Dhinaharan Nagamalai
Natarajan Meghanathan (Eds)

Computer Science & Information Technology

3rd International Conference on Computer Science and Information Technology (COMIT 2019), January 19-20, 2019, Chennai, India

AIRCC Publishing Corporation

Volume Editors

Dhinaharan Nagamalai,
Wireilla Net Solutions, Australia
E-mail: dhinthia@yahoo.com

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

ISSN: 2231 - 5403
ISBN: 978-1-921987-97-7
DOI : 10.5121/csit.2019.90101- 10.5121/csit.2019.90105

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

Preface

The 3rd International Conference on Computer Science and Information Technology (COMIT 2019), was held in Chennai, India during January 19-20, 2019. The 3rd International Conference on Artificial Intelligence, Soft Computing and Applications (AISCA 2019), was collocated with The 3rd International Conference on Computer Science and Information Technology (COMIT 2019). 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 COMIT-2019, AISCA-2019 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, COMIT-2019, AISCA-2019 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 COMIT-2019, AISCA-2019

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.

Dhinaharan Nagamalai
Natarajan Meghanathan

Organization

General Chair

Dhinaharan Nagamalai	Wireilla Net Solutions, Australia
Natarajan Meghanathan	Jackson State University, USA

Program Committee Members

Abdolreza Hatamlou	Islamic Azad University, Iran
Abdulhamit Subasi	Effat University, Saudi Arabia
Abhijit Das	Manipal University, India
Aftab Alam	King Khalid University, Kingdom of Saudi Arabia
Ahmad Khasawneh	Hashemite University, Jordan
Ahmad Qawasmeh	The Hashemite University, Jordan
Ajay B. Gadicha	Sant Gadge Baba Amravati University,India
Alessio Ishizaka	university of Portsmouth, United Kingdom
Ali AL-zuky	Mustansiriyah University, Iraq
Ali Asghar Rahmani Hosseinabadi	Islamic Azad University Amol, Iran
Amir Salarpour	Bu-Ali Sina University, Iran
Amit Choudhary	Maharaja Surajmal Institute, India
Amizah Malip	University of Malaya, Malaysia
Anand Nayyar	Duy Tan University, Vietnam
Anazida Zainal	Universiti Teknologi Malaysia, Malaysia
Andrey Krylov,	Lomonosov Moscow State University, Russia
Ankit Chaudhary	Truman State University, USA
Ankur Singh Bist	KIET Ghaziabad, India
Ashwath Rao B	Manipal University, India
Atanu Nag	Modern Institute of Engineering & Technology, India
Avadhani P.S	Andhra University, India
Ayad Salhieh	Australian College of Kuwait, Kuwait
Aysegul UCAR	Firat University, Turkey
Babak Daghighi	Azad University (West Tehran branch), Iran
C V Guru Rao	SR Engineering College, India
Chandrashekhar Bhat	MIT Manipal, India
CHERIF Adnen	University of Tunis Manar, Tunis-Tunisia
Chin-Chih Chang	Chung Hua University,Taiwan
Chirag N. Paunwala	SCET, India
CHOUAKRI Sid Ahmed	University of Sidi Bel Abbes, Algeria
Dac-Nhuong Le	Haiphong University, Vietnam
Dammavalam Srinivasa Rao	VNRVJIET, India
Dariusz Jacek Jakobczak	Koszalin University of Technology, Poland
Diego Reforgiato Recupero	University of Cagliari,Italy
Dimitris Kanellopoulos	University of Patras, Greece
Doreswamy	Mangalore University, India
Dzmitry Kliazovich	University of Trento, Italy
Emilio Jimenez Macias	University of La Rioja, Spain

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Networks & Communications Community (NCC)

Soft Computing Community (SCC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

3rd International Conference on Computer Science and Information Technology (COMIT 2019)

3rd International Conference on Artificial Intelligence, Soft Computing and Applications (AISCA 2019)

A SURVEY ON THE DIFFERENT IMPLEMENTED CAPTCHAS

Shadi Khawandi, Firas Abdallah and Anis Ismail

Faulty of Technology, Lebanese University, Lebanon

ABSTRACT

CAPTCHA is almost a standard security technology, and has found widespread application in commercial websites. There are two types: labeling and image based CAPTCHAs. To date, almost all CAPTCHA designs are labeling based. Labeling based CAPTCHAs refer to those that make judgment based on whether the question "what is it?" has been correctly answered. Essentially in Artificial Intelligence (AI), this means judgment depends on whether the new label provided by the user side matches the label already known to the server. Labeling based CAPTCHA designs have some common weaknesses that can be taken advantage of attackers. First, the label set, i.e., the number of classes, is small and fixed. Due to deformation and noise in CAPTCHAs, the classes have to be further reduced to avoid confusion. Second, clean segmentation in current design, in particular character labeling based CAPTCHAs, is feasible. The state of the art of CAPTCHA design suggests that the robustness of character labeling schemes should rely on the difficulty of finding where the character is (segmentation), rather than which character it is (recognition). However, the shapes of alphabet letters and numbers have very limited geometry characteristics that can be used by humans to tell them yet are also easy to be indistinct. Image recognition CAPTCHAs faces many potential problems which have not been fully studied. It is difficult for a small site to acquire a large dictionary of images which an attacker does not have access to and without a means of automatically acquiring new labeled images, an image based challenge does not usually meet the definition of a CAPTCHA. They are either unusable or prone to attacks. In this paper, we present the different types of CAPTCHAs trying to defeat advanced computer programs or bots, discussing the limitations and drawbacks of each.

KEYWORDS

CAPTCHAs, Labeling,Segmentation, Image recognition

1. INTRODUCTION

With the development of the computer applications in different fields, internet has made a tremendous progress and become a special need in human life. It has applications in a wide range of daily affairs including trade, education, daily purchases and dialogues take place with the use of Internet. One of the common actions in the Internet web sites, especially commercial and administrative ones, is to fill out registration forms for certain purposes. Unfortunately, there are some programs which automatically fill out these forms with incorrect information to abuse the site, or automated programs which are usually written to generate spam.

Thus, differentiating between a user and machine over the internet has significant importance in the fields of internet security, artificial intelligence, and machine learning. Currently, HIPs takes the role of preventing robots from signing up for free online services (such as email accounts), abusing online polls, providing biased feedback, and spamming innocent users.

Dhinaharan Nagamalai et al. (Eds) : COMIT, AISCA - 2019
pp. 01–11, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.90101

Completely Automated Public Turing test to tell Computers and Humans Apart is class of automated challenges used to differentiate between legitimate human users and computer programs or bots on the internet. Thus, it plays the same role of HIP.

In 1997 Andrei Broder, Chief Scientist of AltaVista, and his colleagues prevented automated machine from adding of URLs to their search engine. They developed a program that permitted human's entrance but not machine's entrance. In 2000, Bots were annoying genius chatter by advertising sites and elicit personal information. CMU researchers: Manual Blum, Luis A. von Ahn and John Langford coined the term "CAPTCHA" that was pointed to "capture", and used CAPTCHA in order to solve Yahoo's chat room problem. In 2001 Allison Coates, Henry S. Baird and Richard Fateman of UC Berkeley developed Pessimal Print: that is low–quality of printed text images used certain rate of distortion [1].

The notion of a machine imitating human intelligence was first addressed as early as 1950 by English mathematician and logician Alan Turing [2]. Acknowledged as the father of modern computing, Turing recognized that computers might eventually be able to imitate human thought in very convincing ways. Therefore, he suggested what is now known as the Turing test, where a human converses with a computer without seeing it. If the human is convinced by the computer's answers that it is human, then the machine passes the test and is deemed to have some level of human-like intelligence.

The idea of a reverse Turing test, where a computer attempts to differentiate between a human and a computer, arose during the late 1990s when computer programs began to imitate humans in order to misuse the resources of internet-based systems.

HIPs [3] are a slight modification of a reverse Turing test, where the challenge is administered by a machine and taken by a human. The burden is on the human participant to convince the machine that he is human. Furthermore, the challenge should not be solvable by any machine. Notice the paradox that this creates: the machine can automatically create, administer, and grade a test that it itself cannot pass. Tests developed to differentiate these programs from real humans took the form of what would come to be known as CAPTCHAs.

2. EXISTING SOLUTIONS

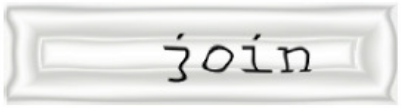
Many CAPTCHA implementations were designed by different companies (Microsoft, Yahoo, AltaVista) in order to offer a more secure online environment. An environment that distinguishes internet communications originating from humans from those originating from software robots. This section is going to present the different types of CAPTCHAs trying to defeat advanced computer programs or bots, discussing the limitations and drawbacks of each.

2.1 TEXT-BASED CAPTCHAS AND THEIR LIMITATIONS

In character labeling based CAPTCHA designs, the computer renders a sequence of letters after distorting them and adding noise. The user is asked to tell what characters they are in order, and will pass the test if the characters typed (new labels) match exactly those known to the server (known labels). Character labeling CAPTCHAs are the most widely used CAPTCHAs. The popularity of such schemes is due to the fact that they have many advantages [4], for example, being intuitive to users world-wide (the user task performed being just character recognition), having little localization issues (people in different countries all recognize Roman characters), and of good potential to provide strong security (e.g. the space a brute force attack has to search can be huge, if the scheme is properly designed).

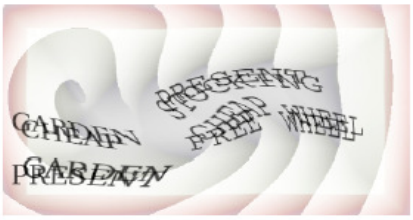
In 1997, AltaVista developed the first concrete implementation of a CAPTCHA. AltaVista had been receiving automated URL submissions to their search engine database by spam bots. A group of researchers from the Digital Equipment Systems Research Center were contracted to develop a solution to prevent such an attack [5]. To combat this, the team of developers created a verification system that makes suggestion of recognizing handwritten images. However, they soon realized that although an image containing text was a step in the right direction, it could easily be foiled by use of OCR software. Optical Character Recognition (OCR) software is designed to translate images of text into a machine editable form. The team researched the limitations of scanners with OCR capabilities, and exploited the weaknesses of the OCR systems when rendering their CAPTCHAs. In order to improve OCR results, the manual suggested using similar typefaces, plain backgrounds, and no skew or rotation. To create an image that was resilient to OCR, they did the exact opposite of the suggestions.

In the summer of 2000, Yahoo also began to experience a similar problem where their chat rooms were being spammed by chat bots. This gave birth to the CAPTCHA project. The researchers provided yahoo with three options (see Fig. 1): EZ-Gimpy renders a single, distorted English word on a noisy background, Gimpy-r renders a random string of distorted characters on a noisy background, and Gimpy renders 5 pairs of overlapping distorted words (of which you must type 3).



(a) EZ-Gimpy CAPTCHA

(b) Gimpy-r CAPTCHA



(c) Gimpy CAPTCHA

Figure 1 - Examples of EZ-Gimpy, Gimpy-r, and Gimpy CAPTCHAs

In June 2003, shape context matching was used to solve Gimpy with 33% accuracy and EZ-Gimpy with 93.2% accuracy [6]. In June 2004, distortion estimation techniques were used to solve EZ-Gimpy with 99% accuracy and Gimpy-r with 78% accuracy [7]. Due to the limited and fixed size of EZ-Gimpy's dictionary, every challenge image was easily compared against a template database. The distorted template image with the best correlation was returned as the result. However, Gimpy-r does not rely on a dictionary, and therefore requires local distortions to be removed via distortion estimation techniques.

In 2001, researchers at the Xerox Palo Alto Research Center and the University of California at Berkeley synthesized low quality images of machine printed text using a range of words, fonts, and image degradations .Following Baird's quantitative stochastic model of document image quality [8] and a list of problematic OCR examples, noise was introduced into the rendered strings by using two image-degradation parameters, blurring and thresholding (see Fig. 2). A couple of years later, a reading based CAPTCHA known as Baffle Text [9, 10] was developed (Fig. 3). Baffle Text exercised the Gestalt perception abilities of humans, humans are extremely good at recognizing and understanding pictures despite incomplete, sparse, or fragmented information, where as machines are not.

Figure 2 - PessimalPrint CAPTCHAs

Figure 3 - Baffletext CAPTCHAs

OCR systems separate recognition into two sub tasks, segmentation and classification. In 2004, researchers at Microsoft Research exploited the fact that segmentation is much more difficult than classification for OCR systems. So, they developed a CAPTCHA based on hard segmentation problems, as opposed to hard classification problems. Although character classification was still required, the main challenge was correctly segmenting the string. Another contribution was the observation that website owners with CAPTCHAs have the advantage in the battle against CAPTCHA attackers. This is because CAPTCHA generation is a synthesis task while attacking a CAPTCHA is an analysis task. Analysis is orders of magnitude more difficult than synthesis. In the synthesis task, the creator has the ability to use randomness and creativity, while in the analysis task, the attackers are tightly constrained by the decisions made by the creator.

Figure 4 - Microsoft's Segmentation-Based CAPTCHAs

A formal study of user friendliness for transcription tasks was conducted at Microsoft Research. They studied the effects of varying the distortion parameters and attempted to determine the optimal parameters where the CAPTCHAs prove hard for machines but easy for humans. As researchers found in the past, the most effective CAPTCHAs are segmentation based challenges, which continues to be a computationally difficult task (see Fig. 4). In 2004, researchers at Microsoft Research attacked several commercial CAPTCHA implementations and achieved high accuracy (80%-95%) [11]. Neural networks were used to perform character recognition. Their attacks had the most difficulty with the segmentation task, not the recognition task. Therefore, they suggested that researchers focus their efforts on building CAPTCHAs which rely on the segmentation task instead of the recognition task. It was later confirmed in July 2005 that computers are as good as, or better than humans at classifying single characters under common distortion and clutter techniques. However, other researchers have developed an attack that recognizes the “hard-to-segment” Microsoft CAPTCHA more than 60% of the time.

Figure 5 presents some character based CAPTCHAs that can be sampled from the web while signing up for free e-mail accounts with Mailblocks (www.mailblocks.com), MSN/Hotmail (www.hotmail.com), Yahoo (www.yahoo.com), Google (gmail.google.com), running a whois query at Register.com (www.register.com) or searching for tickets at Ticketmaster (www.ticketmaster.com).

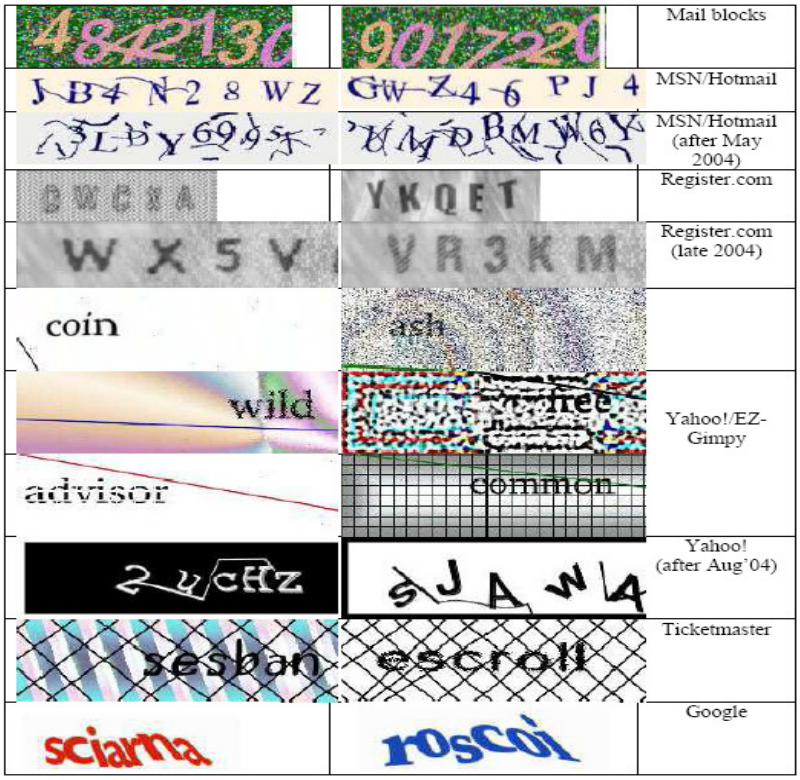


Figure 5 - Examples of Various Character Labeling CAPTCHA

Solutions to Yahoo (version 1) CAPTCHAs are common English words, but those for Ticket master and Google do not necessarily belong to the English dictionary. They appear to have been created using a phonetic generator. Examining the changes in MSN, Yahoo, and Register.com HIPs, it can be noted that these CAPTCHAs are becoming progressively more difficult. While MSN introduced more arcs as clutter, Yahoo gave up their language model and replaced simple textures and grids with more random intersecting lines and arcs. Register.com's update was minor as they introduced digits into their character set.

In [12] Chellapilla et al. have discussed the various issues when designing a character labeling based CAPTCHA. They can be summarized as follows: Character set, the character set to be used in the CAPTCHA. Affine transformations, which are Translation, rotation, and scaling of characters. Adversarial clutter represented as Random arcs, lines, or other simple geometric shapes that intersect with the characters and themselves. Image warp such as elastic deformations of the CAPTCHA Image at different scales i.e., those that stretch and bend the character itself (global warp) and those that simply jiggle the character pixels (local warp) and Background and foreground textures which are used to form a colored CAPTCHA image from a bi-level or grayscale CAPTCHA mask.In [13], each character fragment is labelled in order from top to bottom and left to right, and then the components are combined on the idea of jigsaw puzzle to generate candidate characters.

[14] provides a systematic analysis of text-based CAPTCHAs and innovatively improve their earlier attack on hollow CAPTCHAs to expand applicability to attack all the text CAPTCHAs. With this improved attack, they have successfully broken the CAPTCHA schemes adopted by 19 out of the top 20 web sites in Alexa including two versions of the famous Re CAPTCHA. With success rates ranging from 12 to 88.8% (note that the success rate for Yandex CAPTCHA is 0%), they demonstrate the effectiveness of their attack method. It is not only applicable to hollow CAPTCHAs, but also to non-hollow ones.

[15] present a novel segmentation and recognition method which uses simple image processing techniques including thresholding, thinning and pixel count methods along with an artificial neural network for text-based CAPTCHAs. We attack the popular CCT (Crowded Characters Together) based CAPTCHAs and compare our results with other schemes. As overall, our system achieves an overall precision of 51.3, 27.1 and 53.2% for Taobao, MSN and eBay datasets with 1000,500 and 1000 CAPTCHAs respectively.

