, or at least if some of the sub-formulae ofγ hold. This way the operator may acquire some information and make educated assumptions on unobservable variables, and provide it back to the reasoner in the form of a more informative refinement. For this reason the reasoner acts as an assistant to the human operator, and the formula γ is called *assisted check*. In practical cases, the operator must be able to handle the complexity of the assisted check, thus it is crucial that γ is minimal.

We use the notion of interpolant, provided by most SMT solvers, to compute the minimal assisted check γ. Given two mutually unsatisfiable formulae α and β, a *Craig interpolant* (denoted by $interpolant(\alpha, \beta)$) is a formula η such that $var(\eta) \subseteq var(\alpha) \cap var(\beta)$ and formulae $\alpha \rightarrow \eta$ and $\eta \rightarrow \neg\beta$ are valid. In other words, the formula η is an explanation for the mutual unsatisfiability that uses only the variables that are common in α and β.

Our framework also uses syntactic simplification of logical expressions that most SMT solvers provide. Hereafter $simplify(\alpha)$ denotes the computation of a possibly simpler expression equivalent to α. The analysis of the most effective simplification tactics goes beyond the aim of this work[1] and does not compromise the soundness of our approach.

Since formulae $\kappa \wedge \neg\phi$ and $\kappa \wedge \phi$ are mutually unsatisfiable, the assisted check can be defined as

$$\gamma \coloneqq interpolant(simplify(\kappa \wedge \neg\phi), simplify(\kappa \wedge \phi)) \tag{3}$$

Example. Assume that the example CPS of Section 2 reaches a state where the room temperatures are $15\,^\circ C$ and $23\,^\circ C$, the desired temperatures are $21\,^\circ C$ and $17\,^\circ C$, the main controller sends a Modbus write message to the PLC controlling the water heater to turn it off, and the outdoor thermometer is temporarily broken (i.e. O is unobservable). The observer collects such information from the network traffic and provides the reasoner with

$$\sigma \coloneqq \neg H \wedge T_1 = 15 \wedge S_1 = 21 \wedge T_2 = 23 \wedge S_2 = 17 \tag{4}$$

Assume no further refinement is provided, i.e. $\kappa = \sigma$. In this example, our framework can verify that both formulae $\kappa \wedge \phi_3$ and $\kappa \wedge \neg\phi_3$ are satisfiable. Indeed, the current knowledge κ does not contain enough information to tell if the CPS is in a critical state, because this depends on the actual value of the unobservable variables M and O. In this case, the framework computes the assisted check

$$\gamma \coloneqq M \vee O \geq 21$$

Indeed, in order to discriminate the criticality of the current state it is enough to check if the operator intentionally sent the command or if the outdoor temperature is greater thanS_1, which is 21. Notice that γ is minimal, i.e. it only contains the unobservable variables. If the operator manually operated the system, he/she can provide the refinement $\rho \coloneqq M$.

4.3 Predictiveness: Proximity from Critical Conditions

In this section we define the notion of proximity from a critical condition ϕ. Given a set X, a function $d: X \times X \rightarrow \mathbb{R}$ is called *premetric* if both $d(x, y) \geq 0$ and $d(x, x) = 0$ for all $x, y \in X$. Given a set X, a premetric function $d: X \times X \rightarrow \mathbb{R}$ is called a *metric* if for all $x, y, z \in X$: (i)

[1] As a reference, our prototype uses Z3 [36] with the tactic `(then simplify ctx-simplify ctx-solver-simplify)`.

$d(x, y) = 0$ iff $x = y$, (ii) $d(x, y) = d(y, x)$, (iii) $d(x, y) \leq d(x, z) + d(z, y)$. The pair (X, d) is called *metric space*.

We use the following well known result. Let (X, d) be a metric space. The function $D: 2^X \times 2^X \to \mathbb{R}$ defined as

$$D(A, B) = \inf_{a \in A,\, b \in B} d(a, b)$$

is a premetric.

Provided any enumeration of the CPS variables $\mathcal{V}$, the set of states $\mathcal{S}$ can also be seen as a vector of $\mathbb{R}^n$, where n is the number of variables. Thus, any metric d on $\mathbb{R}^n$ is a metric on $\mathcal{S}$ that induces a premetric D on $2^{\mathcal{S}}$. In the following we use the premetric D to capture the notion of proximity from critical condition.

Our framework requires to specify for each critical condition ϕ an associated metric d. Recall that at runtime the formula $\kappa \coloneqq \sigma \wedge \rho$ represents what the reasoner knowns about CPS variables. The *proximity* of the current CPS state from the critical condition ϕ is defined as

$$D(\mathcal{S}(\kappa), \mathcal{S}(\phi))$$

hereafter denoted by $D(\kappa, \phi)$.

Previous definition is parametric w.r.t. the chosen metric on the set of states $\mathcal{S}$, and the actual choice function depends on the application. Table 2shows possible examples of metrics. For instances, the Hamming distance captures the number of variables that differs, while the Manhattan distance captures each variable variation, and this choice allows for a qualitative vs. quantitative proximity notion.

Table 2. Example of metrics on $\mathcal{S}$.

$d(s, t) = \sum_{v \in V} \mid s(v) - t(v) \mid$	Manhattan distance (i. e. L_1 metric on $\mathbb{R}^n$)
$m_V(s, t) = \frac{1}{\#V} \sum_{v \in V} \frac{\lvert s(v) - t(v) \rvert}{v_{\max} - v_{\min}}$	Normalised Manhattan distance (defined if $v_{\min}, v_{\min} \in \mathbb{R}$)
$h_V(s, t) = \#\{v \in V \mid s(v) \neq t(v)\}$	Hamming distance
$nh_V(s, t) = \frac{1}{\#V} \cdot \#\{v \in V \mid s(v) \neq t(v)\}$	Normalised Hamming distance

where $s, t \in \mathcal{S}$, $V \subseteq \mathcal{V}$, $v_{\min} = \min(range(v))$, $v_{\min} = \max(range(v))$.

When the current CPS state is critical, i.e. $\kappa \wedge \neg\phi$ is unsatisfiable, proximity $D(\kappa, \phi) = 0$. When the CPS is in a critical state, i.e. when $\kappa \wedge \phi$ is unsatisfiable, computing the proximity from the critical condition $D(\kappa, \phi)$ is an optimisation problem on linear constraints, since critical formulae κ and ϕ represent boolean combination of linear inequalities. Our framework uses SMT-based optimisation techniques, such as the one provided by the Z3 prover [36] and by OptiMathSat [37]. Figure 4shows the pseudo-algorithm to compute the proximity $D(\kappa, \phi)$ given the current knowledge of the system κ and the criticality expressed by the formula ϕ.

```
function Proximity($\kappa$, $\phi$)
    (S, T) ← two distinct sets of fresh symbols for variables
    distance ← new real symbol
    solver ← new SMT-Optimizing-Solver
    solver.assert $\kappa[\mathcal{V} \mapsto \mathrm{S}] \wedge \phi[\mathcal{V} \mapsto \mathrm{T}]$
    solver.assert $\mathrm{distance} = \mathrm{metric}(\mathrm{S}, \mathrm{T})$
    solver.goal ← minimize(distance)
    model ← solver.check-sat()
if model not foundthen
return Error: either $\kappa$ or $\phi$ is unsatisfiable
else
returnmodel.getvalue(distance)
```

Figure 4. Proximity pseudo-algorithm.

When the reasoner is not able to discriminate that the CPS is not in a critical condition, i.e. when both $\kappa \wedge \phi$ and $\kappa \wedge \neg\phi$ are satisfiable, it is necessary to compute a notion of *proximity range* from the critical condition ϕ that handles the missing information about unobservable variables, defined as the pair $(D_{\min}, D_{\max})$ where

$$\begin{aligned} D_{\min} &= \inf_{s \in \mathcal{S}(\kappa) \setminus \mathcal{S}(\phi)} D(\{s\}, \phi) = \inf_{\substack{s \vDash \kappa \wedge \neg\phi \\ t \vDash \phi}} d(s, t) \\ D_{\max} &= \sup_{s \in \mathcal{S}(\kappa) \setminus \mathcal{S}(\phi)} D(\{s\}, \phi) = \sup_{s \vDash \kappa \wedge \neg\phi} \inf_{t \vDash \phi} d(s, t) \end{aligned}$$

Computing $D_{\min}$ corresponds to the same optimisation problem on linear constraints as before, while computing $D_{\max}$ requires to iteratively search the maximum results of the same optimization problem, increasing the distance until no result if found. Figure 5shows the pseudo-algorithm to compute the proximity range $(D_{\min}, D_{\max})$.

```
functionProximityRange($\kappa$, $\phi$)
    (S, T) ← two distinct sets of fresh symbols for variables
    distance ← new real symbol
    solver ← new SMT-Optimizing-Solver
    solver.assert $\kappa[\mathcal{V} \mapsto \mathrm{S}] \wedge \neg\phi[\mathcal{V} \mapsto \mathrm{S}]$
    solver.assert $\phi[\mathcal{V} \mapsto \mathrm{T}]$
    solver.assert $\mathrm{distance} = \mathrm{metric}(\mathrm{S}, \mathrm{T})$
    solver.goal ← minimize(distance)
    model ← solver.check-sat()
if model not found then
return Error: either $\kappa \wedge \neg\phi$ or $\phi$ are unsatisfiable
else
mindistance← model.getvalue(distance)
repeat
maxdistance← model.getvalue(distance)
for$v \in \mathcal{V}$do
solver.assert T($v$) ≠ model.getvalue(T($v$))
solver.assert$\mathrm{distance} > \mathrm{maxdistance}$
model ← solver.check-sat()
until model is not found
return (mindistance, maxdistance)
```

Figure 5. Proximity range pseudo-algorithm.