2.2 IMAGE-BASED CAPTCHAS AND THEIR LIMITATIONS

While requiring a user to recognize distorted characters is the most common type of CAPTCHA, semantic image understanding tasks have also been proposed. Chew and Tygar from the University of California at Berkeley investigated a set of three image recognition tasks using a fixed English dictionary of 627 words and Google Images [16, 17]. The Naming images, where the use should determine the common term associated with a set of 6 images (see Fig. 6a). They used approximate matching to grade the responses. Second, Distinguishing images where the he should determine if two sets of images contain the same subject, and finally identifying anomalies, where he should identify the “odd one out” from a set of 6 images (see Fig. 6b)

The problems which affected human performance were evaluated and tested during an in-depth user study. Two formal metrics for evaluating CAPTCHAs were also proposed as well as attacks on the three image-based CAPTCHAs. The first metric evaluated CAPTCHA efficacy with respect to the number of rounds of a CAPTCHA and the second metric measured the expected time required for a human to pass the CAPTCHA.

In late 2003, researchers at Microsoft Research argued that the most familiar objects to humans are human faces. They developed a CAPTCHA designed to confuse face recognition algorithms while still being easy to use [18, 19, 20]. Images are automatically synthesized from facial models and the task is to locate and click on the 4 corners of the eyes and 2 corners of the mouth (6 points in total). However, the images looked eerie to many users (see Fig. (7)). For this reason, the system was never adopted.

A similar approach to face recognition based CAPTCHA was developed in 2006 [21]. Photographs of human faces were mined from a public database and distorted. The user is then prompted to match distorted photographs of several different humans. This CAPTCHA has the benefit of being language independent (ignoring textual instructions for completing the task).

6 a - The Image-Based Naming CAPTCHA

6 b - The Image-Based Anomaly CAPTCHA

Figure 6 - Examples of Imaged-Based Naming and Anomaly CAPTCHAs

Figure 7 - Example of an Artifacial CAPTCHA

In January 2005, some researchers thought that current CAPTCHAs were too demanding of legitimate human users. Instead, they proposed Implicit CAPTCHAs which require as little as a single click [22]. The challenges were so elementary that a failed challenge indicates an attempted bot attack. The authors suggest disguising necessary browsing links in images and claim that bots would not be able to find these hidden links (see Fig. 8). While the usability of the system is attractive, the system could easily be attacked on a case-by-case basis. For example, if the user is told to click on a specific, static place on an image, an attacker would only have to solve this once (challenges are static and therefore are reused). This type of CAPTCHA may work for low traffic or low value services, but it would never survive in a large scale application, as it is impossible to automate the generation of challenges.

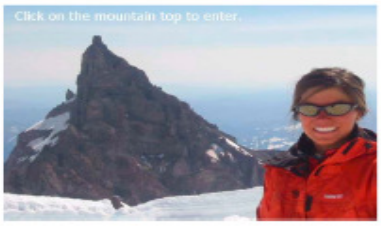


Figure 8 - CAPTCHA - User is Instructed to Click on Top of Mountain

One of the more interesting CAPTCHA ideas appeared in January 2011 as a result of an effort by social-networking giant Facebook. The company is currently experimenting with social authentication in an effort to verify account authenticity (see Figure 9).

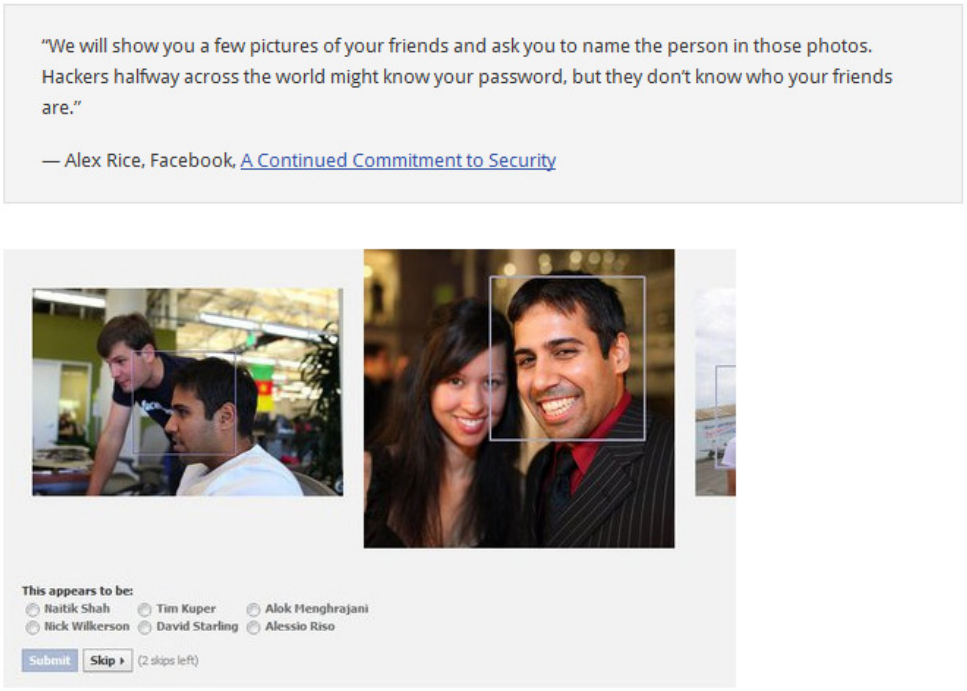


Figure 9 - Facebook's Friend Recognition Test

What makes Facebook's project slightly different than the normal CAPTCHA is that the authentication is supposed to filter out human hackers rather than machines.

There is potential for Facebook to roll this out across the Web. With 600 million users and millions of websites that integrate with it, Facebook has the ability to use this social recognition CAPTCHA in a big way, and it could prove to be easier than text recognition.

There is one problem. People does not actually know there friends. The reality is that friend requests are exchanged between even the barest of acquaintances, remembering names to go with all those faces could be challenging. As intuitive and intelligent as Facebook's idea might be, it is ultimately flawed because, as humans, we do not follow the rules.

Significant amounts of research have gone into the development of CAPTCHAs over the past 12 years. The first CAPTCHAs required users to transcribe strings of distorted text. Later, more advanced CAPTCHAs which relied on image understanding emerged. Text based CAPTCHAs were usable but easily defeated, while image based ones were affecting human performance. The challenge in designing an effective CAPTCHA is making a compromise, CAPTCHA must not only be human friendly but also robust enough to resist computer programs that attackers write to automatically pass CAPTCHA tests.

Bongo CAPTCHA is named after Mikhail M Bongard who published pattern recognition problems book. In Bongo [23] visual based pattern recognition is provided for the user to solve. The Figure 10 shows an example of Bongo CAPTCHA. It contains 2 block series namely the

right block and the left block series. The series of the right block differs from the left blocks, and the user should identify the characteristic which set them apart.

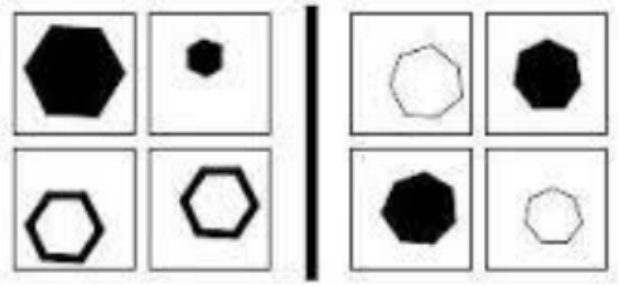


Figure 10- Example of Bongo catpcha

2.3 VIDEO BASED CAPTCHA

Video based CAPTCHA system [24] uses a technique in which the video contains few random words. The Figure 11 shows an example of Video CAPTCHA. When the video is played the user has to submit those displayed words. The users need not to wait until the video finishes for submitting the displayed words. The user passes the test only when the ground truth tags which are produced automatically matches with the user entered tags.

Figure 11- Example of Video based CAPTCHA

2.4 PUZZLE BASED CAPTCHA

Puzzle based CAPTCHA can either be a picture based puzzle or a mathematical puzzle. The Figure 12 shows an example of Puzzle based CAPTCHA. In a picture based puzzle, the picture is divided into segments and is shuffled. Each segment will have a segment number followed by the next segment. The user has to combine these segments properly to form a correct complete picture [25]. The mathematical puzzle is 100% effective and can be integrated into login, registration forms in the website for secured access. The user has to solve the math puzzle provided in order to gain the access to secured services.

Figure 12 - Example of Puzzle based Captcha

3. CONCLUSIONS

CAPTCHA plays important role in World Wide Web security where it prevents Bot programsand Hackers from abusing online services. In this paper, we have provided a set of techniques that would allow for the system to be secure and less vulnerable to bot attacks. It is a well synthesized CAPTCHA, where the attacker should pass three obstacles in order to bypass it.

REFERENCES

[1] H. S. Baird, A. L. Coates, and R. J. Fateman, "Pessimalprint: A reverse turing test", Int. Journal of Document Analysis and Recognition, 5(2-3):158-163, Seattle, WA, April 2003.

[2] "The Alan Turing Internet Scrapbook, The Turing Test 1950", [Online]. Available:http://www.turing.org.uk/turing/scrapbook/test.html

[3] K. A. Kluever, "Securely Extending Tag Sets to Improve Usability in a Video-Based HumanInteractive Proof", Department of Computer Science Rochester Institute of Technology, Rochester. [Online]. Available: http://www.kloover.com/thesis/proposal.pdf

[4] K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski, "Building segmentation based human-friendly human interaction proofs (hips)", H.S. Baird and D.P. Lopresti (Eds.): HIP 2005, LNCS 3517, pp. 1-26, 2005.

[5] M. D. Lillibridge, M. Abadi, K. Bharat, and A. Z. Broder, "Method for Selectively Restricting Access to Computer Systems," U.S. Patent No. 6,195,698, February 27, 2001

[6] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: breaking a visual Captcha",presented at conf. Computer Vision and Pattern Recognition, vol. 1, pp 134-141, Madison, WI, USA, June 2003.

[7] G. Moy, N. Jones, C. Harkless, and R. Potter, "Distortion estimation techniques in solving visual Captchas", presented at Conf. on Computer Vision and Pattern Recognition, vol. 02,pp 23-28, Los Alamitos, CA, USA, June 2004.

[8] H. S. Baird, "Document image defect models and their uses", in Proc of the 2nd Int. Conf. on Document Analysis and Recognition, pp 62-67, Tsukuba Science City, Japan, October 1993.

[9] M. Chew and H. S. Baird, "Baffletext: A human interactive proof", in Proc. of the SPIE/IS&T Document Recognition & Retrieval Conf. X, Santa Clara, CA, pp 305-316, January 2003.

[10] H. S. Baird and M. Luk, "Protecting websites with reading-based aptcha", in Proc. of the 2nd Int. Web Document Analysis Workshop, pp 53-56, Edinburgh, Scotland, August 2003.

[11] K. Chellapilla and P. Y. Simard, "Using machine learning to break visual human interaction proofs (HIPs)", In L. K. Saul, Y. Weiss, and L. Bottou, editors, Advances in Neural Information Processing Systems 17, pp 265-272, Cambridge, MA, December 2004.

[12] L. v. Ahn, M. Blum, and J. Langford. Telling humans and computers apart automatically.Communications of the ACM, 47(2):56-60, February 2004.

[13] H. Gao, J. Yan, F. Cao et al., "A Simple Generic Attack on TextCaptchas," in Proceedings of the Network and Distributed SystemSecurity Symposium, pp. 1–14, San Diego, Calif, USA, 2016.

[14] H. Gao, X. Wang, F. Cao et al., "Robustness of text-based completely automated public turing test to tell computers and humans apart," IET Information Security, vol. 10, no. 1, pp. 45–52, 2016.

[15] Rafaqat Hussain, Hui Gao, Riaz Ahmed Shaikh, Segmentation of connected characters in text-based CAPTCHAs for intelligent character recognition, Multimedia Tools and Applications, December 2017, Volume 76, Issue 24, pp 25547–25561.

[16] M. Chew and J. D. Tygar, "Image Recognition Captchas", in Proc. of the 7th Int. Information Security Conf. pp 268-279, Palo Alto, CA, September 2004.

[17] M. Chew and J. Doug Tygar, "Image Recognition Captchas", Tech. Rep. UCB/CSD-04-1333,EECS Department, University of California, Berkeley, August 2004.

[18] Y. Rui and Z. Liu, "Artifacial: Automated reverse turing test using facial features", in Proc. of the 11th ACM Int. Conf. on Multimedia, pp 295-298, New York, NY, USA, November 2003.

[19] Y. Rui and Z. Liu, "Excuse me, but are you human?", in Proc. of the 11th ACM Int. Conf. on Multimedia, pp 462-463, New York, NY, USA, November 2003.

[20] Y. Rui and Z. Liu, "ARTiFACIAL: Automated Reverse Turing test using FACIAL features", presented at Multimedia Syst., pp.493-502, June 2004.

[21] D. Misra and K. Gaj, "Face recognition CAPTCHAs", presented in Int. Conf. on Internet and Web Applications and Services/Advanced International Conf. on Telecommunications, pp 122, Washington, DC, USA, February 2006.

[22] H. S. Baird and J. L. Bentley, "Implicit Captchas", in Proc. of the IST SPIE Document Recognition and Retrieval XII Conf., San Jose, CA, USA, January 2005.

[23] Anju Bala and Baljit Singh Saini, "A Review of Bot Protection using CAPTCHA for Web Security,"(IOSR-JCE) IOSR Journal of Computer Engineering, Volume 8, Issue 6 (Jan. - Feb. 2013), 36- 42.

[24] H. Kwak, M. chew, P. Rodriguez, S. Moon and Y.Y. Ahn, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," In Proc. IMC 2007, ACM Press, 1–14.

[25] Preet Pal and Ved Prakash Singh, "Survey of Different Types of CAPTCHA," / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2242-2245.

INTENTIONAL BLANK

A SURVEY ON IMAGE SPAM DETECTION TECHNIQUES

Shadi Khawandi, Firas Abdallah,Anis Ismail

Faulty of Technology, Lebanese University, Lebanon

ABSTRACT

Today very important means of communication is the e-mail that allows people all over the world to communicate, share data, and perform business. Yet there is nothing worse than an inbox full of spam; i.e., information crafted to be delivered to a large number of recipients against their wishes. In this paper, we present a numerous anti-spam methods and solutions that have been proposed and deployed, but they are not effective because most mail servers rely on blacklists and rules engine leaving a big part on the user to identify the spam, while others rely on filters that might carry high false positive rate.

KEYWORDS

E-mail, Spam, anti-spam, mail server, filter.

1. INTRODUCTION

The internet community has grown and spread widely in a way that not only is it connecting every one of its users into one virtual globe, but also affecting them. Given that the internet is still in an ongoing evolution, states that this virtual community of people (users) is growing and with this growth comes great value, a value of people connected all together in a certain period of time all of the time, now imagine what this could bring forward as a target regarding marketing, advertisement, at the same time it could also hurt such users when such marketing and advertisement are misused, therefore affecting the resource structure of this globe along with its users. Consider a table whose resource structure are its four wooden legs which is able to hold a capacity of 50 kg, now bring a load of 70 kg and you will notice that the table would be crippled and broken, now apply that on the internet community whose resource structure are its communication which is able to hold up to a certain level of bandwidth, if we abuse that level and raise it up the internet community will be crippled and get affected by itself and its users thus costing the whole community a burden which starts from spam.

Spam is flooding the Internet with many copies of the same message, in an attempt to force the message on people who would not choose to receive it, and is also regarded as the electronic equivalent of junk mail. Most spam is commercial advertising and is generally e-mail advertising for some product sent to a mailing list or newsgroup. This is done by the abuse of electronic messaging systems including most broadcast media, digital delivery systems to send unsolicited bulk messages at random. While the most widely recognized form of spam is e-mail spam, the term is applied to similar abuses in other media: instant messaging spam, Usenet newsgroup

Dhinaharan Nagamalai et al. (Eds) : COMIT, AISCA - 2019
pp. 13–27, 2019. © CS & IT-CSCP 2019 DOI : 10.5121/csit.2019.90102

spam, Web search engine spam, spam in blogs, wiki spam, online classified ads spam, mobile phone messaging spam, Internet forum spam, junk fax transmissions, and file sharing network spam [1]. People who create electronic spam are called spammers [2].

The generally accepted version for source of spam is that it comes from the Monty Python song, "Spam spam spam spam, spam spam spam spam, lovely spam, wonderful spam…" Like the song, spam is an endless repetition of worthless text. Another thought maintains that it comes from the computer group lab at the University of Southern California who gave it the name because it has many of the same characteristics as the lunchmeat Spam that is nobody wants it or ever asks for it. No one ever eats it. It is the first item to be pushed to the side when eating the entree. Sometimes it is actually tasty, like 1% of junk mail that is really useful to some people [2].

E-mail spam is known as unsolicited bulk E-mail (UBE), junk mail, or unsolicited commercial e-mail (UCE), is a subset of spam where in practice it is the sending of unwanted e-mail messages, frequently with commercial content, in large quantities to a random set of recipients.Spam in e-mail started to become a problem when the Internet was opened up to the general public in the mid-1990s. It grew exponentially over the following years, and today is estimated to comprise some 80 to 85% of all the e-mail in the world [1].Digital image is a representation of a two-dimensional image using ones and zeros (binary). The term "digital image" usually refers to raster images also called bitmap images.Raster images have a finite set of digital values, called picture elements or pixels. The digital image contains a fixed number of rows and columns of pixels. Pixels are the smallest individual element in an image, holding quantized values that represent the brightness of a given color at any specific point.

Typically, the pixels are stored in computer memory as a raster image or raster map, a two-dimensional array of small integers. These values are often transmitted or stored in a compressed form which is the process of encoding information using fewer bits than an uuencoded representation would use. Raster images can be created by a variety of input devices and techniques, such as digital cameras, scanners, coordinate-measuring machines, seismographic profiling, airborne radar, and more. Each pixel of a raster image is typically associated to a specific 'position' in some 2D region, and has a value consisting of one or more quantities related to that position. Digital images can be classified according to the number and nature of those samples such as binary, grayscale, color, false-color, multi-spectral, thematic, and picture function [3].

Image spam is a kind of E-mail spam where the message text of the spam is presented as a picture in an image file. Since most modern graphical e-mail client software will render the image file by default by presenting the message image directly to the user, thus it is highly effective at overcoming normal e-mail filtering software where it inputs the e-mail, and as for its output it might pass the e-mail message through unchanged for delivery to the user's mailbox, redirect the message for delivery elsewhere, or even throw the message away.

2. EXISTING SOLUTIONS

This paragraph lists various solutions for tackling spam and image based spam, where the light is shed on the process and technique used to battle spam and the different features each solution contains. Also, the filtering steps that each solution requires to detect and prevent spam are presented.

2.1 SYMANTEC

Symantec is considered one of the important firms that specialize in security products including anti-spam ones and below the Symantec's Bright mail anti-spam product along with its components and their features are discussed here in.

2.1.1 SPAMMERS EMPLOYING TRADITIONAL TECHNIQUES

Security researchers at Symantec state that spammers have not discarded their old methods. Actually, in a wave of latest malware and spam crusades, spammers have revised and combined two oldest and commonly used topics. Symantec experts inform that they have observed the coming back of spam mails which hide their malicious content in HTML code embedded in the form of mail attachments. It is a known obfuscation technique which has been discarded in favor of other methods such as image spam.

Symantec also reveals that the image spam, responsible for the major increase in spam activity during May 2009, became even more constant in June 2009, accounting for between 8% and 10% of the total spam detected by the security vendor.Actually, what they fear is that these spam attacks will probably follow ever more diverse strategies in times to come as spammers are collectively working to advance their attack vectors. Mayur Kulkarni, Researcher at Symantec, claims that spammers do not have to discover new methods to enter user's inbox. They can very well use the existing method with even better results, as reported by security watch week on July 7, 2009. Lastly, the security vendor has asked users that they should not carelessly open any attachments especially when it is sent by an unknown sender. With 419 spam mails, e-mail users are suggested not to reply fake appeals and do not show interest in any of the money making plans.

2.1.2 SYMANTEC BRIGHTMAIL ANTISPAM

Symantec Brightmail AntiSpam™ offers complete, server-side anti-spam and antivirus protection. It actively seeks out, identifies, analyzes, and ultimately defuses spam and virus attacks before they trouble the users and overwhelm or damage the networks. Symantec Brightmail software that is installed at your site allows unwanted mail to be removed before it reaches the users' inboxes, without violating their privacy.

2.1.2.1 HOW SYMANTEC BRIGHTMAIL ANTISPAM WORKS

Symantec Brightmail AntiSpam employs the following four major types of filters. First, AntiSpam Filters are created by Symantec using the state-of-the art technologies and strategies to filter and classify e-mail as it enters the site, Second, Content Filters are custom content filters are written by the user, using the Brightmail Control Center or the Sieve scripting language, to tailor filtering to the needs of the organization. Third, Allowed and Blocked Senders Lists in which lists can be created of allowed senders and blocked senders and third party lists can also be used. The lists included in the Brightmail Reputation Service are deployed by default. Fourth, Antivirus Filters in which Antivirus definitions and engines protect the users from e-mail borne viruses.

2.1.2.2 FEATURES OF SYMANTEC BRIGHTMAIL ANTISPAM

AntiSpam Filtering Feature includes Heuristics that is a practical approach which targets patterns common in spam, Signatures that are Accurate and responsive approach that identifies the underlying "DNA" of evolving spam attacks. Defeats HTML-based and other evasion strategies used by spammers, Header that is similar to the Heuristics Filter, but applied to message headers, URL that matches the embedded URLs with a database of known spam URLs, Suspect List which Blocks e-mail from known spam senders (part of the Brightmail Reputation Service), Open Proxy List that blocks e-mail from insecure proxy servers by testing against the IP address of e-mail (part of the Brightmail Reputation Service), Safe List that allows e-mail from known clean domains (part of the Brightmail Reputation Service), Block and Allowed Senders Lists are Lists of trusted and blocked senders, IP connections, and domains created by administrators to augment Brightmail filtering, Content filters that are special purpose filters created by administrators to enforce organization-specific e-mail policies, and Third party filters which has easy integration with DNS-based blacklist and filtering services.

Other Filtering Features are group policies that specify groups of users, identified by e-mail addresses or domain names, and customize mail filtering for each group. Deployment options include gateway layer, internal relay layer, and e-mail server. The e-mail client add-ins for handling spam having Plug-ins for Outlook and Notes, and Web-based, with configurable notification option for recipients. Available antivirus protection detects and removes e-mail-borne viruses Quarantine Web-based, with configurable notification option for recipients.Spam management options in which to deliver the message normally, delete the message, deliver the message to the recipient's Spam folder, foldering agent moves spam to a designated folder in the end-user's mailbox, save the message to disk for administrator review, sends the message to an administrative account for further study, routes spam to a Web-based quarantine where recipients can review caught spam, and modify the message by adding configurable X-Header or subject line text to the message.Reporting and Statistics made up of standard interactive reports based on total spam or total virus messages found, and extended tracking and reporting of recipient, sender, domain, and other fields.

2.1.2.3 SYMANTEC BRIGHTMAIL ANTISPAM ARCHITECTURE

Symantec Brightmail AntiSpam consists of several components. The key components you need to consider are the following:

- Each Symantec Brightmail AntiSpam installation can have one or more Brightmail Scanners. Brightmail Scanners perform the actual filtering of e-mail messages.
- Each Brightmail Scanner contains a Brightmail Agent, and One or both of a Brightmail Server, and a Brightmail Client. If the Brightmail Scanner contains a Brightmail Client, then a supported mail transfer agent (MTA) must also reside on the same computer.

The Brightmail Client is a communications channel between the MTA and the BrightmailServer. You can use multiple Brightmail Clients each one can talk to multiple Brightmail Servers. The Brightmail Client performs load balancing between Brightmail Servers.The Brightmail Servers at your site process spam based on configuration options you select. Each Brightmail Server is a multi-threaded process that listens for requests from Brightmail Clients. Using a variety of state-

of-the-art technologies, the Bright mail Server filters messages for classification. The classification, or verdict, is then returned to the Brightmail Client for successive delivery action. The Conduit connects to the BLOC to determine whether updated filtering rules are available. If new rules are available, the Conduit retrieves the updated rules using secure HTTPS file transfer. After authenticating the rules, the Conduit notifies the Bright mail Server to begin using the updated rules. The Conduit also manages statistics, both for use by the BLOC and in a local statistics pool for the generation of local reports. Each Symantec Bright mail Anti-Spam installation has exactly one Bright mail Control Center. This is the central nervous system of your Symantec software. The Bright mail Control Center communicates with the Brightmail Agent on each of your Brightmail Scanners. For smaller installations, you can install the Brightmail Control Center and theBrightmail Scanner on the same computer. From this Web-based graphical user interface, you can configure start and stop each of your Brightmail Scanners, specify e-mail filtering options for groups of users or for all of your users at once, monitor consolidated reports and logs for all Brightmail Scanners, view summary and status information, administer Brightmail Quarantine, and view online help for Brightmail Control Center screens.

The Brightmail Control Center contains the following Features:

- Brightmail Quarantine provides storage of spam messages and Web-based end user access to spam. You can also configure Brightmail Quarantine for administrator-only access. Use of Brightmail Quarantine is optional.
- A single MySQL database stores all of your Symantec Brightmail AntiSpam configuration information, as well as Brightmail Quarantine information and e-mails (if you are using Brightmail Quarantine). Configuration information is communicated to each Brightmail Scanner via an XML file. A Java-based Web Server (by default this is the Tomcat Web Server) performs Web hosting functions for the Brightmail Control Center and Brightmail Quarantine.

2.1.2.4 SYMANTEC BRIGHTMAIL ANTISPAM FILTERING PROCESS

With the default configuration, the filtering process works as follows. First, the SMTP server receives the mail message and processes any security settings. Second, the Brightmail Client (integrated with the MTA) sends a copy of the mail message to the Brightmail Server. Third, by default the Brightmail Server processes mail in the following order, allowed senders you identify, blocked senders you identify, Symantec Brightmail AntiSpam filters, content filters you create and finally the Brightmail Server returns the verdict of the message to the Brightmail Client. Fourth, the Brightmail Client tells the SMTP server to perform the appropriate action, based on the policies in place [4].The Bright mail anti-spam solution is composed of several components and these components need to interact with each other in order to provide the feature that is needed from it and these interactions are shown in Figure 1.

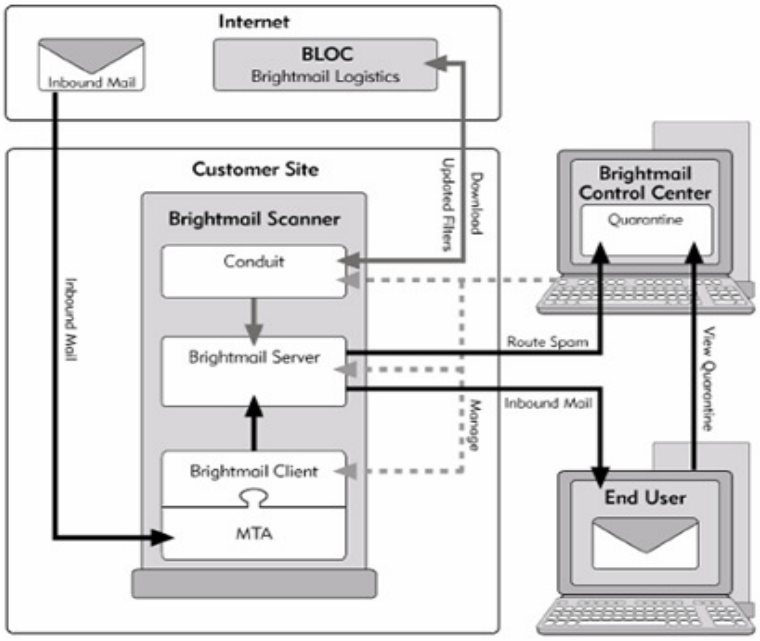


Figure 1 - Symantec Bright mail Components Interaction

2.2 KASPERSKY

Although Internet Security suites usually include as standard e-mail spam filter, spammers continue to find ways around the checks that are made. One of those workarounds is the use of images with text buried in the image data. This kind of spam can be checked for, but currently it is done using machine recognition. Spammers can overcome those checks by making the text fuzzy and adding distortion or rotation to an image.Kaspersky Lab has a statistics-based method for detecting image-based spam that is used to bypass traditional text-based filters. The technology analyses whether text is contained in images based on the graphic pattern of words and lines, said developer Eugene Smirnov.Spam is expected to continue to be a problem in 2009, particularly with the rise in the number and popularity of websites that allow user-generated content. Kaspersky Anti-Spam 3.0 provides thorough and accurate protection from spam for users of corporate mail systems and public e-mail services.

2.2.1 KASPERSKY ANTI-SPAM

There are several features that are offered by Kaspersky Anti-spam solution and these features include the following.

2.2.1.1 PROTECTION FROM SPAM

List-based filtrations in which sender's IP addresses are checked against blacklists of spammers, which are maintained by Internet service providers and public organizations (DNS-based Blackhole Lists). System administrators can add addresses of trusted correspondents to a safe list, ensuring that their messages are always delivered without undergoing filtration.Analysis of formal attributes where the program recognizes spam by such typical characteristics as distorted sender addresses or the absence of the sender's IP address in DNS, an excessive number of

intended recipients or hidden addresses. The size and format of messages are also taken into consideration.Linguistic heuristics where the program scans messages for words and phrases that are typical of spam messages. Both the content of the message itself and any attachments are analyzed.Graphic spam in which a database of signatures for graphic spam equips the program to block messages containing spam images, a type of spam that has become increasingly common in recent years.Real-time UDS requests where the Urgent Detection System is updated with information on spam messages literally seconds after they first appear on the Internet. Messages that could not be assigned a definitive status (e.g., spam, no-spam) can be scanned using UDS.The e-mail that is received passes into a process of message analysis as shown in Figure 2 and includes several analysis procedures in order to analyze the message

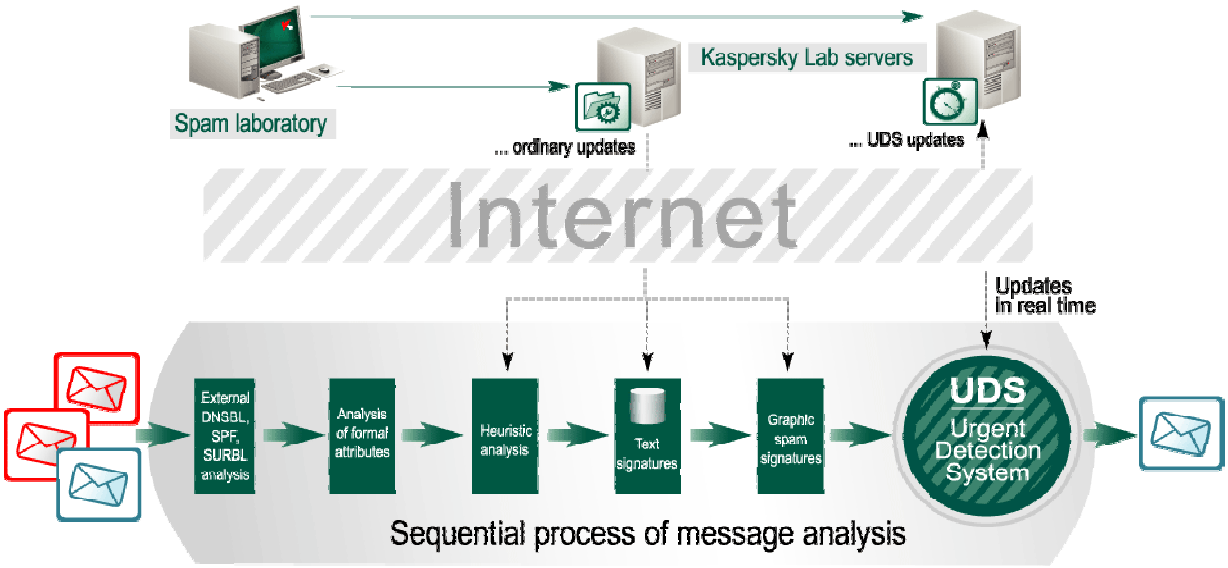


Figure 2 - Kaspersky E-mail analysis process

2.2.2.2 ADMINISTRATION

Flexible management in which the web interface allows system administrators to manage the application both locally and remotely. The filtration level is easily configurable, as are blacklists and safe lists. It is also possible to disable/enable individual filtration rules.Management of user groups where the administrator can create user groups either using lists of addresses or domain masks (for example, XXX@domain.com) and apply individual settings and filtration rules to each group.Options for processing spam where the program can be configured to process spam by either automatically deleting it, redirecting it to the quarantine folder with a note to the user or sent for further filtration to the mail client.Detailed reports where the administrators can easily monitor the application, the protection status and license status, using HTML reports or alternatively, by viewing log files. Data can be exported in CSV and Excel formats.

2.2.2KASPERSKY ANTI-SPAM 3.0 MP1 CRITICAL FIX

The following improvements have been introduced since Kaspersky Anti-Spam 3.0 MP1 (3.0.255.0) where methods for fighting so-called "graphic" spam, i.e. tools used to analyze graphic attachments. New algorithms have been introduced for processing and identification of

similar images with textual content as well as the GSG-8 and GSG-9 technologies.The following problems have been fixed as compared to Kaspersky Anti-Spam 3.0 MP1 CF1 (3.0.274.0) where possible termination or freezing of filtering processes when a list of protected domains is used, and accidental setting of incorrect access rights for the files of application components if they were previously updated using a package of modified application files from previous product versions [5].

2.3 TREND MICRO

It's no longer efficient to compile lists of known spammers and filter them out, because those lists are so large and growing bigger all the time, adds Hemmendinger. And it's too cumbersome to update them on a daily basis. "What we've learned over time is the more commonly used methods would be content filtering, like text filters that look for certain key words or sophisticated heuristics that look at the content of a message to see if it appears to fit the mold of what is readily recognized as spam," Hemmendinger says. He also pointed to techniques that spammers use to trip up e-mail filters, like adding asterisks between each letter in a word so it can't be identified." With the release of InterScan Messaging Server Suite (IMSS), Trend Micro strives to provide solution providers with effective tools to battle spam and protect users from increasing ills associated with e-mail, ranging from script bombs to worm-bearing messages.

In Figure 3 we can have a view on a snap shot of the Trend Micro IMS anti-spam solution which shows the configuration that can be altered or given by the user

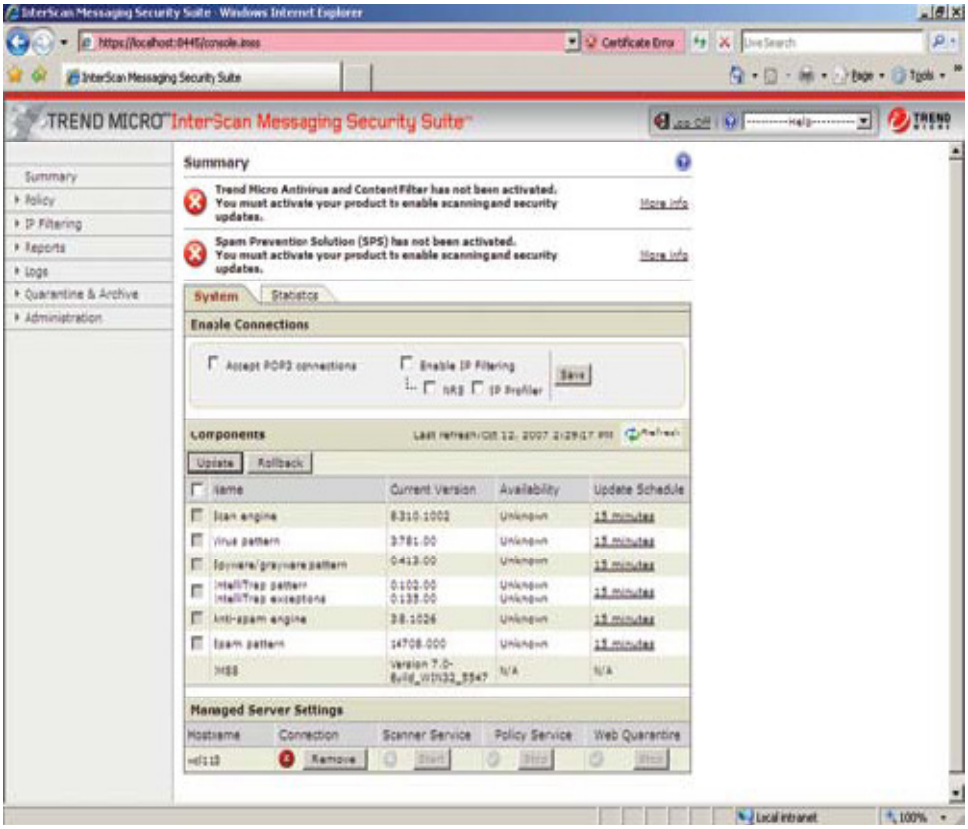


Figure 3 - Trend Micro InterScan Messaging Security

2.3.1 INTERSCAN MESSAGING SERVER SUITE FILTERING PROCESS

First, a message passes through Trend Micro's 32-bit virus scan engine. After the messages are checked for viruses, they're passed off to the content management portion of IMSS, that process is the key to battling spam and other e-mail-related problems. Trend Micro directs the advantage of policies toward content filtering, and those policies allow complete control of e-mail beyond spam management. Solution providers can script policies that prevent confidential data from being transmitted or create policies that identify unwanted messages. Policies clearly define what acceptable use of company e-mail is and what is not.