Example. Assume κ is defined as (4) in the previous example. Recall that both $\kappa \wedge \phi_3$ and $\kappa \wedge \neg\phi_3$ are satisfiable. Using the Hamming distance $h_{\mathcal{V}}$ on the seven variables T_i, S_i, O, H, and M, the value of the proximity and the proximity range from ϕ_3 are

$$D(\kappa, \phi_3) \quad = 0 \quad D_{min} \quad = \frac{1}{7} \quad D_{max} \quad = \frac{2}{7}$$

The fact that the proximity $D(\kappa, \phi_3) = 0$ is correct, since the current CPS state could be critical depending on the actual value of the unobservable variables M and O. For this reason, the reasoner computes the proximity range instead of the proximity. Proximity range gives pessimistic and optimistic estimation of the proximity, under the assumption that the current state is not critical: values 1/7 and 2/7 correctly indicates that the number of variables (out of seven) required to change before reaching ϕ_3 is 1 or 2 respectively.

5. EXPERIMENTAL RESULTS

This section briefly shows how our first prototype is implemented and the first experimental results that prove the feasibility of the approach.

We set up a Docker-based [38] simulation of the CPS example described in Section 2, and specifically the Process Control Network and the Modbus traffic between the main controller and the PLCs, with the following main containers:

- **plcsim**: our python application simulating the PLCs and the physical process. The Modbus interface is implemented using the `pymodbus` Python library [39]. Physics is simulated using the Newton's Law of Cooling, often used in literature (e.g. [40]).

- **nodered**: the HMI, the human manual operations, and the automatic operation logic of Table 1, and the attackers command are implemented using *Node-RED* [41], a flow-based programming tool that supports the Modbus protocol. This way licit and illicit Modbus commands are identical w.r.t their packet signature.

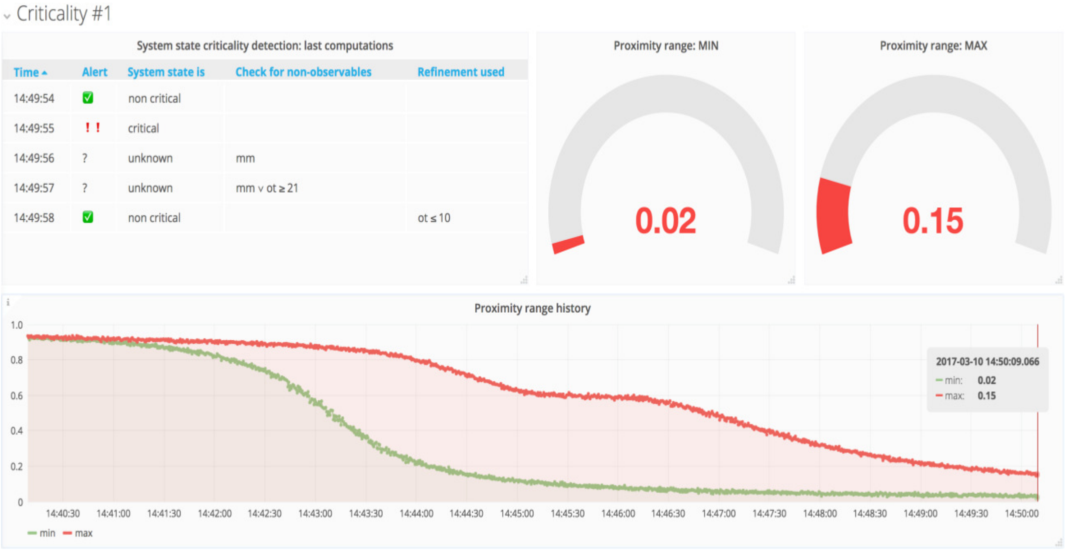
- **monitor**: our Python prototype of the proposed monitoring framework, which detects critical conditions and computes the proximity and proximity range from them using the SMT open source Z3 prover [36].

- **influxdb** [42] and **grafana** [31]: a time series database and a data visualisation software that provide the graphical user interface. Figure 6 shows a screenshot.

In this way the environment is able to simulate the main components depicted in Figure 1and the Process Control Network that exhibits a real Modbus network traffic that is possible to capture and analyse. For instance, the observer component collects the value of variable T_1 by monitoring the Modbus with destination IP of PLC1, port 502, and inspects such traffic to extract from the payload the value of input register (the Modbus term indicating a read-only integer register) corresponding to T_1. The observer sends the value to the reasoner via the local MQTT server on the `sensor/T1` topic.

The observer uses open source tools to monitor the network and system logs, according to variable specification. In particular, the Modbus network traffic is analysed through tshark, the command line tool of Wireshark [10], to extract the values of interest through its basic deep packet inspection functionalities. The observer and the reasoner are two distinct pieces of software that communicate through the MQTT sub/pub protocol [43] (QoS 1).

The reasoner component is a Python application that implements Figure 3. It iteratively gathers the observed values from the observer and, for each critical condition in the criticality specification, it uses the open source SMT Z3 solver [36] with the Python API to evaluates the criticality of the CPS state, to compute the minimal assisted check defined in (3), and to compute proximity and proximity range from the critical condition as Figure 4 and Figure 5.



⌄ Criticality #1

System state criticality detection: last computations

Time ▲	Alert	System state is	Check for non-observables	Refinement used
14:49:54	✓	non critical		
14:49:55	! !	critical		
14:49:56	?	unknown	mm	
14:49:57	?	unknown	mm ∨ ot ≥ 21	
14:49:58	✓	non critical		ot ≤ 10

Figure 6. Screenshot of the graphical interface of the monitoring prototype.

Figure 6 shows our first implementation of the graphical user interface to provide security operators with the real-time results of our framework, given a certain critical condition ϕ. The first block, in tabular form, shows different moments of the recent history. The first column shows the time of the computation of the reasoner. The third column contain the value "critical", "non critical", or "unknown" representing the three cases described in Section 4.2 and in the flow chart in Figure 3. The forth column is empty in case of "critical" and "non critical", and contains the minimal assisted check γ described in Section 4.2. The fifth column contains the user refinement ρ, if provided.

The rest of the graphical dashboard shows the proximity range[2]. The two gauges represent the real-time values of $D_{\min}$ and $D_{\max}$. The chart shows the timeseries of $D_{\min}$ and $D_{\max}$. It is easy to see that the values are constantly decreasing, thus the system is getting closer to the critical condition. In the leftmost part of the chart $D_{\min}$ and $D_{\max}$ are both close to 1, hence the current state of the monitored CPS is distant from the critical states and unobservable variables have a low impact. The central part of the chart shows that $D_{\min}$ and $D_{\max}$ greatly differ each other: this means that unobservable variables have a bigger role on the actual proximity of the CPS from the critical condition. This is the case when a refinement from the human operator can really improve the results of the reasoner. The rightmost part of the chart shows that the system is close to the criticality, because both $D_{\min}$ and $D_{\max}$ are close to 0.

The whole prototype works on an Intel Core i7 laptop with 8 GB or RAM, and it is capable of discriminating the criticality and compute proximity ranges at real-time with an update frequency of about 500 milliseconds, which seems enough for a security monitoring solution. While better performance tests and a characterisation of the attacks and critical conditions are subject of further investigation, first results seem to validate the overall feasibility of the approach.

[2] Values shown here are only for example purposes and for a better graphical explanation.

6. Final Remarks

This work presents a specification-based predictive cyber security monitoring framework for cyber physical systems and improves [5]. It enables specifying known critical conditions, through an easy but expressive formal language, that can be detected at run-time. It defines a notion of proximity of the CPS current state from the specified critical states: checking how the proximity changes in time enables security operators to predict whether the system is evolving towards critical states and how close it is from them.

The novelty of present work is to handle both observable and unobservable aspects of the CPS. This enables a security operator to express a model of criticality that is more complete and suitable for real cases. The monitor is able to continuously gather the value of all the observable variables from the analysis of the network traffic analysis and system logs, and to build a representation of this knowledge that correctly approximates the actual state of the system.

Unobservable variables complicate the criticality detection. When the monitor cannot discriminate if the CPS is in a critical state, a human operator can provide additional knowledge about unobservable variables as a refinement. However, this can be hard in real cases due to the complexity of the CPS and the large number of variables. To this aim, the framework is capable of computing the minimal piece of information that is required to discriminate the criticality of the CPS state, and provide such information as a guide to the operator.

Unobservable variables also complicate computing the proximity from critical states. However, the framework is able to compute a min/max range of the distance from critical states. Our working prototype also presents a graphical interface showing the history of the proximity range, providing an overview of the evolution of the system w.r.t. the specified critical conditions.

This work uses SMT techniques to assess the criticality of the CPS current state and to compute the minimal assisted checks. It also uses SMT-based optimisation techniques to compute proximity ranges from critical states. Preliminary results proves an expressive specification language and an efficient reasoning engine. While first results seem feasible an promising, precise experiments will be the subject of further investigation to assess the limits of our approach.

Another aspect to further improve is the characterization of cyber metrics and aggregate functions that can be leverage in the monitoring framework. In particularly, security monitoring common in the DevOps sector provide a plethora of metrics and functions that can be integrated in our framework. The framework only permits to specify linear constraints and distances computable through linear optimisation problems. Another subject to further investigate are recent works about satisfiability modulo linear and non-linear theories over reals [44]–[48].