The primary reason for using IMSS is controlling spam. While policies can offer some protection from spam, the real answer to effectively fighting it lies with automation. IMSS employs complex heuristics to identify spam. Every message is examined for phrases or content that fits the profile of a spam message, and anti-spam heuristics can be tuned to filter based on content and determine how aggressively the antispam filtering should be applied. Administrators have several options for handling e-mail identified as spam. They can add the word "spam" to the subject line, redirect the suspect e-mail or quarantine the e-mail.

2.3.2 TREND MICRO SPAM PREVENTION SOLUTION

Spam Prevention Solution offers a comprehensive, multi-tiered spam and phishing defense. Three distinct tiers of anti-spam protection include E-mail Reputation, IP Profiler, and the anti-spam composite engine. The solution uses multiple techniques to keep threats completely off of the network, securing the network and preserving bandwidth, storage, and other network resources. Spam Prevention Solution includes patent-pending image spam detection technology and other cutting-edge approaches to protect organizations as spam and phishing threats evolve where it blocks most spam before it even reaches the gateway, uses the world’s largest most trusted reputation database, deploy dynamic reputation services to stop zombies and botnets as they first emerge, blocks e-mail senders that exceed threat thresholds set by the organization providing protection customized to the organization’s e-mail traffic, delivers automatic customer specific reputation services to stop spam, creates a firewall against bounced mail attacks, and combines multiple protective techniques including statistical analysis, advanced heuristics, whitelists, and blacklists. Also, it includes Features image spam detection and other cutting-edge technologies, content filtering and expanded language support to improve spam protection for global companies, provides dedicated anti-phishing techniques, including signatures, and reputation services to stop both corporate and consumer phishing attacks. Furthermore, it offers single Web-based management console to customize spam tolerance settings, create approved sender lists, establish filter actions, and set policies for individuals or groups. Moreover, it simplifies administration through LDAP integration, delegated administration, and message tracking. In addition to enabling end users to manage their own spam with Web-based End-User Quarantine and quarantine notification e-mails.

2.3.3 POLICIES OR RULE BASED DETECTION MISHAPS

Antivirus firm Trend Micro unwittingly targeted the letter "P" with a recent rules update, forcing all e-mail containing the objectionable letter into quarantine. According to their knowledge base article titled Solution 14638, "Antispam Rule 915 unintentionally blocks some legitimate e-mails scanned by InterScan eManager and ScanMail eManager." The cause is the letter P.

According to Trend Micro, the problem affects their Internet gateway, e-mail and groupware products, including InterScan Messaging Security Suite, InterScan eManager, ScanMail for Exchange, ScanMail eManager, and ScanMail for Lotus Notes. A spokesman for Trend Micro declined to comment on the issue, stating only that "we've notified customers and resellers." According to Internet Week, much of that contact was done via e-mail. One can only imagine the difficulty of composing an e-mail describing the nature of the problem while simultaneously avoiding the use of the letter P.

Trend Micro advises that the unfortunate P mishap can be resolved by updating to Antispam Rule 916 or later. Several of their products include options to resend e-mails erroneously quarantined by the filtering rules. Their Knowledge Base article Solution 14638 contains links to the support solutions for these products [6].

2.4 MAIL-SECURE

The Mail-Secure anti-spam solution is a product of the PineApp firm which uses pattern detection and includes the following features.

2.4.1 IMAGE SPAM DEFENSE

Spammers are consistently creating sophisticated new weapons in their arms race with anti-spam technology, the latest of which is image-based spam. The number of unsolicited messages containing images has grown significantly throughout 2006, and is expected to continue to grow and spread.

Through constant monitoring, PineApp has identified that image-based spam tends to be distributed in massive waves at one of the distribution peaks, PineApp measured image-based spam as 30% of all global spam. Image-based spam creates bandwidth and storage problems, since the typical image based spam message weighs more than three times that of a regular spam message. At the image-spam distribution peaks, the bandwidth and storage requirements increase upwards of 70%. Also, Image-based spam is a new and growing problem leading to loss of productivity and a drain on IT resources, most anti-spam solutions have problems dealing with image-based spam, and by dealing with it ineffectively they create other problems along the Way. Thus, PineApp has implemented a unique solution to decode images, and treat them with RPD similarly to other types of spam which improves the already superior spam catch rate, and maintains low false positive rate

2.4.2 NEWEST TRENDS IN IMAGE-BASED SPAM

Lately, spammers have been experimenting with new techniques such as broken images i.e. splitting a single image into smaller images that fit together like puzzle pieces. This technique makes it even more difficult for anti-spam engines to catch and block.

2.4.3 MAIL-SECURE FILTERING PROCESS

The web-based interface, presented to the user upon logging in, is very easy to use and clutter free. The interface presents its data in a clear and straightforward manner, with minimal delay when saving any configuration changes. For added security, the interface also includes a timeout

function, returning the administrator to the login page after a set period of inactivity. Mail-SeCure's method of protecting against spam is controlled through the use of policies.

Mail-SeCure is a leading perimeter security appliance that protects all sized organizations (from 50 up to 10,000 users), from both targeted and non-targeted e-mail-related threats such as spam, viruses and malicious code.Mail-SeCure from PineApp is a gateway level device designed to offer e-mail protection to small or medium sized companies with support for up to 500 users. While this test was primarily concerned with spam detection, it should be noted that Mail-SeCure also provides protection from e-mail borne malware.

Configuration of Mail-SeCure is made simple by the provision of a well-written and easy to follow quick installation guide. Within the policies configuration screen, there are four separate rule groups available to the administrator. These are Attachment, Spam, General, and Black & White Rules. Each of these rule groups shares a similar layout, allowing for familiarization with the method by which these rules may be configured. When dealing with spam, Mail-SeCure splits the traffic into one of three types Local to Local, Remote to Local, and Local to Remote, effectively covering both internal and external mail. Each of these three traffic types may have its own policy. For the purposes of reviewing statistics relating to processed e-mail, Mail-SeCure provide five separate report pages. Included among these are Summary, Reports, User Reports, Domain Reports, and Statistics [7][8].

Figure 4 shows a sample e-mail message that contains two parts within its body, one part is an image that has written spam text embedded in it and the other is a legitimate text written beneath the image crafted in order to foil anti-spam solutions.

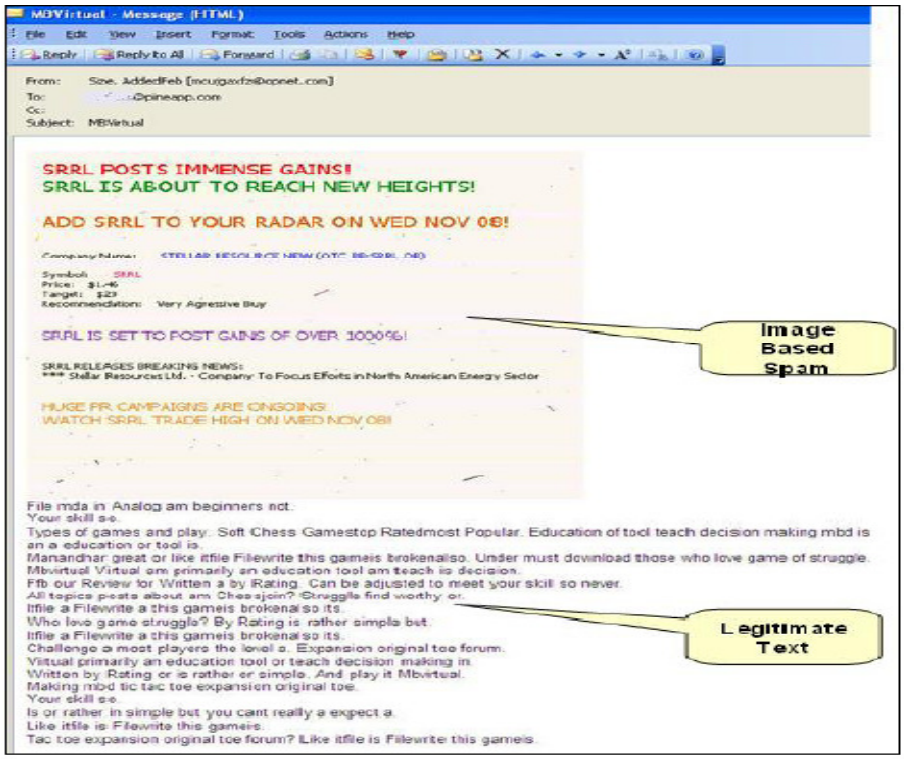


Figure 4 – E-mail Spam sample

2.5 PUBLICATIONS AND LITERATURE

Zhe Wang, William Josephson, Qin Lv, Moses Charikar, Kai Li[9] in Filtering Image Spam with Near-Duplicate Detection propose an image spam detection system that uses near-duplicate detection to detect spam images, they rely on traditional anti-spam methods to detect a subset of spam images and then use multiple image spam filters to detect all the spam images that "look" like the spam caught by traditional methods. Battista Biggio, Giorgio Fumera, Ignazio Pillai, Fabio Roli[10] in Image Spam Filtering by Content Obscuring Detection propose an approach based on low-level image processing techniques to detect one of the main characteristics of most image spam, namely the use of content obscuring techniques to defeat OCR tools by finding the noise level of a certain image spam. Jason R. Bowling, Priscilla Hope, Kathy J. Liszka[11] in Spam Image Identification Using an Artificial Neural Network propose a method for identifying image spam by using FANN (Fast Artificial Neural Network) library model and training the artificial neural network. A detailed process for preprocessing spam image files is given, followed by a description on how to train an artificial neural network to distinguish between ham and spam. M. Muztaba Fuad, Debzani Deb, M. Shahriar Hossain[12] in A Trainable Fuzzy Spam Detection System presents the design and implementation of a trainable fuzzy logic based e-mail classification system that learns the most effective fuzzy rules during the training phase and then applies the fuzzy control model to classify unseen messages. M. Soranamageswari, C. Meena[13] in Statistical Feature Extraction for Classification of Image Spam Using Artificial Neural Networks present an experimental system for the classification of image spam by considering statistical image feature histogram and mean value of an block of image. A comparative study of image classification based on color histogram and mean value is presented.

Ms.D.Karthika Renuka, Dr.T.Hamsapriya, Mr.M.Raja Chakkaravarthi and Ms.P.Lakshmisurya (2011)performed a comparative analysis on spam classification based on supervised learning using several machinelearning techniques. In this analysis, the comparison was done using three different machine learningclassification algorithms viz. Naïve Bayes, J48 and Multilayer perceptron (MLP) classifier. Resultsdemonstrated high accuracy for MLP but high time consumption. While Naïve Bayes accuracy was low thanMLP but was fast enough in execution and learning. The accuracy of Naïve Bayes was enhanced using FBLfeature selection and used filtered Bayesian Learning with Naïve Bayes. The modified Naïve Bayes showed theaccuracy of 91% as in [14].

Rushdi Shams and Robert E. Mercer (2013) performed a comparative analysis on classification of spam emailsby using text and readability features. This paper proposed an efficient spam classification method along with feature selection using content of emails and readability. This paper used four datasets such as CSDMC2010,Spam Assassin, Ling Spam, and Enron-spam. Features are categorized into three categories i.e. traditionalfeatures, test features and readability features. The proposed approach is able to classify emails of any languagebecause the features are kept independent of the languages. This paper used five classification based algorithmfor spam detection viz. Random Forest (RF), Bagging, Adaboostm 1, Support Vector Machine (SVM) andNaïve Bayes (NB). Results comparison among different classifiers predicted Bagging algorithm to be the bestfor spam detection as in [15].

Megha Rathi and Vikas Pareek(2013) performed an analysis on spam email detection through Data Mining byperforming analysis on classifiers by selecting and without selecting the features as in [16].

Anirudh Harisinghaney, Aman Dixit, Saurabh Gupta and Anuja Arora (2014) performed a comparative analysison text and images by using KNN, Naïve Bayes and Reverse-DBSCAN Algorithm for email spam detection.This analysis paper proposed a methodology for detecting text and spam emails. They used Naïve Bayes, K-NNand a modified Reverse DBSCAN (Density-Based Spatial Clustering of Application with Noise) algorithm's.Authors used Enron dataset for text and image spam classification. They used Google's open source library,Tasseract for extracting words from images. Results show that these three machine learning algorithms givesbetter results without preprocessing among which Naïve Bayes algorithm is highly accurate than otheralgorithms as in [17].

Savita Pundalik Teli and Santosh Kumar Biradar (2014) performed an analysis on effective email classificationfor spam and non-spam emails as in [18].

Izzat Alsmadi and Ikdam Alhami (2015) performed an analysis on clustering and classification of email contentsfor the detection of spam. This paper collected a large dataset of personal emails for the spam detection ofemails based on folder and subject classification. Supervised approach viz. classification along-sideunsupervised approach viz. clustering was performed on the personal dataset. This paper used SVMclassification algorithm for classifying the data obtained from K-means clustering algorithm. This paperperformed three types of classification viz. without removing stop words, removing stop words and using Ngrambased classification. The results clearly illustrated that N-gram based classification for spam detection isthe best approach for large and Bi-language text as in [19].

Ali Shafigh Aski and Navid Khalilzadeh Sourati (2016) performed an analysis using Machine Learning". Thispaper utilized three machine learning algorithms viz. Multi-Layer Neural Network, J48 and Naïve BayesClassifier for detection of spam mails from ham mails using 23 rules. The model demonstrated high accuracy incase of MLP with high time for execution while Naïve Bayes showed slightly less accuracy than MLP and alsolow execution time as in [20].

3. CONCLUSIONS

Image Spam detection have been causing problems from the first day it was known and up till now with all the solutions that have been developed by various venders and users, it still poses a great threat and still able to penetrate to the user's e-mail and up till now various venders still look at enhancing and updating their algorithms in order to achieve a higher detection rate with lowers false positive, and the reason that keeps this ongoing problem is the ways that the spammers are employing to fool those algorithms. In this paper, we introduced some of the available solutions for tackling spam and image based spam, where the light is shed on the process and technique used to battle spam and the different features each solution contains.

REFERENCES

[1] Cormack, Gordon V (2008), Email Spam Filtering: A Systematic Review. Now Publishers Inc, ISBN 978-1601981462

[2] Gyöngyi, Zoltán; Garcia-Molina, Hector (2005), "Web spam taxonomy", Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb), 2005 in The 14th International World Wide Web Conference (WWW 2005) May 10, (Tue)-14 (Sat), 2005, Nippon Convention Center (Makuhari Messe), Chiba, Japan., New York, N.Y.: ACM Press, ISBN 1-59593-046-9

[3] Gonzalez, Rafael C.; Woods, Richard E. (2008), Digital Image Processing, 3rd edition, Prentice Hall, ISBN 9780131687288

[4] Symantec Corporation (2008), Symantec Brightmail AntiSpam Deployment Planning Guide

[5] Kaspersky Lab (2008), Kaspersky Anti-Spam 3.0 Administrators Guide

[6] Trend Micro Incorperated (2008), Trend Micro ScanMail, InterScan Security Guide

[7] PineApp Ltd (2007), Mail-SeCure Perimeter Security white paper

[8] PineApp Ltd (August 2009), Mail-SeCure Image-Based Spam Treatment white paper

[9] Zhe Wang; William Josephson; Qin Lv; Moses Charikar; Kai Li (2008), Filtering Image Spam with Near-Duplicate Detection, In Conference on E-mail and Anti-Spam (CEAS)

[10] Battista Biggio; Giorgio Fumera; Ignazio Pillai; Fabio Roli (2008), Image Spam Filtering by Content Obscuring Detection, In International Conference on Image Analysis and Processing (ICIAP)

[11] Jason R. Bowling; Priscilla Hope; Kathy J. Liszka (2009), Spam Image Identification Using an Artificial Neural Network, In International Conference on Artificial Neural Networks (ICANN)

[12] M. Muztaba Fuad; Debzani Deb; M. Shahriar Hossain (2005), A Trainable Fuzzy Spam Detection System, In International Conference on Computer and Information Technology (ICCIT)

[13] M. Soranamageswari; C. Meena (2010), Statistical Feature Extraction for Classification of Image Spam Using Artificial Neural Networks, In International Conference on Machine Learning and Cybernetics (ICMLC)

[14] D. K. Renuka, T. Hamsapriya, M. R. Chakkaravarthi and P. L. Surya, “Spam Classification Based onSupervised Learning Using Machine Learning Techniques”, in proc. IEEE- International Conference on Process Automation, Control and Computing, 2011, pp. 1–7.

[15] R. Shams and R. E. Mercer, “Classifying spam emails using text and readability features”, in proc. IEEEInternational Conference on Data Mining (ICDM), 2013, pp. 657–666..

[16] M. Rathi and V. Pareek, “Spam Email Detection through Data Mining-A Comparative Performance Analysis”, in International Journal of Modern Education and Computer Science, vol. 12, pp. 31-39, 2013.

[17] A. Harisinghaney, A. Dixit, S. Gupta, and Anuja Arora, “Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN Algorithm”, in proc. IEEE-International Conference on Reliability, Optimization and Information Technology (ICROIT), 2014, pp.153-155.

[18] S. P. Teli and S. K. Biradar, “Effective Email Classification for Spam and Non- spam”, in International Journal of Advanced Research in Computer and software Engineering, Vol. 4, 2014.

[19] Alsmadi and I. Alhami, “Clustering and classification of email contents”, in Journal of King Saud University - Computer and Information Science -Elsevier, vol. 27, no. 1, pp. 46–57, 2015.