References

[1] V. M. Igure, S. A. Laughter, and R. D. Williams, "Security issues in SCADA networks," *Computers & Security*, vol. 25, no. 7, pp. 498–506, Oct. 2006.

[2] P. Huitsing, R. Chandia, M. Papa, and S. Shenoi, "Attack taxonomies for the Modbus protocols," *International Journal of Critical Infrastructure Protection*, vol. 1, pp. 37–44, Dec. 2008.

[3] S. East, J. Butts, M. Papa, and S. Shenoi, "A Taxonomy of Attacks on the DNP3 Protocol," in *Critical infrastructure protection iii*, 2009, pp. 67–81.

[4] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, "Guide to Industrial Control Systems (ICS) Security," National Institute of Standards; Technology, Gaithersburg, MD, Jun. 2015.

[5] A. Coletta and A. Armando, “Security Monitoring for Industrial Control Systems,” in *Security of industrial control systems and cyber physical systems. CyberICS 2015*, 2016, pp. 48–62.

[6] “MODBUS Application Protocol Specification V1.1b3,” 2012.

[7] M. Roesch, “Snort: Lightweight Intrusion Detection for Networks.” *LISA ’99: 13th Systems Administration Conference*, pp. 229–238, 1999.

[8] B. Caswell and J. Beale, *Snort 2.1 intrusion detection*. Syngress, 2004.

[9] Suricata, “Suricata Open Source IDS / IPS / NSM engine.” 2017.

[10] G. Combs and Others, “Wireshark,” *www.wireshark.org*, 2017.

[11] A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal network protocol analyzer toolkit*. Syngress, 2006.

[12] C. Sanders, Practical packet analysis: Using Wireshark to solve real-world network problems. No Starch Press, 2011.

[13] V. Paxson, “Bro: a system for detecting network intruders in real-time,” *Computer Networks*, vol. 31, nos. 23-24, pp. 2435–2463, 1999.

[14] I N. Fovino, A. Carcano, M. Masera, and A. Trombetta, “Design and Implementation of a Secure Modbus Protocol,” in *International conference on critical infrastructure protection*, 2009, pp. 83–96.

[15] M. Majdalawieh, F. Parisi-Presicce, and D. Wijesekera, “DNPSec: Distributed network protocol version 3 (DNP3) security framework,” in *Advances in computer, information, and systems sciences, and engineering*, Springer, 2007, pp. 227–234.

[16] G. Gilchrist, “Secure authentication for DNP3,” in IEEE power and energy society general meeting - conversion and delivery of electrical energy in the 21st century, 2008, pp. 1–3.

[17] D. Bolzoni, S. Etalle, P. Hartel, and E. Zambon, “POSEIDON: a 2-tier Anomaly-based Network Intrusion Detection System,” in *Fourth ieee international workshop on information assurance (iwia)*, 2006.

[18] W. Heimerdinger, V. Guralnik, and R. VanRiper, “Anomaly-based intrusion detection.” Google Patents, 2006.

[19] C. Zimmer, B. Bhat, F. Mueller, and S. Mohan, “Time-based intrusion detection in cyber-physical systems,” *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems - ICCPS ’10*, p. 109, 2010.

[20] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, “Using Model-based Intrusion Detection for SCADA Networks,” *Science And Technology*, vol. 329, pp. 1–12, 2006.

[21] R. Mitchell and I. R. Chen, “Behavior Rule Specification-Based Intrusion Detection for Safety Critical Medical Cyber Physical Systems,” *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 16–30, 2015.

[22] K. Xiao *et al.*, “A Workflow-Based Non-intrusive Approach for Enhancing the Survivability of Critical Infrastructures in Cyber Environment,” in *Third international workshop on software engineering for secure systems (sess’07: ICSE workshops 2007)*, 2007, pp. 4–4.

[23] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” *Computers & Security*, vol. 28, nos. 1-2, pp. 18–28, Feb. 2009.

[24] M. Caselli *et al.*, "Specification Mining for Intrusion Detection in Networked Control Systems Specification Mining for Intrusion Detection in Networked Control Systems," *Proceedings of the 25th USENIX Security Symposium*, pp. 791–806, 2016.

[25] D. Carasso, *Exploring Splunk*. CITO Research, 2012.

[26] J. Diakun, P. R. Johnson, and D. Mock, *Splunk Operational Intelligence Cookbook*. Packt Publishing Ltd, 2016.

[27] C. Gormley and Z. Tong, *Elasticsearch: the Definitive Guide*. O'Reilly Media, Inc., 2015.

[28] J. Turnbull, *The Logstash Book*. James Turnbull, 2013.

[29] Y. Gupta, *Kibana Essentials*. Packt Publishing Ltd, 2015.

[30] G. S. Sachdeva, *Practical ELK Stack*. Apress, 2017.

[31] Grafana Labs, "Grafana." 2017.

[32] LogRhythmInc, "LogRhythm security intelligence and analytics platform." 2017.

[33] I. NaiFovino, A. Coletta, A. Carcano, and M. Masera, "Critical State-Based Filtering System for Securing SCADA Network Protocols," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 10, pp. 3943–3950, Oct. 2012.

[34] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. NaiFovino, and A. Trombetta, "A Multidimensional Critical State Analysis for Detecting Intrusions in SCADA Systems," *IEEE Transactions on Industrial Informatics*, 2011.

[35] I. NaiFovino, A. Carcano, A. Coletta, M. Guglielmi, M. Masera, and A. Trombetta, "State-Based Firewall for Industrial Protocols with Critical-State Prediction Monitor," in *Critical information infrastructures security*, vol. 6712 LNCS, 2011, pp. 116–127.

[36] L. De Moura and N. Bjørner, "Z3: An efficient SMT solver," in International conference on tools and algorithms for the construction and analysis of systems, 2008, pp. 337–340.

[37] R. Sebastiani and P. Trentin, "OptiMathSAT: A Tool for Optimization Modulo Theories," in *International conference on computer aided verification*, 2015, pp. 447–454.

[38] Docker Inc, "Docker." 2017.

[39] G. Collins, "Pymodbus 1.2.0." 2017.

[40] A. Cole, B. Jury, and K. Takashina, "A Leidenfrost Thermostat," *Journal of Heat Transfer*, vol. 137, no. 3, p. 034502, Mar. 2015.

[41] JS Foundation, "Node-RED." 2017.

[42] InfluxDataInc, "InfluxDB." 2017.

[43] A. Banks and R. Gupta, "MQTT Version 3.1. 1," *OASIS standard*, 2014.

[44] S. Gao, S. Kong, and E. Clarke, "Satisfiability Modulo ODEs," in *Formal methods in computer-aided design*, 2013, pp. 105–112.

[45] S. Kong, S. Gao, W. Chen, and E. Clarke, "dReach: δ-Reachability Analysis for Hybrid Systems," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2015, pp. 200–205.

[46] S. Gao, L. Xie, A. Solar-Lezama, D. Serpanos, and H. Shrobe, “Automated vulnerability analysis of AC state estimation under constrained false data injection in electric power systems,” in *2015 54th ieee conference on decision and control (cdc)*, 2015, vols. 2016-Febru, pp. 2613–2620.

[47] K. Bae, P. C. Ölveczky, S. Kong, S. Gao, and E. M. Clarke, “SMT-Based Analysis of Virtually Synchronous Distributed Hybrid Systems,” in *Proceedings of the 19th international conference on hybrid systems: Computation and control - hscc ’16*, 2016, pp. 145–154.

[48] S. Gao and D. Zufferey, “Interpolants in Nonlinear Theories Over the Reals,” in Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol. 9636, 2016, pp. 625–641.

AUTHOR

Alessio Coletta possesses a Master Degree in Computer Science at the ScuolaNormaleSuperiore di Pisa and a Master Degree in Information Security at the Royal Holloway University of London. He has worked at the Joint Research Centre of the European Commission as a scientific officer in the Security of Networked Critical Infrastructure unit. He has also worked in the Global Cyber Security Center foundation of Poste Italiane and in the Incident Prevention and Management structure of Poste Italiane. He is a PhD candidate at the University of Trento performing R&D activities on security of cyber physical system within the Security and Trust unit of the Fondazione Bruno Kessler in Trento (Italy).

INTENTIONAL BLANK

A TRADEOFF-BASED SECURITY MODEL AGAINST CLICK SPAM ORIGINATED BY SINGLE IP ADDRESSES

N.Zingirian, M.Benini

Department of Information Engineering - University of Padova - ITALY

ABSTRACT

This paper shows a vulnerability of the pay-per-click accounting of Google Ads to the attacks of a malicious single agent and proposes a statistical tradeoff-based approach to reduce this vulnerability. The contribution of this paper is a model to calculate the overhead cost per click necessary to protect the subscribers from click spam and a simple algorithm to implement this protection.

KEYWORDS

Pay-per-click Advertising, Google Ads, Web Advertising

1. INTRODUCTION

The pay-per-click accounting method adopted by Google Ads service for online advertising services [1][2] enables the Advertising Provider (AdP, e.g., Google) to automatically charge the Advertising Subscribers (AdS) for each single advertised page access, i.e., the "click", activated by each web user. Differently from previous online pay-per-click methods, this method does not need AdP and AdS to agree on respective web access logs and "referrer headers" [3], as a consequence the management cost of subscriber's signup and charging processes are kept extremely low. This method allows a single AdP to manage millions pay-per-click contracts, thus making the pay-per-click advertising a mass service, as it appears today.

Considering that the click count is the key number upon which such a mass business calculates huge turnovers, the paper presents a contribution to the very relevant topic of assessing the accuracy of such a number. In particular, the paper evaluates the robustness against malicious web agents, who might be motivated to spam the click counts, for instance, to attack a target AdS to exhaust its daily budget, eventually making it disappear from the advertising network during the first minutes of each accounted day [4].

While big effort has been spent to setup heuristics to detect distributed click spam [5][6][7] [8], this paper shows (Section 2) how a malicious single user agent (web client program), bound to one IP address only, can make the click count increase for a given AdS, even if the charged clicks do not correspond to any real advertisement.

The paper contributions are

- a statistical trade-off model that shows how to make the accounting scheme much safer for the AdS versus the AdP's risk of losing a small percentage of revenues (Section 3),
- a simple algorithm to take advantage operatively of the trade-off model (Section 4), and

David C. Wyld et al. (Eds) : CNSA, DMDB, AIAP - 2018
pp. 79– 86, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80106

- a simulation that validates the model (Section 5).

A discussion of the results and possible extensions concludes the paper (Section 6).

2. ATTACK

The attack presented in this paper is directed to the “fairness principle” of the AdP, that is to charge one click only to the AdS, even when an user agent accesses the same advertised web page more than once. This principle considers that only the first click corresponds to a real advertisement, whilst the other subsequent clicks, referred to as “repeated clicks” in this paper, do not provide any benefit in terms of advertised contents.

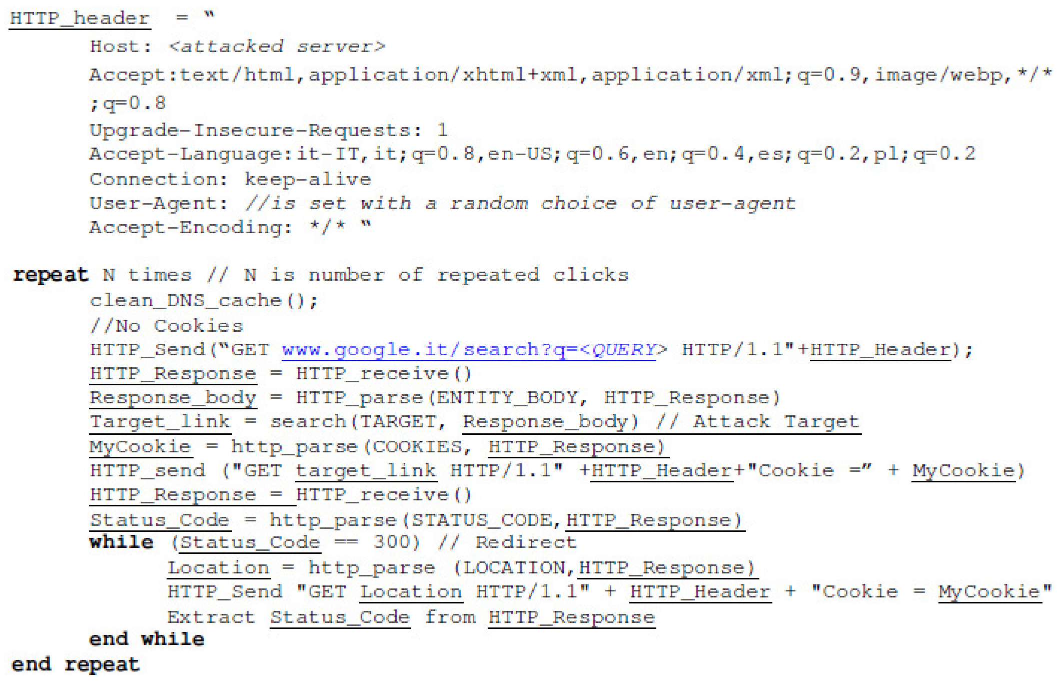
The vulnerability to this attack depends on the fact that no deterministic and secure rules to implement this principle are applicable on the AdP server side, to determine whether two clicks are originated by the same user agent or not.

According to our experiments, the approach followed by Google algorithm adopts the heuristic to consider *n* clicks as originated by the same user agents according two conditions

(a) if the user agent “cookie” header values is the same in the *n* HTTP requests corresponding to the *n* clicks OR

(b) if the *n* clicks are originated by the same IP address and also *n* DNS queries are originated from the same IP address to resolve the AP Server name, just before each of the *n* HTTP request corresponding to the clicks,

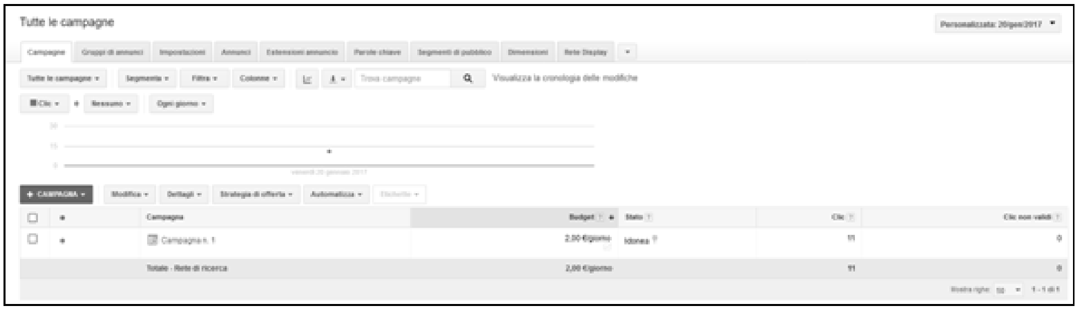
otherwise the *n* clicks are both accounted, as if they were originated by two different user agents.



```
HTTP_header  = "
          Host: <attacked server>
          Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*
          ;q=0.8
          Upgrade-Insecure-Requests: 1
          Accept-Language:it-IT,it;q=0.8,en-US;q=0.6,en;q=0.4,es;q=0.2,pl;q=0.2
          Connection: keep-alive
          User-Agent: //is set with a random choice of user-agent
          Accept-Encoding: */* "

repeat N times // N is number of repeated clicks
          clean_DNS_cache();
          //No Cookies
          HTTP_Send(“GET www.google.it/search?q=<QUERY> HTTP/1.1"+HTTP_Header);
          HTTP_Response = HTTP_receive()
          Response_body = HTTP_parse(ENTITY_BODY, HTTP_Response)
          Target_link = search(TARGET, Response_body) // Attack Target
          MyCookie = http_parse(COOKIES, HTTP_Response)
          HTTP_send ("GET target_link HTTP/1.1" +HTTP_Header+"Cookie =” + MyCookie)
          HTTP_Response = HTTP_receive()
          Status_Code = http_parse(STATUS_CODE,HTTP_Response)
          while (Status_Code == 300) // Redirect
                    Location = http_parse (LOCATION,HTTP_Response)
                    HTTP_Send "GET Location HTTP/1.1" + HTTP_Header + "Cookie = MyCookie"
                    Extract Status_Code from HTTP_Response
          end while
end repeat
```

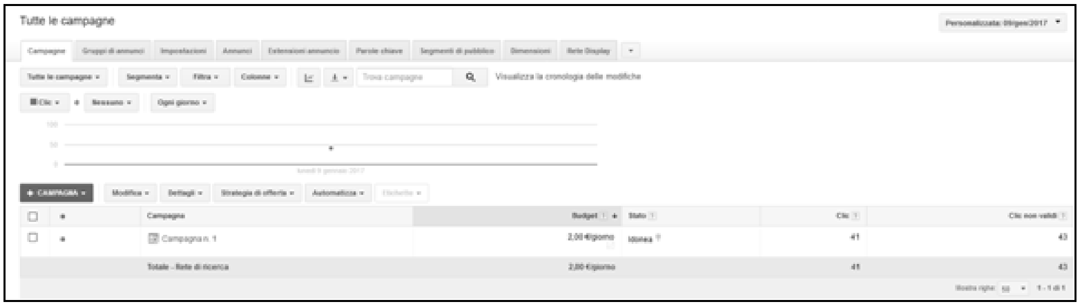
Figure 1 - Pseudo code of the attack



Campagna	Budget	Stato	Clic	Clic non validi
Campagna n. 1	2,00 €/giorno	Idonea	11	0
Totale - Rete di ricerca	2,00 €/giorno		11	0

Figure 1 - Click count report before the attacks, 11 clicks per day.

Consequently, the attack works as follows. A malicious HTTP client, whose pseudocode is available in Figure 1, performs many HTTP requests to the same link advertised by the AdP server, simply resetting both the cookie header value and the client's system DNS cache before sending each HTTP request. The client sends many requests from the same source IP address. In case of attack, our AdP has detected only about 42% of false clicks, while the others are normally accounted. As shown in Figure 1 the clicks accounted when attacks take place are much more than the regular average clicks per day shown in Figure 2.



Campagna	Budget	Stato	Clic	Clic non validi
Campagna n. 1	2,00 €/giorno	Idonea	41	43
Totale - Rete di ricerca	2,00 €/giorno		41	43

Figure 2 - Click count report after the attacks, only 43 false click detected over 73

Figure 2 and 3 shows the clicks accounted by Google Ads, before and after our attack respectively.

The reader could wonder why Google heuristic does not consider the IP address as a safe information to determine if the same user agent originates two clicks. The answer is that two or even more hosts, each running a different user agent, might send distinct HTTP requests from the same IP address to the same AdP server, as most IP networks over the Internet are IPV4 network adopting the Network Address Translation. The AdP server might see, in that case, different clicks of really different users coming from the same IP address.