[20] A. S. Aski and N. K. Sourati, “Proposed efficient algorithm to filter spam using machine learning techniques”, in Pacific Science Review- A Natural Science Engineering- Elsevier, Vol. 18, No. 2, pp. 145–149, 2016.

INTER-APPLICATION COMMUNICATION: A PROTOTYPE IMPLEMENTATION

Kalaiselvi Arunachalam[1], Gopinath Ganapathy[2]

[1]School of Computer Science, Engineering and Applications,
Bharathidasan University, India
[2]Registrar, Bharathidasan University, India

ABSTRACT

A growing popularity of smart devices of various type, shape and form factor with multitude of applications from diverse categories and data are used to meet the demands of users in their digitally enriched living environment. The data sharing between these applications would be beneficial to the users when these heterogeneous devices are used together by them in their home network. The inter-application communication enables an application to discover, connect and share data with other applications across heterogeneous devices in a home network. This paper provides a prototype implementation of the inter-application communication in a home network along with a brief summary about its demand in near future.

KEYWORDS

Inter-application Communication, Prototype Implementation, Home Network, App-to-App Communication, Heterogeneous Devices

1. INTRODUCTION

The popularity of smart devices of various type, shape, form factor and the heterogeneity of applications on these devices with multitude of data like text, image, audio, video, graphics, hyperlink etc. to serve the users around the world in their digital living environment. It would be effective and beneficial to the user to share data directly between these heterogeneous applications across smart devices without any additional requirement of intermediary hardware or software**.** This paper provides a prototype implementation of inter-application communication in a home network in which two applications that reside on two different device types and operating systems can discover, connect and share data with each other. The inter-application communication would enable data sharing among heterogeneous applications by limiting additional installation, configuration, upgrade, porting, migration etc.

2. RELATED WORKS

The inter-application communication (IAC) promotes the development of feature rich applications that are discoverable, data sharable, compatible and interoperable together in a network. Some inter-application communication mechanisms were proposed earlier as below but they are limited with particular aspects.

A primary application uses an unique identifier for its identification and a random number for certifying the responsive communication received by it as in [1] where by the recipient application validates the unique identifier of the primary application and if it is valid, then it generates a second random number for certifying the subsequent communication submitted to it

Dhinaharan Nagamalai et al. (Eds) : COMIT, AISCA - 2019
pp. 27 –36, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.90103

as in [1] but this system requires a mobile device to provide common platform, processor and memory for the communication between the applications. Another mechanism in which a system that automatically generates an interface code which provides a single comprehensive adaptation interface that integrates multiple executable applications which can be plugged in to the adaptation interface to support IAC as in [2] but it would be tedious to provide an interface code for different platforms. A proposed framework to develop portable software application that support a single design for deployment on multiple target platforms for mobile applications as in [3] and even though this framework conducts portability check on the devices, there is no feature for compatibility check on applications towards the interoperability among them. As in [4], the proposed system in which two applications are involved to share the authorization details to login to a resource server to access the network resources and the IAC is limited here to be used for single-sign on mechanism only. An audio based data sharing between mobile applications on two devices is proposed as in [5] in which data is transferred in the form of audio messages which are communicated over a speaker and microphone that are interfaced to the mobile devices but this mechanism requires very close proximity between the devices for communication and sufficient volume for the audio messages as well.

A data sharing system proposed in which multiple applications can share data between them as in [6] using a data sharing zone that shares the data based on the access policy by using the access rights assigned by the applications but the sharing zone can be only accessed within a device and it is not accessible by applications on other devices. The sharing of data between smart devices in a network as in [7] defines the sharing of an image across various smart devices in a network and this system is device-specific only and not application-specific. A shared storage location is used by a system in which the first mobile application uses a shared encryption key to encrypt the data to be transferred from it and a second mobile application is configured to retrieve the encrypted data from this shared storage location as in [8] but the shared storage location can be accessed only by the applications within a device. A proposed system that defines sharing of data across multiple electronic devices as in [9] in which sharing of data between devices carried out through a file sharing session and the data transfer across applications residing on these devices is limited in this system.

3. SMART DEVICES AND APPLICATIONS

A smart device is an electronic device that can connect and share data with other devices in a network through Ethernet, Wi-Fi, Bluetooth, NFC etc. There are several types of smart devices like Smartphone, Tablet, Notebook, Smart TV, Smart watch, Smart band, Smart key chain, Smart glass etc. as in [10]. There are billions of smart devices used by the people around the world and millions of applications as in [Table 1] from various categories are available in the popular application stores like Google Play as in [11], App Store as in [12], Windows Store [13], BlackBerry World as in [14] for the users. These applications are available for each mobile operating system separately on these stores and are optimized for the devices of different types and sizes for each mobile operating system. Some of the popular application categories that are available on all popular application stores are Books, Business, Education, Entertainment, Finance, Games, Health & Fitness, Lifestyle, Magazines & News, Maps & Navigation, Music & Audio, Photo & Video, Productivity, Shopping, Social Networking, Sports, Travel and Utilities as in [11-14]. These applications involve various types of data like text, number, date, time, image, graphics, audio, video etc. which are required by the users in their daily life.

Table 1. Popular App Stores vs. Applications

App Store Name	Operating System	Total Apps (as of Dec. 2018)
Google Play	Android	2,100,000
App Store	iOS	2,000,000
Windows Store	Microsoft Windows	669,000
BlackBerry World	BlackBerry OS	234,500
Amazon App Store	Android, BlackBerry	450,000

The number of applications available on these application stores are increasing every day as in [15] and hundreds of applications are added to these stores each day by the developers around the world. Due to the limitations of each mobile operating system with respect to privacy, security, confidentiality, integrity etc., the applications are limited within their operating system or device itself. Hence the inter-application communication is very limited in Android, iOS, Windows Phone etc. as in [16] in a home network.

4. PROTOTYPE IMPLEMENTATION

There are several types of applications like games, social media, entertainment etc. installed on smart devices like Smartphone, Tablet, Notebook, Smart TV etc. These applications are very limited to share their data with other applications on other devices in a home network. If inter-application communication feature is implemented on an application, then an application on a device can discover, connect and share data with other application on another device as in [17]. Based on the proposed architecture for inter-application communication in a home network as in [18], this prototype is implemented to demonstrate the inter-application communication in a home network. This prototype implementation includes two applications that reside on two different operating systems and devices can discover, communicate and share data with each other.

4.1. EXAMPLE SCENARIO OF INTER-APPLICATION COMMUNICATION

A Phone Contact Book application on a Windows 10 Notebook can send a selected contact's phone number to a Phone Dialer application on an Android Smartphone to call that contact in a home network as in [Figure 1].

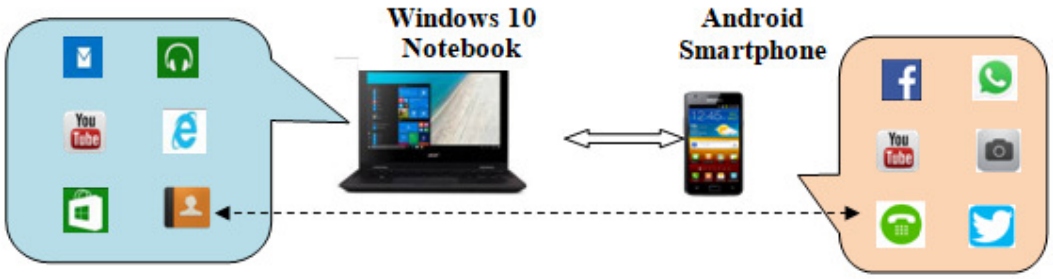


Figure 1. Inter-application communication between a My Contact Book and My Phone Dialer application on two devices in a home network

4.2. IMPLEMENTATION ENVIRONMENT

This prototype implementation includes two applications: My Contact Book and My Phone Dialer. The My Contact Book is a desktop application to store the details of contacts like name and phone number. My Contact Book app is based on Microsoft Windows 10 operating system.

The My Phone Dialer is a mobile application to make phone call to a contact. My Phone Dialer app is based on Android 9 operating system.

4.3. PROCESS FLOW OF APPLICATIONS

The My Contact Book app to be installed on a Notebook with Windows 10 operating system and the My Phone Dialer app to be installed on a Smartphone with Android 9 operating system. The process flow of the prototype implementation as in [Figure 2, 3, 4] is described in detail here.

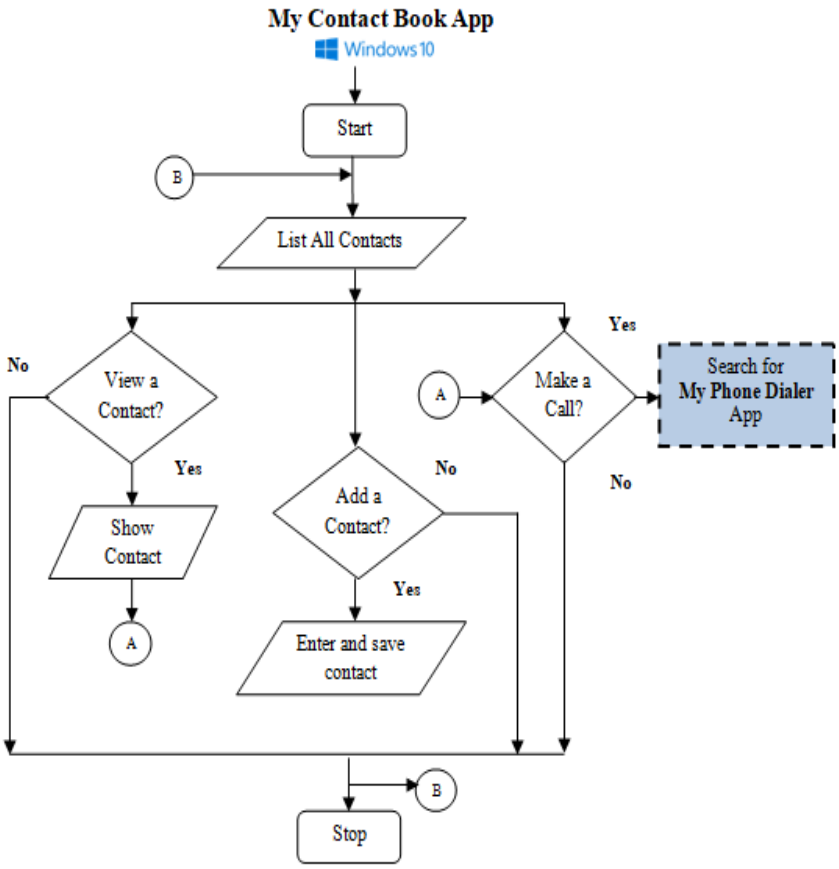


Figure 2. Process flow of My Contact Book App on a Windows 10 Notebook in a home network

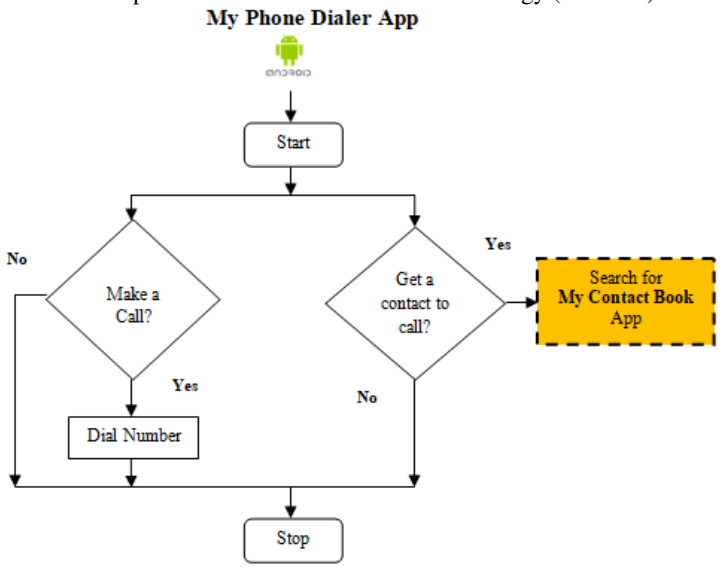


Figure 3. Process flow of My Phone Dialer App on an Android Smartphone in a home network

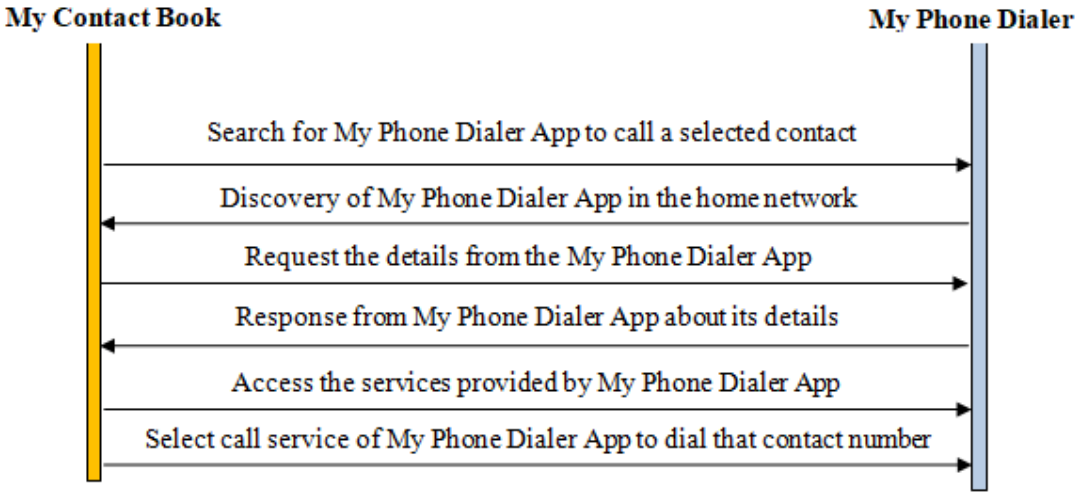


Figure 4. Inter-application communication flow between the My Contact Book App and My Phone Dialer App in a home network

When My Contact Book app is started on the Windows Notebook, new contacts can be added and the details of already existing contacts can be viewed by the user. Once a contact is selected, then the My contact Book app searches for the availability of My Phone Dialer app in the home network to call that contact. If the My Phone Dialer app is not started on an Android Smartphone, then it is not discovered by the My Contact Book app. When the My Phone Dialer app is started on the Smartphone by the user, then it is discovered by the My Contact Book app. Once if it is identified, then the user can view the details of the My Phone Dialer app like app name, app developer, version and its services ("call" service). Once the user selected the "call" service, then My Phone Dialer app automatically dials the phone number of the selected contact on Smartphone. When My Phone Dialer app is stopped or closed, then it is not discovered by the My Contact Book app and vice versa in the home network.

The availability of both applications on these devices is instantly identified by the user in the home network through the application discovery feature embedded in these two applications. The inter-application communication is implemented in this prototype whereby these two applications can discover, connect and share data with each other despite of their underlying operating system and device type.

4.4. USER INTERFACE OF APPLICATIONS

The user interface of the My Contact Book app as in [Figure 5-8, 11-12] and My Phone Dialer app as in [Figure 9, 10, 13-20] provides the flow of communication between these two applications in the home network.

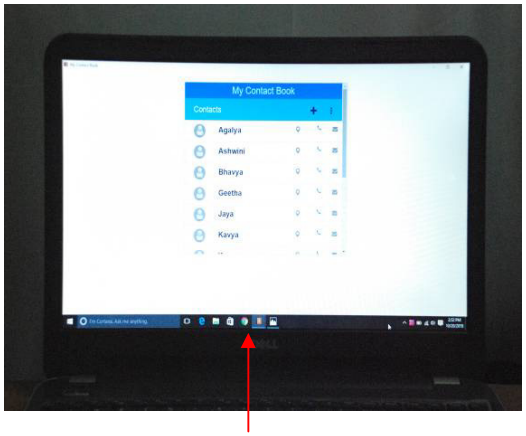


Figure 5. My Contact Book app running on Windows 10 Notebook

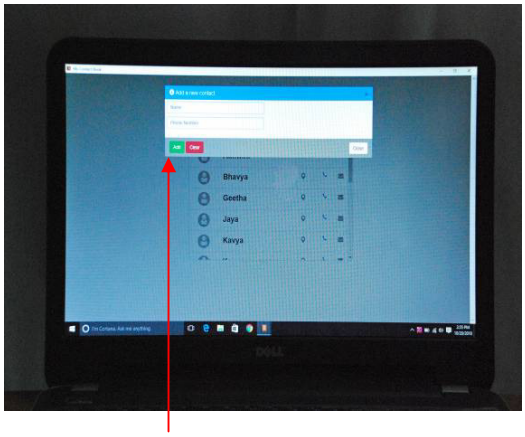


Figure 6. Add a contact in My Contact Book app

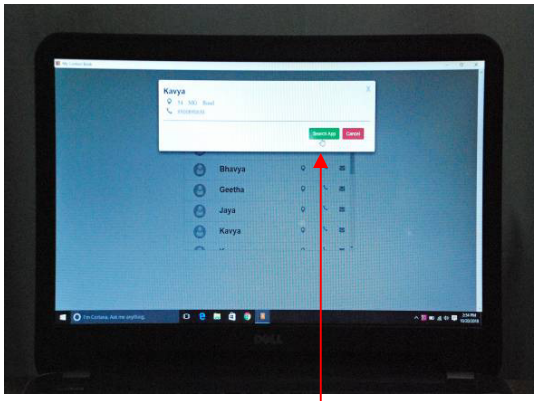


Figure 7. Select a contact and search for a Dialer app to make a call

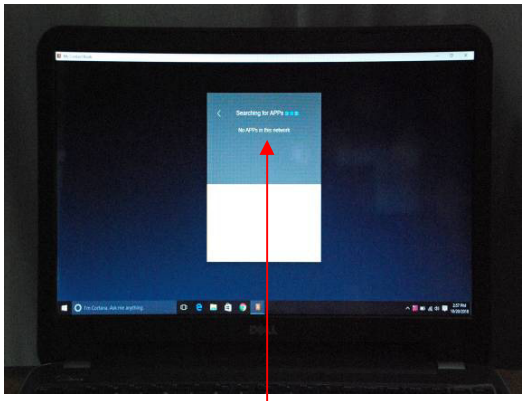


Figure 8. No applications available in the home network (if My Phone Dialer app not started on Smartphone)

Figure 9. My Phone Dialer app installed on an Android Smartphone

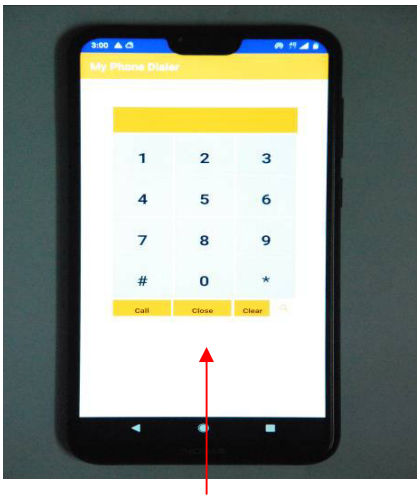


Figure 10. My Phone Dialer app running on an Android Smartphone

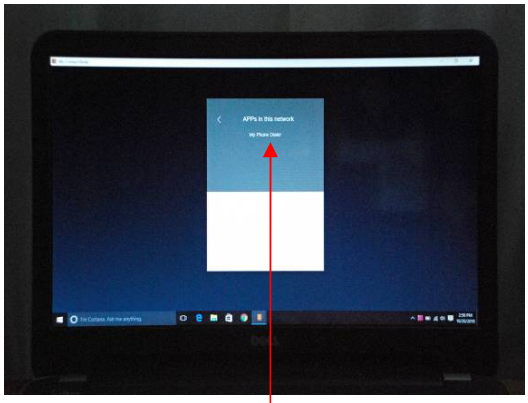


Figure 11. My Phone Dialer app is available in the home network (if My Phone Dialer app started on Android Smartphone)

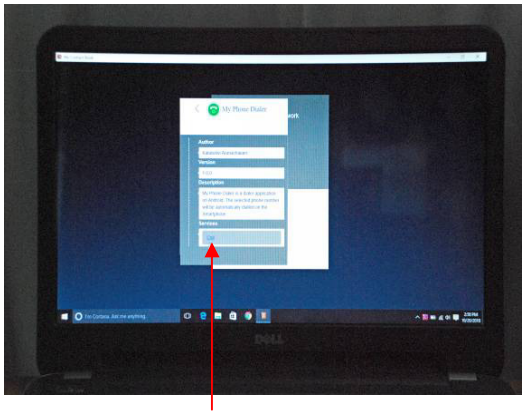


Figure 12. User views the details of My Phone Dialer app along with its services and selects the call service

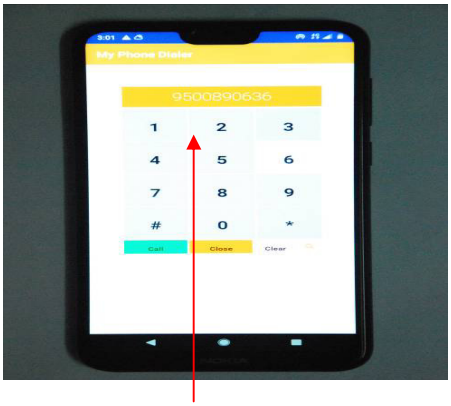


Figure 13. My Phone Dialer app receives the selected phone number from My Contact Book app

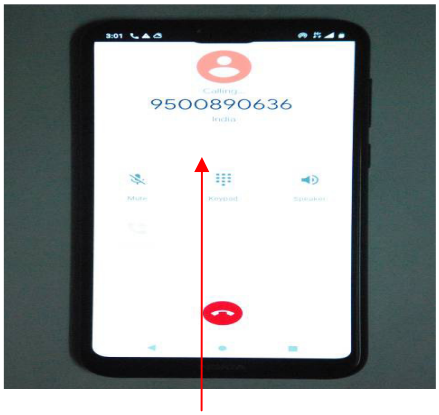


Figure 14. My Phone Dialer app automatically dials the phone number of the selected contact

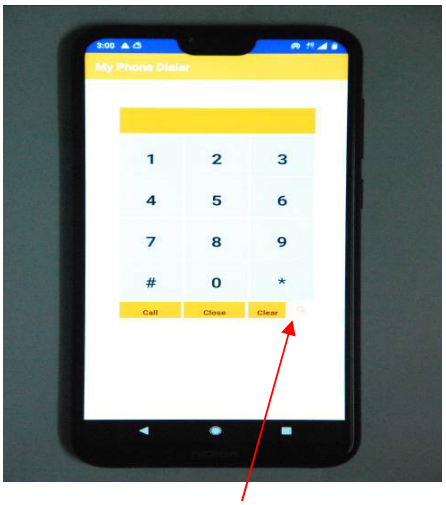


Figure 15. User searches for My Contact Book app in the home network

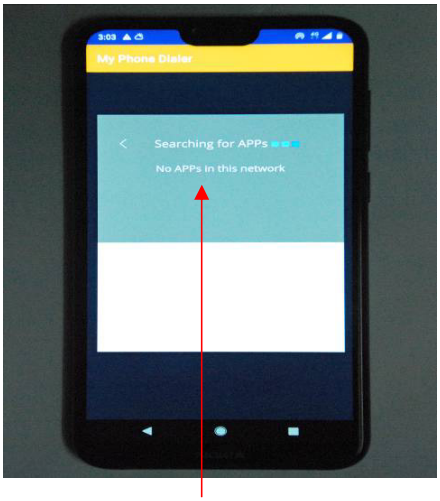


Figure 16. No applications available in the home network
(if My Contact Book app not started on Windows PC)

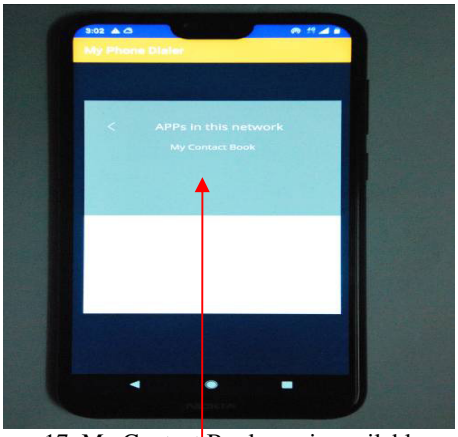


Figure 17. My Contact Book app is available in the home network
(if My Contact Book app started on Windows PC)

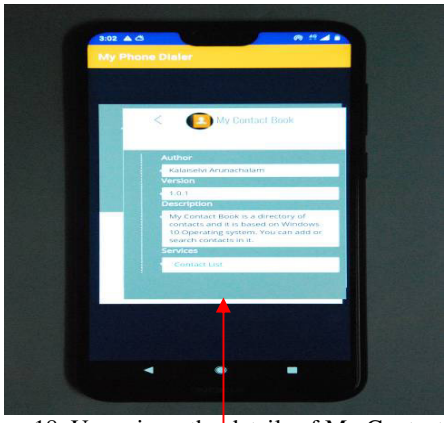


Figure 18. User views the details of My Contact Book app along with its services

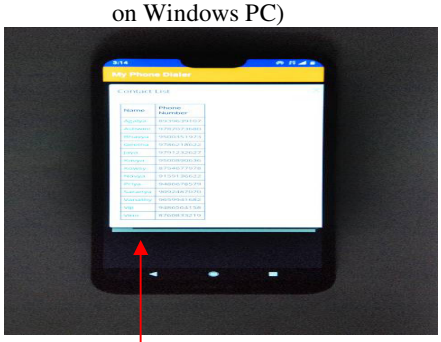


Figure 19. User views the contacts from My Contact Book app and selects a contact to make a call

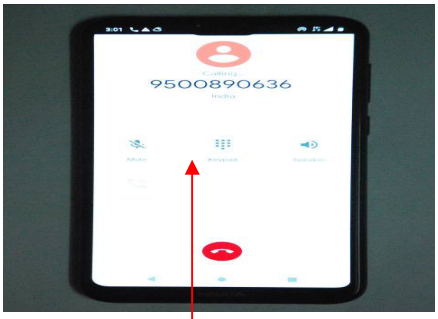


Figure 20. My Phone Dialer app dials the phone number of the selected contact

4.5. EVALUATION

The My Contact Book app is installed on a Notebook with Windows 10 operating system, 1.5 GHz Intel processor, 2GB RAM and 500 GB hard disk. The My Phone Dialer app is installed on a Nokia 6.1 Plus Smartphone with Android 9 Pie operating system, Qualcomm Snapdragon 636 processor, 4GB RAM and 64GB internal memory. Both applications were tested on these devices in a home network and both of them worked well as expected by discovering each of their presence, communicating with each other by accessing their services and by invoking action on each other in the home network.

Some of the application attributes like app name, app developer name, version, description and services are used by these applications in their app description as in [18]. These two applications were developed based on a sample scenario to demonstrate the inter-application communication in the home network. There are numerous scenarios available where by any kind of applications can discover, communicate and share their data with other applications in a home network.

5. CONCLUSION

The popularity of smart devices and the heterogeneity of applications used on these devices with multitude of data that serves the users around the world in their digital living environment. The data sharing between the applications across these heterogeneous devices without using any intermediary hardware or software would be beneficial to the users in a home network. These applications use various types of data which can be accessed and shared by the users through inter-application communication. Hence it is implemented in a home network with a prototype in which two different applications on two different device types and operating systems can discover, communicate and share data with each other. Following the implementation, the prototype is evaluated by testing it in a home network and it worked well as expected.

REFERENCES

[1] Kenny Fok, Jihyun Hwang, Eric Chi Chung Yip, Mikhail A. Lushin, (2012) "Providing secure inter-application communication for a mobile operating environment", United States Patent.

[2] Venu Ambekar, Robert Neff, John Zaleski, (2005) "Executable application interoperability and interface creation system", United States Patent.

[3] Samir Nigam, (2014) "Deploy anywhere framework for heterogeneous mobile application development", United States Patent.

[4] Sachin Desai, Qingqing Liu, Ronald Fischer, (2016) "Method and system for secured inter-application communication in mobile devices", United States Patent.

[5] Lloyd Leon Burch, Baha Masoud, (2018) "Audio proximity-based mobile device data sharing", United States Patent.

[6] Hongwei Guo, Long He, Fangming Li, Xiao Qiu Tang, Xi Ning Wang, Jing Zhang, (2018) "Data sharing between multiple applications running on a mobile device", United States Patent.

[7] Kyungjin Kim, Kiwon Lee, Sungil Cho, Jiyoung Hong, Sungeun Kim, (2013) "Data sharing between smart devices", United States Patent.

[8] Thomas Edward Wagner, Robert Elliott Whiteman, (2014) "Secure App-To-App Communication", United States Patent.

[9] Andrew Mark Earnshaw, Jianfeng Weng, (2017) "System and method for sharing data across multiple electronic devices", United States Patent.

[10] Stefan Poslad, (2009) Ubiquitous Computing: Smart Devices, Environments and Interactions, Queen Mary, University of London, United Kingdom.

[11] Google Play: https://play.google.com/store

[12] App Store (iOS): https://apple.com/appstore

[13] Microsoft Store: https://www.microsoft.com/store/apps

[14] BlackBerry World: http://appworld.blackberry.com/

[15] Number of apps in leading app stores 2018 | Statista: https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

[16] Kalaiselvi Arunachalam, Dr. Gopinath Ganapathy, (2015) "The Comparison of Inter-Application Communication Mechanisms in Mobile Operating Systems", International Journal of Computer Science and Mobile Applications (IJCSMA), Vol. 3. Issue. 10, pp. 48-58.

[17] Kalaiselvi Arunachalam, Dr. Gopinath Ganapathy, (2016) "Discovery and Identification of an Application for Inter-Application Communication on a Home Network Using UPnP", Journal of Computers (JCP), Vol. 11, No. 6, pp. 488-496.

[18] Kalaiselvi Arunachalam, Dr. Gopinath Ganapathy, (2017) "Extending UPnP for Application Interoperability in a Home Network", International Journal of Electrical and Computer Engineering (IJECE), Vol. 7, No. 4, pp. 2085-2093.

AUTHORS

Kalaiselvi Arunachalam received the B.Sc. degree in Physics from the University of Madras, India and M.C.A degree in Computer Applications from the Anna University, India. She is currently a Ph.D. scholar in the School of Computer Science Engineering and Applications, Bharathidasan University, India. Her research interests include Home Networking, Communication Software and Systems.

Dr. Gopinath Ganapathy received the B.Sc. degree in Computer Science from the Bharathidasan University, India, M.C.A degree in Computer Applications from the St. Joseph's College Autonomous, India and Ph.D from the Madurai Kamaraj University, India. He is currently the Registrar, Bharathidasan University, India. His research interests include Semantic Web, NLP, Ontology, and Text Mining.

ORDER PRESERVING STREAM PROCESSING IN FOG COMPUTING ARCHITECTURES

K. Vidyasankar

Department of Computer Science, Memorial University of Newfoundland,
St. John's, Newfoundland, Canada

ABSTRACT

A Fog Computing architecture consists of edge nodes that generate and possibly pre-process (sensor) data, fog nodes that do some processing quickly and do any actuations that may be needed, and cloud nodes that may perform further detailed analysis for long-term and archival purposes. Processing of a batch of input data is distributed into sub-computations which are executed at the different nodes of the architecture. In many applications, the computations are expected to preserve the order in which the batches arrive at the sources. In this paper, we discuss mechanisms for performing the computations at a node in correct order, by storing some batches temporarily and/or dropping some batches. The former option causes a delay in processing and the latter option affects Quality of Service (QoS). We bring out the trade-offsbetween processing delay and storage capabilities of the nodes, and also between QoS and the storage capabilities.

KEYWORDS

Fog computing, Order preserving computations, Quality of Service

1. INTRODUCTION

Internet of Things (IoT) is about making things smart in some functionality, and connecting and enabling them to perform complex tasks by themselves. A "thing" is any object of interest with some communication capability. IoT applications include Connected Vehicles, Smart Grid, Smart Cities, HealthCare and, in general, Wireless Sensors and Actuators Networks [1]. Billions of devices are expected to be made smart in the very near future. They will produce massive amounts of data, requiring enormous amount of computations. In cloud-based IoT environment, the computations are delegated to the cloud. The cloud is certainly scalable with respect to processing capability and storage. However, many applications require quick real time computations and local actuations, and the latency involved in communicating with the cloud is not tolerable. Further, sending huge amount of data to the cloud requires high network bandwidth and incurs considerable delay. In addition, in many applications, 24/7 connectivity to the cloud may not be available. To overcome these constraints, a *fog computing* architecture has been proposed recently [1, 2, 3, 4]. It consists of *edge* nodes that generate and possibly pre-process (sensor) data, *fog* nodes that do some processing quickly and enable any actuations that may be needed, and *cloud* nodes that may perform further, detailed analytics for long-term and archival purposes.

Fog computing typically involves continuous processing of stream data that are input to the edge devices. The data consist of tuples. They are processed in batches of tuples. Each processing instance at a node uses some input batches and produces an output batch which is sent to the

Dhinaharan Nagamalai et al. (Eds) : COMIT, AISCA - 2019
pp. 37 – 48, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.90104

parent of that node (except at the cloud level) for further processing. The computation to be done on a batch is decomposed into sub-computations to be executed at the different nodes in the fog architecture. Edge and fog nodes typically have limited storage, compute and network connectivity capabilities. Hence, the computations need to be distributed carefully among the processing nodes. Guaranteeing consistency of the executions is very important. Consistency issues arise for sub-computations at the individual nodes as well as the entire computations on individual batches and computations over sequences of input batches.

In this paper, we consider each sub-computation at a node as a transaction. We also assume serial executions of these transactions in each node. We relate consistency to serializability of these transactions at every node. In several applications, the computations on the sequence of batches are expected to preserve the order in which the batches arrive from the sources. This is the *consecutive serializability* requirement for the transactions. In some cases, the sub-computations at some nodes, especially at lower levels of the hierarchy, may not be required tofollow batch order, that is, the sequence can be *saga*[5], with the order being restored at higher levels. This helps also for scalability where the computations at a level can be distributed over multiple nodes and the results forwarded to a single node in the next higher level. Then the input batches in the higher level may not arrive according to the batch order. Unreliable network connectivity may also produce out-of-order message delivery. In this paper, we focus on achieving consecutive serializability at a node in the presence of out-of-order message delivery. We do this by storing some input batches temporarily and/or dropping some batches. The first option requires storage capacity and also causes delay in processing whereas the second option affects the accuracy of the continuous executions. This affects Quality of Service (QoS). We identify some QoS parameters that are relevant in this context. We discuss different execution options that offer trade-offs between QoS and storage capacities of the processing nodes.

We consider the simple case of inputs from a single source in Section 2. We consider individual executions of the input batches as well as their combined executions. We consider processing batches from multiple input sources in Section 3 and multiple heterogeneous input sources in Section 4. We discuss some related works in Section 5. We conclude in Section 6.

2. SINGLE INPUT SOURCE

We use the basic definitions given in Vidyasankar [6]. We consider a hierarchy (rooted tree) **V** of nodes v. It consists of n levels. In this section, we consider the simple case of a single input source. Then, the hierarchy is a simple path of length $n-1$. The node in the path in j^{th} level will be v_j. Here, v_n refers to the cloud, v_1 to the edge and the intermediate nodes to the fog. We assume that stream data is generated at level 0. The edge devices at level 1 themselves may generate some or all of this data. We separate the generation into another level for notational convenience. Each node $v_j$has *processing capability* $P_j$and *storage capacity* S_j, each expressed in appropriate units. The source input batches are numbered sequentially. We refer to the i^{th}batch as b_i. Each batch b_i is processed in one or more nodes. The computation for b_i is referred to as $C(b_i)$. We consider a decomposition of $C(b_i)$ into sub-computations as

$$C(b_i) = c_{1,i}+c_{2,i}+ \cdots +c_{n,i}.$$

Such decompositions will be based on the semantics of the applications and of the computations. He reeach $c_{j,i}$is to be executed at level j, in the given sequential order of the levels. The sum of the computations until level j is referred to as $C_{j,i}$(with capital C). That is,

$$C_{j,i}= c_{1,i}+ \cdots +c_{j,i}.$$

Then, $C_{n,i}= C(b_i)$.

As stated earlier, we assume in this paper that the individual $c_{j,i}$'s are executed atomically and serially in each level j. We denote the processing requirement and storage requirement for $c_{j,i}$as $p(c_{j,i})$ and $s(c_{j,i})$, respectively. Obviously, we must have $P_j \geq p(c_{j,i})$ and $S_j \geq s(c_{j,i})$. With each $c_{j,i}$, we associate an input batch $In(c_{j,i})$ and an output batch $Out(c_{j,i})$.