We observe that the experiment shows that the heuristic adopted by Google Ads is clearly safe for the AdP but it is unfortunately unsecure for the AdS.

Our investigation explores how to use the probability, for any network, that two independent clicks, coming from the same IP address, originate from two different user agents, to decide whether to count or not to count the clicks.

3. TRADE-OFF MODEL

The model analyzes the clicks coming from a NAT address space using the following variables

- Time interval *T*, called memory interval, within which two clicks are counted only once if they are activated by the same user and have the same target. If the time distance between two following clicks is more than time *T*, then they are counted twice even when directed by the same user to the same advertised resource.

- Integer number *A*, corresponding to the cardinality of the NAT address pool.

- Integer number *C* corresponding to the number of clicks coming from any address in the NAT pool and directed to the same AS resource

The average number *N(A,C)* of "repeated clicks", is the numbers of clicks directed to the same resource and coming from the same IP address but from different users over a NAT pool consisting of *A* addresses, provided that *C* clicks in total are coming from that NAT pool.

N(A,C) is calculated considering that each user performs only one click, so that two different users using the same IP address originate repeated clicks, excluding any click spam event.

We define as loss factor *L*

$$L(A,C) \; = N(A,C) \times A \,/\, C$$

the average percentage of real clicks that are potentially lost (i.e. not accounted) if all multiple clicks (from the second one on), coming from the same address are systematically ignored. The average number of clicks N exceeding the single click is multiplied by the number A of addresses available, to obtain the average click over all the NAT network, and is divided by C to calculate the percentage over all clicks. This percentage corresponds to the loss rate of revenue that is paid by the AdP to protect the AdS.

If *L* is below a fixed threshold, e.g., 1% the protection heuristic decides to ignore all repeated click.

To calculate N(A,C) we consider the clicks as uniformly random independent events falling in a 1-d continuous space that is splitted in A equal segments, each representing the subspace of probability that a click is originated by the address represented by that segment. We assume that there is no correlation between IP address and user interest for a specific content, as there is no reason to suppose any correlation.

This corresponds a Poisson Distribution where λ = C/A.

Considering that the average number of repeated events falling in the same interval is

$$\sum_{c=2}^{C} \quad (c-1)\cdot P(clicks \;=\; c) \;=\sum_{c=2}^{C} \quad (c-1)\cdot \frac{\lambda^{c}e^{-\lambda}}{c!}$$

Then, as shown in the Appendix,

$$N(A,C) = \sum_{c=2}^{C} (c-1) \cdot \frac{(C/A)^c e^{-C/A}}{c!} = \frac{C}{A} + e^{-C/A} - 1$$

it is worth noticing that, as shown in the Appendix, if A >> C then

$$N(A,C) < \frac{1}{2} C^2/A^2$$

As a consequence

$$L(A,C) < \frac{1}{2} C \quad /A$$

This approximation, being a second-order Taylor polynomial approximation, is very precise for typical C/A values for large networks, e.g. If C/A = 10^{-3} then $\left|L(A,C) - \frac{1}{2} C/A\right| < 10^{-6}$

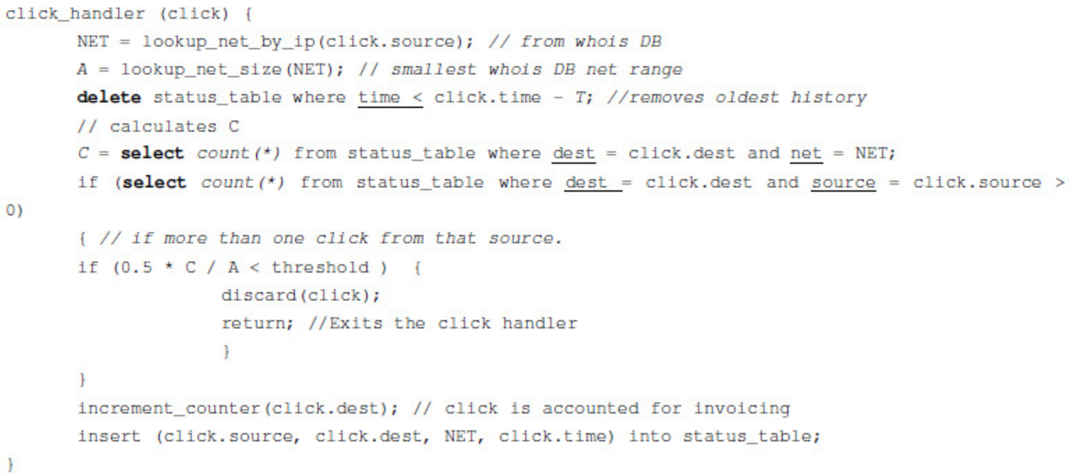
4. ALGORITHM

The proposed algorithm bases on the following entities:

1. A table, called the *status table* having the following fields:
    - dest : the destination URL of the Click
    - source: the source IP address (possibly NATted)
    - net: the id of the smallest network range registered in whois DB
    - time: timestamp (epoch)

2. An object, called click, having the properties of each click received, i.e.,
    - click.dest : the destination URL of the Click
    - click.source: the source IP address (possibly NATted)
    - click.time: the click message receiving timestamp (epoch)

Figure 4 shows the pseudo code of a click handler. The actions done after each click is to count the click as valid incrementing the click counter through the function `increment_counter()` or to reject the click through the function `discard` .The algorithm discards the clicks as long as the average statistical number of repeated clicks is below a given threshold.



```
click_handler (click) {
        NET = lookup_net_by_ip(click.source); // from whois DB
        A = lookup_net_size(NET); // smallest whois DB net range
        delete status_table where time < click.time - T; //removes oldest history
        // calculates C
        C = select count(*) from status_table where dest = click.dest and net = NET;
        if (select count(*) from status_table where dest = click.dest and source = click.source >
0)
        { // if more than one click from that source.
        if (0.5 * C / A < threshold )  {
                        discard(click);
                        return; //Exits the click handler
                        }
        }
        increment_counter(click.dest); // click is accounted for invoicing
        insert (click.source, click.dest, NET, click.time) into status_table;
}
```

Figure 4 Pseudo code of trade-off based protection algorithm

5. SIMULATION

We considered in our simulation the Italian provider Vodafone IT having 28.870.000 subscribers and 5.538.048 IP addresses registered. Supposing an advertisement having a huge impact e.g., the percentage of 0.1% over the whole population clicks the same advertised link then C= 28.870 and C/A = 5.21 . 10^{-3}.
According to the model presented in Section 4, the click loss ratio is ½ C/A = 2.6 . 10^{-3} if repeated clicks for each IP address are never accounted i.e. it is less than 0.26 %.

A simulation has been carried out to confirm the model.

The simulation programs:

- distributes N users randomly over the all IP address
- selects randomly N/1000 users who clicks the advertised link
- Counts how many clicks have been originated by each IP address
- Yields the number of IP addresses that originated 2 or more clicks divided by the total number of addresses

The simulation has executed 100.000 times and yielded the average value of 2.5368 x 10^{-3}.

The difference between model and simulation is:

$$|2.6065 \times 10^{-3} - 2.5368 \times 10^{-3}| = 6{,}9652 \times 10^{-5}$$

As a result, the impact of the loss is very low, even in case in which the number of clicks *C* is very high, and the discrepancy between the model and the simulation is negligible.

6. CONCLUSION

The tradeoff protection model presented in this paper allows the AdP precisely assigning a part of investment as an insurance to protect the customers from this very simple attack. The threshold used in the algorithm exactly corresponds to the percentage of turnover loss that can be decided by the AdP. Accepting this loss, the AdP protects the customer from click spam coming from a single IP address.

This model is particularly suitable for the large mass of small AdS who receive a few clicks per day for which even a moderate repeated attack could completely vanishes their investments. In that case, the customer could get the option of paying an additional security fee corresponding to e.g., 0.5% to 2% of the click price to get the protection against this attack, deciding the algorithm threshold accordingly. Alternatively, the AdP can set the threshold to modulate internal security costs to maintain appropriate service levels.

The time interval *T* should be large enough to collect enough statistics on *C*, and small enough to keep the memory of the clicks not too large to count the accesses of the same users who click again the same resource after long time.

This latter aspect is the key of evolution of the presented algorithm, as the standard deviation of the repeated clicks will be also considered to calculate the loss more precisely, possibly postponing the calculation when the mean is considered enough significant.

7. APPENDIX

By replacing λ = C/A, N(A,C) can be rewritten as

$$N(\lambda) = \sum_{c=2}^{C} (c-1)\,\frac{\lambda^{c} e^{-\lambda}}{c!}$$

and can be splitted into two parts:

$$N(\lambda) = \sum_{c=2}^{C} c\,\frac{\lambda^{c} e^{-\lambda}}{c!} - \sum_{c=2}^{C} 1 \cdot \frac{\lambda^{c} e^{-\lambda}}{c!}$$

The first part yields:

$$\sum_{c=2}^{C} c\,\frac{\lambda^{c} e^{-\lambda}}{c!} = \sum_{c=0}^{C} c\,\frac{\lambda^{c} e^{-\lambda}}{c!} - \lambda e^{-\lambda} - 0 =$$
$$= \lambda - \lambda e^{-\lambda} - 1$$

because $\sum_{c=0}^{C} c\,\frac{\lambda^{c} e^{-\lambda}}{c!}$ corresponds exactly to the Poisson mean λ

The second part yields

$$\sum_{c=2}^{C} \frac{\lambda^{c} e^{-\lambda}}{c!} = \sum_{c=0}^{C} \frac{\lambda^{c} e^{-\lambda}}{c!} - (e^{-\lambda} + \lambda\, e^{-\lambda})$$
$$= 1 - (e^{-\lambda} + \lambda e^{-\lambda})$$

because $\sum_{c=0}^{C} \frac{\lambda^{c} e^{-\lambda}}{c!} = 1$ as it is exactly the sum of the whole distribution.

Composing the two parts, we obtain

$$N(\lambda) = \lambda - \lambda e^{-\lambda} - 1 - (e^{-\lambda} + \lambda\, e^{-\lambda}) =$$
$$= \lambda + e^{-\lambda} - 1$$

Expanding N(λ) the Taylor's series of f(x) up to the second order calculated in 0 we obtain:

$$N(\lambda) = N(0) + N'(0)(\lambda - 0) + \frac{N''(0)(\lambda - 0)^2}{2!} + R_3(\lambda)$$

$$= (1 - 1) + (1 - 1)\lambda + \frac{1}{2}\lambda^2 + R_3(\lambda) =$$

$$= \frac{1}{2}\lambda^2 + R_3(\lambda)$$

Lagrange theorem states that there exists $\xi \in (0,\lambda)$ such that $R_3(\lambda) = -1/6\, e^{-\xi}\lambda$. As a consequence $R_3(\lambda) < 0$.

$$N(\lambda) = \frac{1}{2}(\lambda)^2 + R_3(\lambda) < \frac{1}{2}(\lambda)^2$$

Replacing λ with C/A we obtain:

$$N(C/A) = \frac{1}{2}(C/A)^2 + R_3(C/A) < \frac{1}{2}(C/A)^2$$

REFERENCES

[1] Q. Zhang, T. Ristenpart, S. Savage, and G. M. Voelker. Got traffic?: an evaluation of click traffic providers. In Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality, WebQuality '11, 2011.

[2] Yuan, Shuai, Ahmad Zainal Abidin, Marc Sloan, and Jun Wang. "Internet advertising: An interplay among advertisers, online publishers, ad exchanges and web users." arXiv preprint arXiv:1206.1754 (2012).

[3] T. Berners-Lee, J. Gettys, R. Fielding, J. Mogul, L. Masinter, P. Leach and H. Frystyck, Hypertext Transfer Protocol-HTTP/1.1, RFC 2616, HTTP Working Group, June 1999.

[4] V. Dave, S. Guha, and Y. Zhang. Measuring and Fingerprinting Click-Spam in Ad Networks. In ACM SIGCOMM Conference on Data Communication, 2012

[5] L. Zhang and Y. Guan. Detecting Click Fraud in Pay-Per-Click Streams of Online Advertising Networks. In Proceedings of the IEEE Conference on Distributed Computing Systems, 2008

[6] Exposing Click-Fraud Using a Burst Detection Algorithm, D. Antoniou, M. Paschou, E. Sakkopoulos, E. Sourla, G. Tzimas, A. Tsakalidis, E. Viennas

[7] M. Taneja, K. Garg, A. Purwar, S. Sharma, Prediction of click frauds in mobile advertising Contemporary Computing (IC3), 2015 Eighth International Conference on, IEEE (2015)

[8] H. Haddadi. Fighting Online Click-Fraud Using Bluff Ads. SIGCOMM CCR, 40(2):22–25, Apr. 2010.

MEDICAL IMAGES ANALYSIS IN CANCER DIAGNOSTIC

Jelena Vasiljević[1], Ivica Milosavljević[2], Vladimir Krstić[1], Nataša Zivić[3], Lazar Berbakov[1], Luka Lopušina[5], Dhinaharan Nagamalai[4] and Milutin Cerović[5]

[1]Institute Mihajlo Pupin, University of Belgrade, Belgrade, Serbia
[2]Center for Pathology and Forensics of the Military Medical Academy in Belgrade, Belgrade, Serbia
[3]University of Siegen, Germany
[4]Wireill, Australia
[5]The School of Computing, University Union

ABSTRACT

This paper shows results of computer analysis of images in the purpose of finding differences between medical images in order of their classifications in terms of separation malign tissue from normal and benign tissue. The diagnostics of malign tissue is of the crucial importance in medicine. Therefore, ascertainment of the correlation between multifractals parameters and "chaotic" cells could be of the great appliance. This paper shows the application of multifractal analysis for additional help in cancer diagnosis, as well as diminishing. of the subjective factor and error probability.

KEYWORDS

Multifractal, Fractal, Medical Images, Cancer

1. APPLICATION OF MULTIFRACTAL ANALYSIS IN IMAGE PROCESSING

Multifractal analysis of images is based on the definition of measurements with images that are gray levels. Then the multifractal spectrum is calculated. In contrast to many classic approaches, there is no filtering. The spectrum uses local as well as global information for segmentation, noise reduction or edge detection at image points.

Image analysis is a fundamental component of a computer visual problem, with applications in robotics, medical or satellite images ... Segmentation is an important step that provides a description of a basic individual process. Filtering then gives signal gradients where extremes roughly correspond to contours. Then, multi-resolution techniques can be used to "purify" the

David C. Wyld et al. (Eds) : CNSA, DMDB, AIAP - 2018
pp. 87– 97, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80107

results obtained. The main drawback of this approach is loss in precision due to preliminary filtering.

An alternative approach is to observe that a picture is a measure known as a fixed resolution. The irregularities of this measure can then be studied using a multifractal analysis. The general principle is the following: first, the different dimensions and capacitance are defined from the image which is the gray level. Then, the corresponding multifractal spectrum is calculated, providing both local (over) and global (through) information. There are no hypotheses about the regularity of the signal. Multifractal analysis (MF) can be successfully used in image processing. The idea of applying (inverse) MF analysis in extracting characteristic details in the picture is presented in [1].

The importance and the advantage of fractal and multifactal analysis (MFA), in relation to the "classic" signal analysis, lies in the way in which irregularity is considered. The MFA tries to extract information directly from singularity, while in the "classic" mode, most often, NF filtered versions are viewed, possibly with different filtering depths, to detect irregularities and suppress noise. In particular, based on a certain value i, the points of inhomogeneity in the original signal can be separated [1, 2, 3, 4]. By dividing pixel images that satisfy the selected parameter value, or spectrum, by inverse multifractal analysis (IMFA), it is possible to extract from the image of a region that can not otherwise be noticed by any of the known methods. An additional advantage is that such a segmentation does not cause any degradation of the initial image: all the mutual relations of the pixels remain unchanged, so that the details of the image are kept completely. This feature is particularly important in medical diagnostics, so the potential of IMFA in this area is high.

It is shown that a large number of frequently variables of a different nature (electrical signals, modern telecommunication traffic, meteorological and biomedical signals) can be described in a similar way. It is necessary to examine the fractal characteristics for the expression of significant variability. The use of classical statistical methods in such a case (mean value) could cause error estimates. The pronounced singularities indicate the multifunctionality of the process.

2. Fractal Morphometry Applied To Tumors

Despite the huge increase in our understanding of the molecular carcinoma mechanism, most diagnoses are still determined by visual examination of radiographic images, microscopic and biopsy patterns, direct examination of the tissue, and so on. Usually, these techniques are applied in a quality manner by clinicians who are trained to classify images showing abnormalities such as structural irregularities or high indications for mitosis. A more qualitative and reproducible method, which can serve as an ancillary tool for diagnostic training, is an analysis of images using computer tools. This lies in the potential of fractal analysis as a morphometric measure of irregular structures that are typical of tumor growth.

Pathologists are skilled in examining the boundary surface of the epithelial-connective tissue, which separates the tumor and surrounding healthy tissue. The nature of the tumor edge, whether infiltrative or invasive or poorly expansive, provides information useful not only for prognosis, but also for diagnosis (either benign or malignant tumors). In the study of Landini and Ripini [5], the border area of the epithelial-connective tissue of oral mucosa was examined. Lesions are classified by routine diagnosis into four categories: a) normal; b) medium dysplasia; c) moderate to severe dysplasia; d) carcinomas. Fractal lesion analysis, which subsequently followed, revealed the following fractal dimensions for the above four categories: 1.07 ± 0.05, 1.08 ± 0.09, 1.16 ± 0.08, and 1.41 ± 0.08, respectively. Although the differences were not large enough to be accepted as an independent tool in diagnostics, they are regardless of consistent measurements of the degree of distortion of the boundary surface. Landini and Ripini then proceeded to describe using a more sophisticated multifunctional analysis method that gives a spectrum of fractal values instead of one value for each image. This method has provided more reliable discrimination of the pathological conditions of the tissue. Lefebre and Benali [6] and Polman et al [7] have shown that fractal methods can also be useful for analyzing digitized mammograms, increasing the hope that the number of incorrect positive mammograms will be reduced in this way. Considering that the increase in irregularities, with associated fractal fracture enlargement, is a common indicator of tumor growth this undoubtedly represents a universal result. Out of everything contained in this chapter Fractals in biomedical systems, the hypotheses set out in this paper are derived.

2.1. Histopathological characteristics of normal mucous membrane, adenoma and adenocarcinoma of the colon

The normal mucous membrane of the mucous membrane of the colon is from the lumen of the intestine to the surface of the laminae epithelialis tunicae mucosae consisting of a layer of cylindrical goblet cells, absortive, endocrine and undifferentiated cells lying on the basal membrane and invaginating the so-called crypts They reach the next layer, laminae muscularis mucosae built of smooth muscle cells, and underneath there is a lamina propria of tuniciae mucosae with blood vessels, loose connective tissue, and individual mesenchymal cells.