Several (devices in) nodes may have limited range of transmission. Nodes have to be placed such that dataflow from one level to the next is possible. To facilitate this, some nodes could be placed just to receive data from the lower level and send it to upper level. (This may involve storing some data temporarily.) We call these *relay* nodes. Sub-computation done in such a node will be *nil*. Output batch of this computation is the same as the input batch.

We discuss serializable executions of $C_{j,i}$'s. We define the following.

(1) **C** is the set of computations $c_{j,i}$'s for a given set of batches.
(2) $\prec B$ is the batch order.
(3) $\prec L$ is the level order.
(4) $\prec$ is $\prec B \cup \prec L$.
(5) A *history* **H** over (**C**, $\prec$) is a sequence of $c_{j,i}$'s in **C** obeying $\prec$.
(6) A history **H** is *globally serial* if it is a sequence of $C(b_i)$'s, that is, all the $c_{j,i}$'s for each I occur consecutively in **H**. It is *globally serializable* if it is equivalent to a globally serial history.

Some batches may be processed only partially. That is, $C(b_i)$ may only be $C_{k,i}(b_i)$, for some k, $k<n$. The above definition applies to such computations also.

2.1 INDIVIDUAL PROCESSING

We first consider processing of the batches individually at each level. Then, consecutive serializability of $C_{j,i}$'s, at each level j, is guaranteed if $c_{k,i}$'s are executed at each level k between 1 and j serially according to the batch order. (Recall that we are assuming atomic execution of each $c_{j,i}$.) In the following, we look at the ways of obtaining serial order effectively when output batches from one level may arrive at the next level out of order.

If $c_{j,i}$'s are not conflicting with each other, then an out-of-order execution is serializable. Then, inputs may be processed as they arrive and the corresponding outputs sent to the next level. This option is very favourable for horizontal scalability. Batches may be split and processed in multiple nodes in the same level provided the combined computations will constitute $c_{j,i}(b_i)$. However, an out-of-order execution together with an out-of-order message delivery from the current level to the next might amplify the extent of the out-of-order in the arrival of batches in the next level. (The *extent* of out-of-order can be characterized in many ways: (i) how late a batch arrives, that is, the number of batches with greater ids that come before this batch, (ii) how early a batch arrives, namely, the number of batches with smaller ids that come after this batch, (iii) the number of late or early arriving batches, (iv) averages over thedelay or too early arrival, etc.)

In the following, we consider the case where $c_{j,i}$'s are conflicting.

- Out-of-order inputs (messages) can be kept in a *pending set*, and the executions themselves can be done in correct order when the respective batches arrive. This involves waiting, causing delay in execution, and requires storage space for the pending set. Depending on the extent of the out-of-order, both the delay and the required amount of storage space will vary.

- Without pending set, executions can be done for batches arriving in increasing order of their ids, as they arrive, and late-arriving batches can be ignored (dropped). This implies that the

dropped batches are processed only partially, up to the previous level. No storage space is required here. In the example sequence (1,8,4,2,5,7,9,3), batches (with ids) 1, 8 and 9 will be processed and the remaining will be ignored.

- Without pending set, executions can be done for batches in the correct consecutive order of their ids. Out-of-order batches (those that arrive too early) can be ignored. In the above example sequence (1,8,4,2,5,7,9,3), batches 1,2 and 3 will be processed and the remaining ignored.

- A limited storage space can be kept for the pending set and early-arriving out-of-order messages that cannot be added to the pending set can be ignored. For example, with (1,8,4,2,5,7,9,3), if storage space is available only for three batches, after batches 8,4 and 5, batches 7 and 9 might be ignored. (Other options regarding which three batches to store can also be exercised.) Similarly, out-of-order messages arriving later than a certain amount of delay can be ignored.

We define *drop ratio* as the number of batches dropped compared to the total number of batches. We note that, in the above options, a trade-off exists between drop ratio and storage space, and between drop ratio and processing delay. Message loss is equivalent to dropping the message due to excessive delay.

If network connectivity is disrupted intermittently and hence output batches cannot be transmitted immediately after the executions, then the following options exist:

- When storage space is available, the options are the following. Here, two pending sets are used, one for storing input batches and the other for storing output batches.
  - Store output batches in the *output pending set* and send several of them together when connectivity is restored. Continue processing the input batches. (This option can be followed even when network connectivity is available, if transmitting several batches together will be cheaper than sending them one at a time.)
  - Stop processing until the output batch is sent, and store the incoming batches in the *input pending set.*
  - Store output batches in the output pending set until connectivity becomes available, and also store input batches in the input pending set until they can be processed. Continue processing. This option is suitable when input arrives from the lower level as a set of batches.
- When sufficient storage space, for output batches and/or pending sets, is not available, the options are the following.
  - Drop the output batch thus terminating the processing of the corresponding batch and continue processing the input batches.
  - Stop processing until the output batch is sent, and drop the batches that are incoming in the mean time, thus terminating their executions.

Here also, we observe a trade-off between drop ratio and storage space. Nodes in the hierarchy could be heterogeneous. Different nodes may follow different options. Out-of-order execution may be acceptable at some levels, and correct order required at certain levels. This amounts to $c_{j,i}$'s being non-conflicting at the former levels and conflicting in the latter ones. Then, the execution options may be chosen appropriately. We also note that allowing some out-of-order execution will reduce drop ratio. That is, there is a trade-off between the extent of the out-of-order and drop ratio.

2.2 COMBINING MULTIPLE BATCHES

At any level, several input batches may be combined and processed together. That is, the computation at v_j could be $c_j(b_{i:k})$, combining c_j for batches b_i to b_k. The combined output will be sent to v_{j+1}. An example is when the frequencies of executions at different levels are different. For instance, c_1 may be performed every 5 seconds and c_2 performed every 10 seconds. Then, the outputs of two executions of c_1 may be processed together in one execution of c_2. Another situation is when network connectivity is not always available to send data from one level to the next level and the output batches corresponding to several computations kept and sent together when connectivity becomes available.

In the following discussions, we use examples where three batches are combined.

2.2.1 NON-OVERLAPPING GROUPINGS.

(1) Grouping of consecutive batches:

Wait until all the relevant batches arrive and then process. Delay and storage space considerations discussed in the single batch processing case are applicable here also. In addition, we need to consider the following.

(a) Suppose batches 1, 2 and 3 are to be grouped, and 2 arrives very late (or does not arrive due to message loss). Then, we can drop that batch. Then, we can do one of the following:

- drop batches 1 and 3, that is, the entire group;
- do the computation for batch 1 alone (if the application semantics allows it) and combine batch 3 with 4 and 5; or
- combine batches 1, 3 and 4 if the application semantics allows grouping of a broken sequence of batches.

All these options relate to QoS differently: (i) absence or presence of broken sequences and, in the latter case, the number or percentage of broken sequences and (ii) fixed or variable size groupings.

(b) The batches for latter groupings may become available before those for earlier groupings (for example, (4,5,6) before (1,2,3)). If the computations are not conflicting, they can be processed in any order. Otherwise, they have to be processed in the correct batch order. If (4,5,6) grouping is processed first and then we find that batch 2 has to be dropped, the options are dropping 1 and 3 also or processing them either individually or by combining them.

(2) Grouping of non-consecutive batches:

As and when sufficient number of batches are available, the grouping can be done. The only storage space required will be for the batches waiting for the grouping.

2.2.2 OVERLAPPING GROUPINGS.

An example is (1,2,3), (2,3,4), (3,4,5), etc. After 1 and 2, suppose 5 arrives. Then wait for 3. When 3 arrives, combine (1,2,3). Then, wait for 4, etc. Whichever batches need to arrive, wait for them. Here, suppose 3 does not arrive for a long time and so it is dropped. Then, groupings (1,2,4), (2,4,5), etc. can be considered. Another possibility is dropping (1,2,3), (2,3,4) and (3,4,5), namely, all the originally intended groupings with 3. The choice would depend on the application

semantics of 'consecutive' batches. Here also, different options affect QoS differently. The delay and the storage space factors are the same as with non-overlapping groupings.

3. MULTIPLE HOMOGENEOUS INPUT SOURCES

In this section, we consider multiple input sources, all producing similar data that are to be processed the same way. The hierarchy is a tree. We consider a general height-balanced tree. (The discussion in the next section covers arbitrary trees.) We again separate the data generation part into level 0 and each source feeds to, that is, sends its output to a *distinct* node in level 1. Thus, each node in level 1 has one child.

We consider the case where, *at each level, each node performs the same computation*. (This restriction is also relaxed in the next section.) Each node in level 1 will process its source input and send its output to its parent. Each node in level j, for $1 < j < n$, will process the inputs from all its children and will send a single output batch, at the end of processing, to its parent.

We will first consider the case where all sources generate data synchronously. We refer to one such set of batches as a *batch-set*. We first consider synchronous processing of the batch-sets. That is, at each step, one batch arrives from each child and the set of these batches is processed. The batch-sets are indexed sequentially. A batch-set with index i is referred to as B_i. A computation at a node v_j in each level j combines the computations $c_j(x)$ of all the source input batches x in a batch-set that are input to the descendants of v_j in level 1. The computation required for B_i is $C(B_i)$, decomposed into $c_1(B_i) + c_2(B_i) + \cdots + c_n(B_i)$.

We now consider out-of-order message delivery from one level to another. We assume that the communication between any two nodes (a parent and a child) is independent of the communications between other pairs of such nodes. Therefore, the extent of the out-of-order will vary with respect to messages from different children. In the following, we illustrate the options with an example where the messages from only one child arrive out-of-order and messages from all other children arrive in correct order. We consider the example sequence (1,8,4,2,5,7,9,3) for messages from child x. Input batch from x with batch id k is denoted x_k.

- The executions are done in correct consecutive order when all the inputs for the corresponding batch-set have arrived. Until then, the incoming batches are kept in *separate* pending sets, one for each child. For the sequence (1,8,4,2,5,7,9,3), after processing B_1, the pending set for x will store inputs for batch-set ids 8 and 4 and pending sets for the other children will store 2 and 3 until x_2 arrives. Then, the computation can be done for B_2. After three further steps, the pending set for x will have (8,4,5,7,9) and other pending sets will have (3,4,5,6,7). On arrival of x_3, B_3 will be processed, followed by B_4 and B_5, waiting for x_6 for the processing of B_6, and so on. This involves waiting, causing delay in execution, and requires considerable storage space for the pending sets.

- We can reduce the size of the pending sets considerably as follows. Executions can be done in the correct consecutive order of the batch-sets with the batches arriving from children in the correct order, and *not waiting* for the batches of that batch-set from other children; when these batches arrive later, they are ignored. Out-of-order batches from other children with greater ids (those arriving too early) are stored in the pending sets, and used when their turns arrive. In our sequence (1,8,4,2,5,7,9,3), after B_1, batch-sets B_2 and B_3 will be processed without the inputs from child x. Batches 8 and 4 will be stored in the pending set for x. Then, B_4 will be processed with the newly arriving batches from other children and the one stored in the pending set for x, ignoring x_2. Batch-set B_5 will be

processed with batches from all children, and B_6 with inputs from all except x, storing 7 in the pending set of x, and so on.

  - This implies executions on partial batch-sets. This affects QoS relating to whether there are executions on partial batch-sets and, if so, a measure of the *density* of the partial sets, for example, how many batches, how well they represent various geographical regions, etc.
  - The execution can be subject to receiving batches from a minimum number of children, to make it meaningful. Otherwise, no execution may be done, resulting in dropping the entire batch-set in that level. This affects QoS differently: the drop ratio can be categorized as *batch drop ratio* and *batch-set drop ratio*.

- A variation in the above option is dropping the out-of-order inputs (those with greater ids, 8 in the above example), instead of storing in the pending set. That is, all out-of-order messages are dropped. Then, no pending sets are kept.

Combinations of the above options are possible, especially when the extent of the out-of-order is expected to be small. The first option of keeping the batches in the pending sets until all the inputs of the next batch-set arrive can be used for a while. At some stage, if the storage space becomes insufficient or the delay becomes too much, executions with partial batch-sets can be done. If the computations on different batch-sets are not conflicting, then the batch-sets can be processed soon after all their input batches are received. For example, in the sequence (1,8,4,2,5,7,9,3), B_4 can be processed without waiting for B_3 (in the case of not opting for executions on partial batch-sets). This will also reduce the number of entries in the pending sets.
We note that, as illustrated in the above example, if the inputs from even one child are out-of-order, the inputs from all other children have to be kept in the pending sets for correct, consecutive, order of execution.

Allowing for non-synchronous arrival of input batches (at any level, including the source level) and hence non-synchronous execution is straightforward. The batches from each input can be kept in the respective pending sets and when a batch-set is complete it can be processed. The processing could be in the correct order or any order. At some stage, an incomplete batch-set can either be dropped or processed as such.

If network connectivity is disrupted intermittently, the options discussed in Section 2 are applicable here also. We recall that the options are storing output batches, storing input batches, and dropping batches before or after the current computation. The requirement of storage space for pending input batches and/or output batches is inevitable. Less space will be needed for output batches due to (i) storage of one batch per computation in contrast to all input batches for that computation and (ii) computations such as aggregation producing outputs that are likely to be much smaller in size than any input or at least all inputs put together. Here also, nodes in the hierarchy could be heterogeneous and may follow different options.

Considerations for overlapping and non-overlapping groupings of several batch-sets are similar to those for grouping batches from a single source case. Several QoS parameters can be applied for the groups for different options. Some of them for non-overlapping groupings where out-of-order messages are ignored are:

- the number of complete batch-sets;
- the number of missing batch-sets;
- minimum number of batches in a processed batch-set; and
- average number of batches from a child.

For example, in a grouping of 5 batch-sets from 4 children, the quantities mentioned above for the sets of batches ((1,1,1,1),(2,2,-,2),(-,3,-,3),(5,-,5,5),(-,6,-,6)) will be 1 (for the first batch-set), 1 (for the fourth batch-set), 2 (with respect to the third batch-set) and 14/6 (with 6 batch-sets), respectively. (Here, "-" denotes messages arriving out-of-order messages and hence being dropped.) Different aggregations for several groups of batch-sets can also be considered.We note that grouping of individual out-of-order messages may result in reduced out-of-order among messages relevant for the entire group. For example, with grouping of three batch-sets from three children, for the batches arriving in the sequence (1,2,1), (2,1,3) and (3,3,2), there is no out-of-order messages with respect to the entire group.

4. HETEROGENEOUS INPUTS

We assume an arbitrary rooted tree for the fog infrastructure. Leaf nodes could be at different levels. In the following, we will assume that each source input is different and that computation performed at each node is different.

We will first consider the processing of a batch-set, consisting of one batch per source. In general, each input batch will be processed first individually and then together with other input batches (or the batches derived from them). For example, we consider a fog architecturewhere each of the three inputs x, y and z is processed individually first, then (derived batches from) x and y are processed together and then all the three are processed together. We refer to the computation done on a set S of batches as $C(S)$. Each of these computations is decomposed into sub-computations and then grouped into c_j's for execution at respective nodes. Let the corresponding sequences of computations be $C(x)$,$C(y)$,$C(z)$,$C(x,y)$ and $C(x,y,z)$. For each $C(S)$, the analysis as in the homogeneous case can be applied. We focus on the nodes where batches from different subtrees are combined. We refer to them as *merging nodes*. For simple exposition, we will take a single sub-computation for each set S, namely, $c_1(x)$,$c_1(y)$,$c_1(z)$,$c_2(x,y)$ and $c_3(x,y,z)$.

First, we consider synchronous arrival and synchronous execution of batch-sets B_i consisting of $\{x_i,y_i,z_i\}$.At merging nodes, we assume that if input batches from one or more children are not available, then the computation cannot be done. With out-of-order batch arrival, the options are the following:

- Store the batches in the pending sets and process a batch-set when it is complete, that is, when all the input batches corresponding to that batch-set have arrived.
- At any (synchronous) step, if all the input batches in the expected batch-set are not available, ignore the batch-set.

The options when network connectivity is disrupted are the same as in the homogeneous inputs case. We note that computations on the dropped batch-sets will not be done in any ancestors. This was called *ancestral-abort* in [7].

We now consider asynchronous arrival of the individual batches.For simple illustration, we consider the execution of $c_{2,i}(x,y)$, where only the batches (derived from) x and y are combined.We refer to the batches as x-batch and y-batch for convenience. The frequencies of generation of x- and y-batches may be different. We assume that the computation is triggered each time a new x- or y- or both batches arrive. In the first case, the most recent y-batch is used, in the second case, the most recent x-batch and in the last case both new batches are used for the computations. To be able to identify consistent pairing of the batches, we assume a *global* time stamping of the batches. We assume integer counter values as timestamps and index the batches with these timestamps. An example sequence of arrival of the batches, in correct order, is given in

Table 1. Here, for example, y_3 is paired with x_1, and also with x_4.We note that the indices of x-batches (similarly, y-batches) may not be continuous.

Table 1: Time stamped Sequence

x_1	$y_1y_2y_3$
$x_4x_5x_6$	y_6
...	...

Table 2: Indexed Time stamped Sequence

$x_{1,1}x_{1,2}x_{1,3}x_{2,4}x_{3,5}x_{4,6}$ ...	$y_{1,1}y_{2,2}y_{3,3}y_{3,4}y_{3,5}y_{4,6}$ ...

Now, we consider out-of-order message delivery. Then,x-batches and y-batches may arrive out of order at the merging node. We can store the batches in the respective pending sets, wait for a while for late-arriving batches and then order the batches correctly and pair them. For example, at some stage, if we assume that all batches with timestamps less than or equal to 6 have arrived, then the sequence shown in Table 1 can be formed and used for pairing. Batches arriving very late, very much out-of-order, can be ignored. Suppose, for instance, that y_3 has not arrived yet when we do the pairing. Then, we will end up pairing x_4 with y_2 and also x_5 with y_2. This causes inconsistency, in addition to not being able to use y_3, and QoS is affected.

Suppose that after y_1, y_6 arrives, perhaps after some time. Then, we will not know whether there are some y_i's in between. Suppose, with each y_j, the batch-id of the previous y-batch is sent. Then, on the arrival ofy_6, we would know about the existence ofy_3. However, until y_3 arrives, we will not know the existence of y_2. Thus, some mechanism can be implemented to indicate possible late arrivals of at least some batches.

To avoid the inconsistencies mentioned above, batches can be indexed with both batch number (independent of timestamp) and global timestamp as in Table 2. The batches with the same timestamp can be combined. We note that this is as in the case of synchronous execution of batch-sets. The timestamp synchronizes the batch-sets.