The basic histological characteristics that point to the normality, regularity are: uniformity of the cells of the liminal epithelialis tunicae mucosae, polarization of the sails to the basal memebran, small sails, preserved secretion of the mucin by the goblet cells. Figure 1 shows a photo of a sample of the normal colon tissue, he (hematoxilieosin-coloring method), a transverse cross sectional view, and in Fig. 2, photographs of a sample of the normal tissue of the column, he is a longitudinal sectional view.

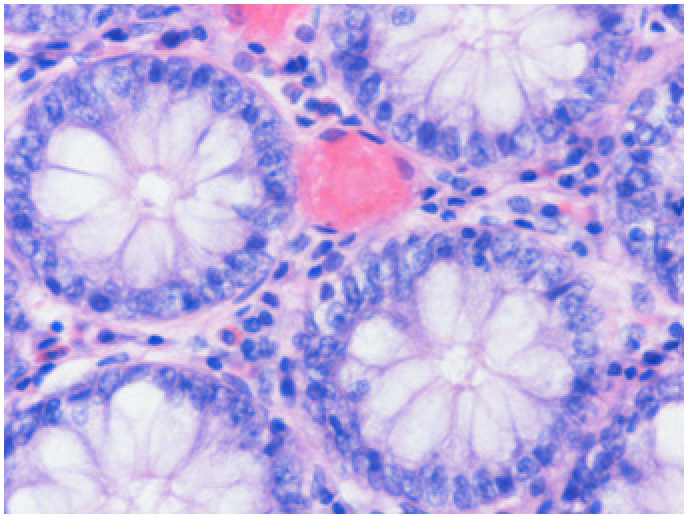


Figure 1: Photo of a sample of normal colon tissue, (he), a transversal section view

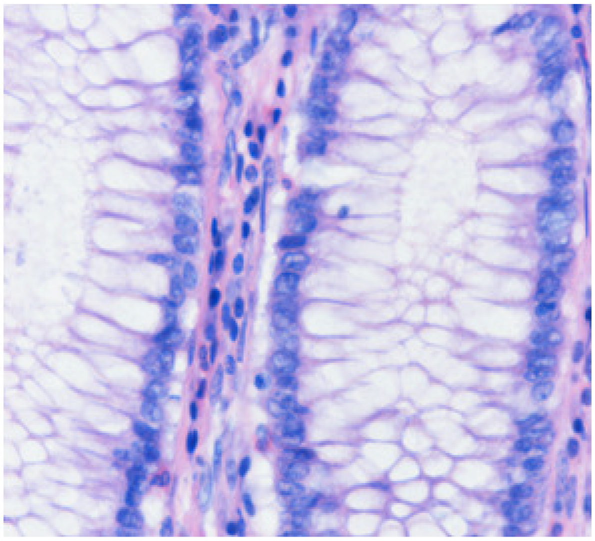


Figure 2: Photo of a sample of normal colon tissue, (he), a longitudinal section view

The most common tumors in the colon are the origin of the epithelium and they can be benign-nature-adenomas and malignant-nature-adenocarcinomas. Adenoma is a benign tumor of the epithelial origin in which proliferation of the epithelium occurs with the formation of glandular structures, which are coated with dysplastic cells. The disposable epithelium loses the uniformity of its cells, the cells are pleomorphic, and the usual architectural appearance is lost, tubular and / or vilosic structures are formed which are coated with cylindrical epithelial cells, with elongated and hyperchromatic nuclei and which may and may not have preserved mucigen activity, while the mitotic activity of the cells is increased. In Figure 3, a photo of the sample of the adenoma column is shown, he is a transversal sectional view, and in Figure 4 a photo of the sample of the adenoma column is shown, and he shows the longitudinal cross section.

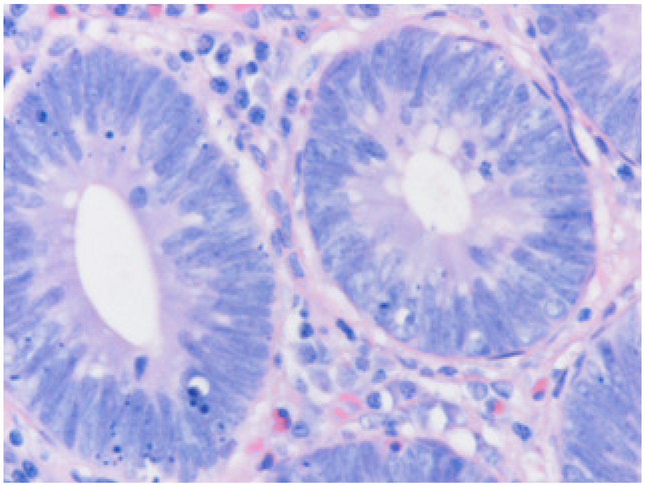


Figure 3: Photo of the column adenoma sample, (he), a transverse cross sectional view

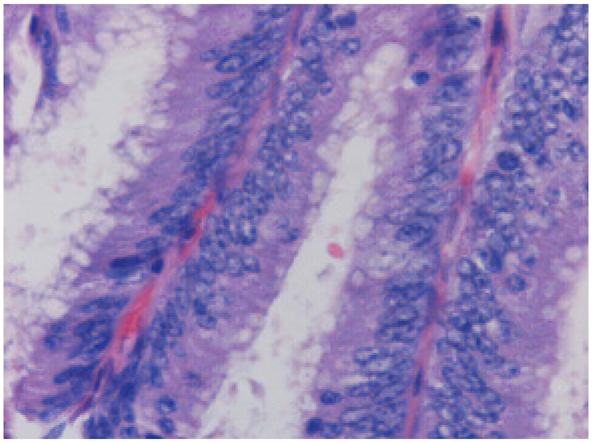


Figure 4: Photo of the column adenoma sample, (he), a longitudinal section view

Adenocarcinoma of the colon is one of the most common tumors in the human population and is one of the major challenges of human medicine, precisely because it can arise from the adenomas and what produces the symptoms relatively early, which allows diagnosis and treatment. Carcinoma tissue is constructed of tubular formations, irregular shape and size, as well as from cribriform formations, which are coated with cubic and cylindrical atypical cells whose nuclei are pleomorphic, hyperchromatic, with and without prominent cats. The surrounding stroma of the tumor is desmoplastic, with multiplied binders and with inflammatory lymphocyte infiltrate, plasmocyte, and granulocyte. The properties of adenocarcinoma are: local invasive growth by penetrating the basal membrane, infiltrating the column wall and possibly spreading into the surrounding structures, as well as the ability to metastasize to regional lymph nodes, and metastasis to distant organs: in the liver, lungs, bones, and other organs [CECI89]. Figure 5 shows a photograph of a sample of colon carcinoma, a he-transversal sectional view, and Figure 6 shows a photo of a sample of colon carcinoma, he- an indication of the longitudinal cross-section.

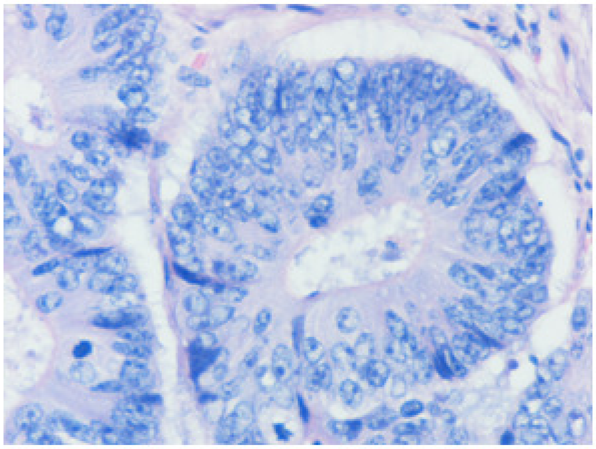


Figure 5: Photo of the column carcinoma sample, (he), a transverse cross-sectional view

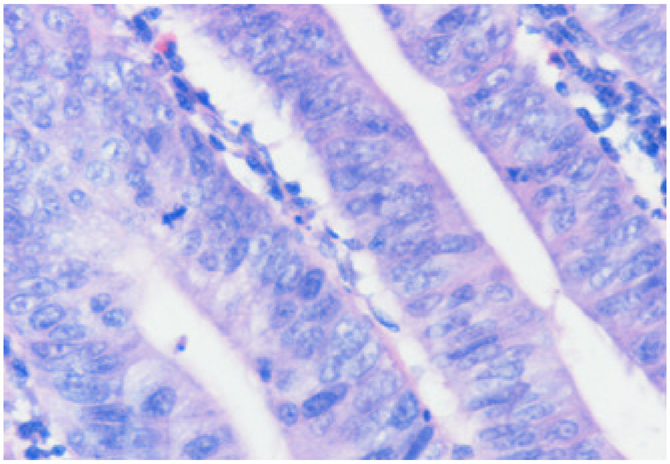


Figure 6: Photo of the column carcinoma sample, (he), a longitudinal section view

2.2. Methodological part of the research

Research goals

The intention behind this research is to determine the existence of differences in the parameters of the multifractal analysis of digital medical images between the following three tissue groups:

1. Normal mucosal tissue of the colon

2. Thick bowel mucosal tissue with diagnosis of malignant tumor origin of the epithelium – carcinomas

3. Thick bowel mucosal tissue diagnosed as benign tumors of the origin of the epithelium – adenomas

In order to obtain the required research results, it is necessary to perform the following steps:

1. Determine the parameters of the multifaceted analysis for the previously stated groups of tissue pictures

2. Determine the existence of statistically significant differences between the parameters corresponding to the previously mentioned tissue image groups.

2.3. Method

It is a non-experimental correlation study on the sample. In relation to the goal, the research is parametric.

2.4. Variables of research

Independent variable

1. Type of photographed tissue. Variables are operatively defined with the following categories:

a) tissue without pathological changes

b) tissue with cancer

c) adenoma tissue

Dependent variables

In the FracLac program

1. $D_{\max}$ - maximum

2. $\overline{Q}$ -Q which corresponds to the maximum

3. $\underline{\alpha}$ - which corresponds to the minimum

4. $f(\alpha)_{\min}$ - minimum

5. $\overline{\alpha}$ - which corresponds to the maximum

6. $f(\alpha)_{\max}$ - maximum

In FracLab

1. α_{sr} - mean value

2. $f(\alpha)_{sr}$ - mean value

3. $\overline{\alpha}$ - which corresponds to the maximum

4. α_{stdev} - standard deviation

5. $f(\alpha)_{stdev}$ - standard deviation

2.5. Instruments

The following programs were used for multifractal analysis of the obtained digital medical images and obtaining the parameters of multifactal analysis:

1. "FracLac" program for multifractal image analysis [8].
2. "FracLab" program for multifractal image analysis [9].
3. Program for statistical analysis of data SPSS (Statistical Package for Social Sciences), standard for statistical analysis of clinical research results.

2.6. FracLac

The "FracLac" program is created in the Java programming language and represents one of the plugins in the "ImageJ" digital image analysis program. "ImageJ" is freely available image analysis software written in the Java programming language authored by Wayne Rasband of the National Institute of Health of the United States of America from Bethesda, Maryland, [8].

The author of "FracLac" is Audrey Karperian from Charles Sturt University in Australia. She was contacted with this research. At her request, they were sent one sample sample from each of the three groups to test and improve the program for the specific problem of this research. With great help from her professional team, this program is adapted to handle a large number of images at once, relatively quickly. This is especially suited for use by medical personnel, where it would not be complicated training, nor would it take up a lot of their time. The images necessary for processing can be collected and then all at once processed.

2.7. The program "FracLab"

The FracLab program was developed by a team of experts at the INRIA and IRCCyN Institutes in France, led by Jacques Levy Vehell.

The principle of image segmentation using multifactal analysis is as follows: the points lying in the picture can be classified according to their Holder exponent. Let's look at the example of the points that lie on the contours. These points often correspond to the discontinuities of the gray level map or from its output. They therefore generally have a "low" Holder regularity. However, the exact value of the exponent will depend on the characteristics of the image. Additionally, the boundary edge feature is not purely local, and therefore a global criterion is required to decide whether a point is assigned, a point that belongs to the edge. Indeed, the points lying on the textures of the region also have a generally low regularity, and it is necessary to find a way to make them different from contours. Here, the other component of the multifractal analysis comes to the fore: since the edges are by definition the sets of points of the dimension one, we declare that the point lies on the contour if there is an exponent such that the associated value of the multifractal spectrum is one. In addition to the geometric characterization of the edges of the edges, it is also possible to make statistical points: the points of the edges can be defined by their probability of being affected when the pixel is randomly selected in the image at a given resolution. The relationship between the geometric and statistical representation of the edges of the edge provides multifractal formalism. Instead of edge detection, a much more complicated structure can be extracted using the same principle: starting again from Holder's exponents, points can be kept where the spectrum has a certain value. For example, by selecting a value of about 1.5, it is generally possible to extract very irregular contours. The value close to 2 corresponds either with smooth regions or textures.

The general procedure is as follows: it begins by calculating the Holder exponent at each point. This gives a picture of Holder's exponents.

The second step is calculating the multifunctional spectrum. In this paper the Hausdorf spectrum is calculated from the three spectra offered. The Hausdorf spectrum gives geometric information that relates to the dimension of the set of points in the image with the given exponent. This spectrum is a function in which apscis represents all Holder's exponents appearing in the image, and the ordinate is the dimension of the set of pixels with the given exponent.

The second spectrum is a large deviation spectrum that gives statistical information related to the probability of finding a point with a given exponent in the image (or more precisely, as this probability acts in the change of resolution).

The third spectrum is the so-called. The Leandre's spectrum, which represents only the concave approximation of the spectrum of large deviations, and its main contribution is to give much more robust calculations, although at the cost of losing information.

2.8. Hypotheses

General hypotheses

The parameters of multifractal analysis will significantly differ from all three groups: normal tissue, carcinomas and adenomas.

1. In particular, in the case of multifractal analysis of digital images of three groups observed, FracLac will distinguish the following obtained parameters for all three groups observed:

- $D_{\max}$ - maximum

- $\overline{Q}$ - Q that corresponds to the maximum

- $\underline{\alpha}$ - which corresponds to the minimum

- $f(\alpha)_{\min}$ - minimum

- $\overline{\alpha}$ - which corresponds to the maximum

- $f(\alpha)_{\max}$ - maximum

2. In the case of multifractal analysis of digital images of the three groups observed, FracLab will distinguish the following obtained parameters for all three groups observed:

- α_{sr} - Middle value

- $f(\alpha)_{sr}$ -Middle value

- $\overline{\alpha}$ - which corresponds to the maximum

- α_{stdev} - standard deviation

- $f(\alpha)_{stdev}$ - standard deviation

2.9. Sample

The sample consisted of 150 preparations obtained from biopsy from the gastrointestinal tract, more precisely from the colon, Adenocarcinoma tubulare coli. Of the 150 preparations, 50 were previously diagnosed as normal colon mucosal tissue, 50 were diagnosed as colon mucosal tissue with malignant epithelial tumors - carcinomas, and 50 diagnosed as colon mucosal tissue with benign epithelial tumors - adenomas.

In the Center for Pathology and Forensics of the Military Medical Academy in Belgrade, the preparations were prepared for analysis under a microscope with a magnification of 40x and photographed on a coolscope device (by the author with the help of Dr. Ivana Tufegčić) in digital form (Figures 1, 2 and 3). Coolscope is a kind of hybrid microscope in the body of the computer, the manufacturer is a Japanese firm, Nikon. Five different photographs were taken from each preparation, in order to obtain the most valid results of statistical analysis. In this way, a total of 750 digital images were obtained, 250 of each of the three groups.

3. CONCLUSIONS

In this study, multifractal analyzes were performed using two programs FracLac and FracLab, three groups of tissue pictures: normal tissue, carcinoma, and adenomas. Then, statistical processing of the obtained results was made using the SPSS statistical treatment program, which is usually used in clinical trials. In this way, the answers to the hypotheses posed in this paper are obtained. The general conclusion is that the basis of the general hypothesis proved to be correct, that the parameters of multifractal analysis differ significantly for all three groups of tissue tissues observed, and therefore the zero hypothesis about the non-separation of these groups was denied. This applies when applying both programs for multifractal analysis of FracLac and FracLab images.

Since the general hypothesis of this paper is confirmed, it can be concluded that this research has obtained positive results.. In the case of the FracLac program, the reliability of the classification of all three tissue groups analyzed based on the obtained multifractal parameters is 65.3%, which is more than the 60.7% obtained in the case of FracLab. In the case of the FracLac program, 80.0%, 73.0% and 85.0% were obtained, successfully classified cases for the following group relationships (respectively): normal tissue and carcinomas, carcinomas and adenomas, normal tissue and adenomas.

In the case of the FracLab program for the same relationships, tissue image groups, respectively, received the following reliability of 64.0% (which is considerably worse in relation to the same case with the FracLac program), 74.0% (which is a little better in relation to the same case for the FracLac program) and 80.0% (which is worse in relation to the same case with FracLac), we can conclude that the resulting classifications are very effective and this is generally more in the case of FracLac.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education and Science, Republic of Serbia, Science and Technological Development grant TR32037.

REFERENCES

[1] J. L. Vehel, (1996) "Introduction to the multifractal analysis of images", http://www-rocq.inria.fr/fractales

[2] B. Reljin, I. Pavlović,I. Reljin, I. Rakočević, (1999) "Multifraktalna analiza medicinskih slika", Zbornik VII konf. TELFOR-99, page.469-472

[3] P. Mannersalo, I. Norros, (1997) "Multifractal analysis: A potential tool for teletraffic characterization?", COST257TD(97)32, pp.1-17

[4] P. Mannersalo, I. Norros, (1997) "Multifractal analysis of real ATM traffic: A first book", COST257TD(97)19, pp. 1-8

[5] Landini G., Misson G. P., Murray P. I., (1993) "Fractal analysis of the normal human retinal fluorescein angiogram". Curr. Eye Res, 12: 23-27,.[Medline]

[6] Lefebvre F., Benali H., (1996) "A fractal approach to the segmentation of microcalcifications in digital mammograms". Med. Phys, 22: 381-390, 1995.[Medline]eds. Fractal Horizons, : 251-262, St. Martin's Press New York.

[7] Pohlman S., Powell K., Obuchowski N. A., Chilcote W. A., Grundfest-Broniatowski S., (1996) "Quantitative classification of breast tumors in digitized mammograms". Med. Phys, 23: 1337-1345,.[Medline]

[8] Wayne Rasband, ImageJ, http://rsb.info.nih.gov/ij/

[9] J. L. Vehel, FracLab, http://www.irccyn.ec-nantes.fr/hebergement/FracLab/

AUTHOR INDEX