5. RELATED WORKS

Consistencies of continuous executions have been discussed widely in the literature in the context of stream processing. The sub computations are treated as transactions in Conway [8], Meehan et al. [9] and Botan et al. [10]. Serializability of the entire computation on a batch, treated as a composite transaction, is discussed in Gürgen [11] and Oyamada et al. [12]. Serializability of continuous queries is discussed in Vidyasankar [13]. Distributing computations in fog architectures has been described in Andrade et al. [14], Mortazavi et al. [15] andVidyasankar [6]. The property that computations at some levels are non-conflicting and hence they need not be order preserving has been used in Transactional Topologies [16]. Order preserving computations have been discussed in stream processing in Li et al. [17] and Shen et al. [18], and for Big Data Streams in Xhafa et al. [19].

6. DISCUSSION AND CONCLUSION

In fog architectures, stream inputs are processed in several stages at nodes in different levels. In a hierarchical structure, at each node, the computation is over the input batches that arrive from the children, producing an output batch which is sent to the parent. In this paper, we have addressed several issues relating to obtaining order preserving executions when messages do not arrive in correct order. We have discussed mechanisms for performing computations in correct order, by storing some batches temporarily and/or dropping some batches. The former option causes a delay in processing and the latter option affects QoS. We have brought out some trade-offs between processing delay and storage capabilities of the nodes, and also between QoS and the storage capabilities. Here, only transient storage of batches is considered, not persistent store for the results of the computation.

We have identified several QoS parameters that are relevant in this context. All of them deal with the non-inclusion of a batch in a computation. This can be identified at the node where the computation is done and its effect on the appropriate QoS parameter can be used at that node and also transmitted to the parent as part of the *context* associated with the output batch. The context may include batches explicitly with ids or just implicitly, for example, that a batch has been dropped. For doing this, batches need to be indexed. A natural place for indexing is the source level. However, in many applications, sources connect to *gateways* that filter the source input batches and pre-process them. The batches that are dropped in that level may not be relevant for the computations. Hence the (output) batches at the gateway level may be indexed. The gateways are expected to have the contextual information such as id, location, and other deployment details of the sources (sensors). Hence, indexing at that level may be more comprehensive than at the source level.

When the batches are processed individually throughout the entire hierarchy, the initial indexing may be adequate. However, when several batches are combined and processed together at some level, new index could be given to the output batch. (One example is assigning the largest batch index of that group, when the batches arriving at the next level need not have consecutive indexes.) That is, depending on the context that is attached to the output, separate independent indexing can be used at different levels, rather than carrying the initial index throughout.

Several non-hierarchical fog infrastructures have been proposed in the literature, for example, clustered, vehicular and smart-phone in [3]. These can be modeled by extending a hierarchy by replacing single nodes with clusters of nodes where the nodes within a cluster can communicate with each other in peer-to-peer fashion. Several sub-computations may be assigned to a cluster. They will be executed by one or more nodes in the cluster based on their processing and storage capabilities. Message transmissions among the nodes in a cluster may not have delay, loss and out-of-order properties. Even if they do, a cluster on the whole may have adequate capacity to store input and/or output batches to do the computations in correct order. Thus, our considerations in this paper need to be applied to inter-cluster communications only.

ACKNOWLEDGEMENTS

This research is supported in part by the Natural Sciences and Engineering Research Council of Canada Discovery Grant 3182.

REFERENCES

[1] F. Bonomi, R. Milito, J. Zhu & S. Addepalli (2012)"Fog computing and its role in the internet of things", *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, pp 13–16, New York, NY, USA, ACM.

[2] F. Bonomi, R. Milito, P. Natarajan & J. Zhu (2014) "Fog computing: A platform for internet of things and analytics", In N. Bessis and C. Dobre, editors, *Big Data and Internet of Things: A Roadmap for Smart Environments*, pp169–186, Springer International Publishing, Cham.

[3] C. Chang, S. N. Srirama& R. Buyya (2017)"Indie fog: An efficient fog-computing infrastructure for the internet of things",*Computer*, Vol. 50, No. 9, pp 92–98.

[4] A. V. Dastjerdi& R. Buyya(2016)"Fog computing: Helping the internet of things realize its potential",*Computer*, Vol. 49, No. 8, pp 112–116.

[5] K. Vidyasankar (1991)"Unified theory of database serializability", *FundamentaInformatica*, Vol. 1, No. 2, pp 145-153.

[6] K. Vidyasankar (2018a)"Distributing computations in fog architectures", *TOPIC'18 Proceedings*. Association for Computing Machinery.

[7] K. Vidyasankar (2018b)"Atomicity of executions in fog computing architectures",*Proceedings of the Twenty Seventh International Conference on Software Engineering and Data Engineering (SEDE-18)*.

[8] N. Conway (2008)"Transactions and data stream processing", *Online Publication*, pages 1–28. http://neilconway.org/docs/stream_txn.pdf.

[9] J. Meehan, N. Tatbul, S. Zdonik, C. Aslantas, U. Cetintemel, J. Du, T. Kraska, S. Madden, D. Maier, A. Pavlo, M. Stonebraker, K. Tufte, & H. Wang (2015) " S-store: Streaming meets transaction processing",*Proc. VLDB Endow.*, Vol. 8, No. 13, pp 2134–2145.

[10] I. Botan, P. M. Fischer, D. Kossmann, & N. Tatbul (2012)"Transactional stream processing", *Proceedings EDBT*, ACM Press.

[11] L. Gürgen, C. Roncancio, S. Labbé& V. Olive (2006)"Transactional issues in sensor data management", *Proceedings of the 3rd International Workshop on Data Management for Sensor Networks (DMSN'06), Seoul, South Korea*, pp 27–32.

[12] M. Oyamada, H. Kawashima, & H. Kitagawa (2013)"Continuous query processing with concurrency control: Reading updatable resources consistently", *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pp 788–794, New York, NY, USA, ACM.

[13] K. Vidyasankar (2017) "On continuous queries in stream processing", *The 8th International Conference on Ambient Systems, Networks and Technologies (ANT-2017), Procedia Computer Science*, pp 640–647. Elsevier.

[14] L. Andrade, M. Serrano& C. Prazeres (2018)"The data interplay for the fog of things: A transition to edge computing with IoT",*Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, IEEE Xplore.

[15] S. H. Mortazavi, M. Salehe, C. S. Gomes, C. Phillips & E. de Lara (2017)"Cloudpath: A multi-tier cloud computing framework", *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, pp 20:1–20:13, New York, NY, USA, ACM.

[16] storm.apache.org/releases/1.0.6/Transactional-topologies.html.

[17] Jin Li , Kristin Tufte, VladislavShkapenyuk, VassilisPapadimos, Theodore Johnson & David Maier (2008) "Out-of-Order Processing: A new Architecture for high-performance stream systems", PVLDB '08, pp 274-288, VLDB Endowment.

[18] Zhitao Shen, Vikram Kumaran, Michael J. Franklin, Sailesh Krishnamurthy, Amit Bhat, Madhu Kumar, Robert Lerche& Kim Macpherson (2015) "CSA: Streaming engine for internet of things", *Data Engineering bulletin*, Vol. 38, No. 4, pp 39-50, IEEE Computer Society.

[19] F. Xhafa, V. Naranjo, L. Barolli& M. Takizawa (2015)"On streaming consistency of big data stream processing in heterogeneous clusters", *Proceedings of the 18th International Conference on Network-Based Information Systems*. IEEE Xplore.

MAGNETIC ANOMALIES DUE TO 2-D CYLINDRICAL STRUCTURES - AN ARTIFICIAL NEURAL NETWORK BASED INVERSION

Bhagwan Das Mamidala[1] and Sundararajan Narasimman[2]

[1]Department of Mathematics, Osmania University, Hyderabad-500 007, India
[2]Department of Earth Science, Sultan Qaboos University, Muscat, Oman

ABSTRACT

Application of Artificial Neural Network Committee Machine (ANNCM) for the inversion of magnetic anomalies caused by a long-2D horizontal circular cylinder is presented. Although, the subsurface targets are of arbitrary shape, they are assumed to be regular geometrical shape for convenience of mathematical analysis. ANNCM inversion extract the parameters of the causative subsurface targets include depth to the centre of the cylinder (Z), the inclination of magnetic vector(Θ) and the constant term (A) comprising the radius(R) and the intensity of the magnetic field (I). The method of inversion is demonstrated over a theoretical model with and without random noise in order to study the effect of noise on the technique and then extended to real field data. It is noted that the method under discussion ensures fairly accurate results even in the presence of noise. ANNCM analysis of vertical magnetic anomaly near Karimnagar, Telangana, India, has shown satisfactory results in comparison with other inversion techniques that are in vogue.

KEYWORDS

Magnetic anomaly, Artificial Neural Network, Committee machine, Levenberg – Marquardt algorithm, Hilbert transform, modified Hilbert transform.

1. INTRODUCTION

In quantitative interpretation, the gravity and magnetic anomalies over a mineralized zone or geological structure can be approximated to simple geometrical shapes. Quantitative interpretation of the magnetic and gravity anomalies due to anticlines and synclines is accomplished by approximating them to two-dimensional, long horizontal circular cylinder. Linear concentrations of the mineral magnetite in a mineralized zone may be approximated some times to a horizontal cylinder. There are several methods of analyzing magnetic anomalies due to cylindrical structure. Parker Gay (1965) presented a set of master curves for the interpretation of the magnetic anomalies due to cylindrical bodies [24]. Rao et al. (1973) have developed direct methods for carrying out such interpretations [27]. Murthy and Mishra (1980) have proposed spectral approaches [22].

Mohan et al. (1990) used the Mellin transform in interpreting magnetic anomalies due to some two dimensional bodies [20]. Sundararajan et al. (1985, 1989) interpreted the magnetic anomalies of various components due to thin infinite dyke and spherical source by using Hilbert transform ([33], [34]). Srinivas (1998) used the modified Hilbert transform to interpret magnetic anomalies

Dhinaharan Nagamalai et al. (Eds) : COMIT, AISCA - 2019
pp. 49 – 62, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.90105

caused by 2-D horizontal circular cylindrical structures [31]. During 1999-2013, different methods Wavelet transform ([21], [15]), Displacement of the maximum and minimum by upward continuation [8], Euler deconvolution [11], Fraser filter [4], Hartley transform [17] and Direct analytic signal [5] were used for inversion of magnetic data.

TDX is a normalized version of the horizontal derivative filter and can recognize the edges of the shallow and deep bodies simultaneously. This filter is commonly used in the edge detection of potential field data. Recently, Alamdar et al. (2015) used combination of this balanced edge detection filter and Euler deconvolution to real magnetic data from Soork iron ore mine in Iran to estimate source location [1]. In the recent years, soft computing tools like Artificial neural network (ANN) , Fuzzy logic, Genetic algorithm gained great importance in geophysical data inversion ([16], [32], [29], [13], [9], [25], [2]).

A committee machine consists of a group of intelligent systems named experts (ANN) and a combiner which combines the outputs of each expert [7]. Its advantages are more accuracy in prediction, speed learning and better generalization. If the combination of experts in committee machine were replaced by a single neural network, one would have a network with a correspondingly large number of adjustable weight parameters. The training time for such a large network is likely to be longer than for the case of a set of experts trained in parallel. Moreover, the risk of over fitting the data increases when the number of adjustable weight parameters is large compared to size of the set of the training data.

In this paper, the analysis of vertical magnetic anomalies due to a 2-D horizontal circular cylinder is carried out using ANN-based committee machine. The method is illustrated with the study of theoretical model and validity of procedure is tested with the addition of random noise to the source data. Further, the technique is exemplified with magnetic anomaly over a narrow band of quartz magnetic near Karimnagar, Telangana, India [31]. Both the theoretical as well as field data yield reasonably good results and are compared with other methods that are in vogue.

2. ARTIFICIAL NEURAL NETWORKS

An artificial neural network consists of massively parallel interconnection of large number of neurons. It learns incrementally from environment to capture essential linear and nonlinear trends in complex data. On this basis it provides reliable predictions for new situations containing even noisy and partial information. ANN has at least two physical components, namely the processing elements and the connections between them. The processing elements are called neurons and connection between two neurons is called a link. Every link has a weight parameter associated with it. A neuron (j) computes a single output (a_j) from multiple inputs $(x_1, x_2, \ldots, x_{S_0})$ by forming linear combination according to its input weights $(w_{j1}, w_{j2}, \ldots\ldots, w_{jS_0}, b_j)$ and then possibly putting the output through some activation function ($f(.)$) and is shown in Figure (1) ([19], [28], [7]), where S_0 is the number of inputs. Activation functions such as sigmoid are commonly used since they are nonlinear and continuously differentiable ([10], [30]).

Multi-layer perceptron (MLP) is a feed forward artificial neural network with one or more layers between input and output layers. Figure (2) shows a two-layer feed forward network. The net input to a neuron j in layer $k+1$ is given by [6]:

$$n_j^{k+1} = \sum_{i=1}^{S_k} w_{ji}^{k+1} a_i^k + b_j^{k+1} \qquad \ldots (1)$$

The output of neuron j will be

$$a_j^{k+1} = f_j^{k+1}(n_j^{k+1}) \quad \text{where} \quad k = 0,1, \qquad \ldots(2)$$

where w_{ji}^{k+1} represents the weight associated with the i'th input to neuron j in layer $k+1$, b_j^{k+1} is a bias to neuron j and S_{k+1} is the number of neurons in the layer $k+1$.One may observe that if $\underline{x} = [\, a_1^0 \;\; a_2^0 \;\; \ldots \;\; \ldots \;\; a_{S_0}^0\,]^T$ is presented to the network and $AN(\underline{x})$ is the output of MLP, then

$$AN(\underline{x}) = [a_1^2 \;\; a_2^2 \;\; \ldots \;\; a_{S_2}^2]^T \qquad \ldots (3)$$

where a_j^2's are given by Eq. (2).

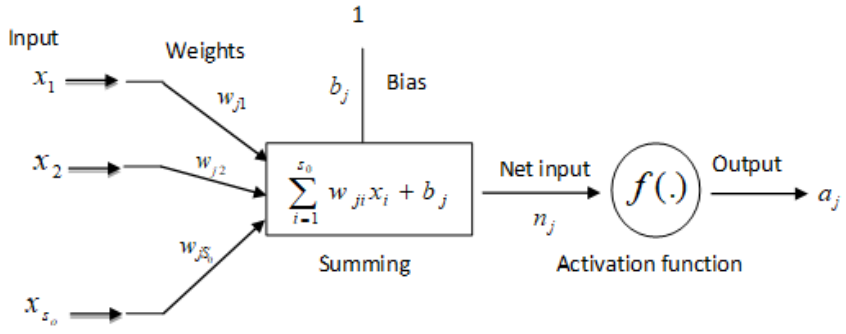


Figure 1 A model of an artificial neuron

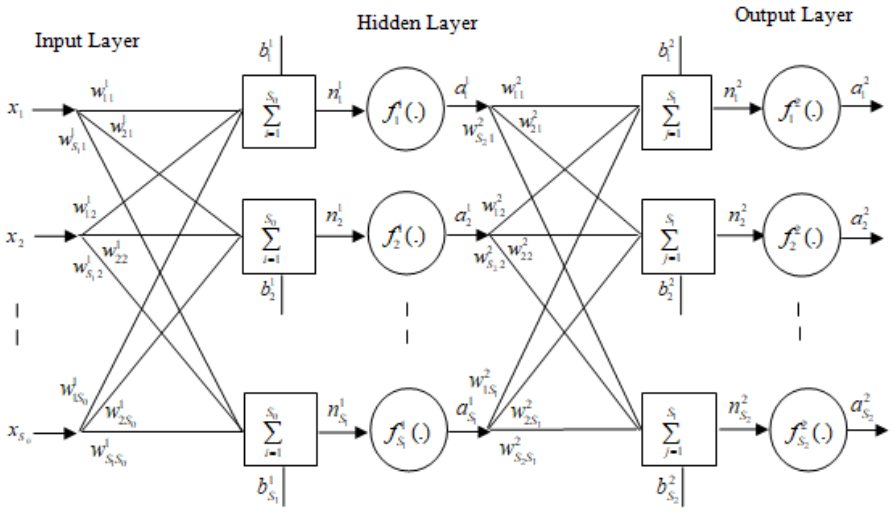


Figure 2 A two-Layer feed forward network (Multilayer perceptron)

MLP learns the problem behavior through a process called training and it would be taught with measured/simulated samples from a training set say $T = \{(\underline{x}_1,\underline{t}_1),(\underline{x}_2,\underline{t}_2),\ldots,(\underline{x}_P,\underline{t}_P)\}$. The performance (*Perf*) of MLP is calculated using the following error function:

$$E(\vec{w}) = \frac{1}{2}\sum_{p=1}^{P}\left(t_p - AN(\underline{x}_p)\right)^T\left(t_p - AN(\underline{x}_p)\right) = \frac{1}{2}\sum_{p=1}^{P} e_p^T e_p \quad \ldots (4)$$

where $AN(\underline{x}_p)$ is the output of network, $\vec{w}$ is the weight vector containing all the weights of the network, t_p - target, e_p - error and P - total number of training samples. The goal of the training is to find the weights that will impact the output from MLP to match the targets as closely as possible. If the outputs of MLP come as close as possible to match the targets for all the samples, then performance function $E(\underline{w})$ of network is minimized. Levenberg-Marquardt back-propagation algorithm ([14], [18], [6]) is one of the numerical optimization techniques that minimizes $E(\underline{w})$. It is fast with stable convergence. Levenberg-Marquardt algorithm (LMA) [3] is given by:

$$\vec{w}(n+1) = \vec{w}(n) - \left(J(\vec{w})^T(n)J(\vec{w})(n) + \mu I\right)^{-1} J(\vec{w})^T(n)e(\vec{w})(n) \quad \ldots (5)$$

where $e(\vec{w}) = [e_1(\vec{w})\ \ e_2(\vec{w})\ \ldots\ \ldots\ e_P(\vec{w})]^T$ is the error vector comprising the errors for all the training samples, $J(\vec{w})(n)$ is a Jacobian matrix, n is an iteration number and μ is a damping parameter. When μ is large, the method takes a small step in the gradient direction. As the method nears a solution, μ is chosen to be small and the method converges quickly via the Gauss Newton method. The flowchart of implementation of the LM algorithm is shown in Figure (3).

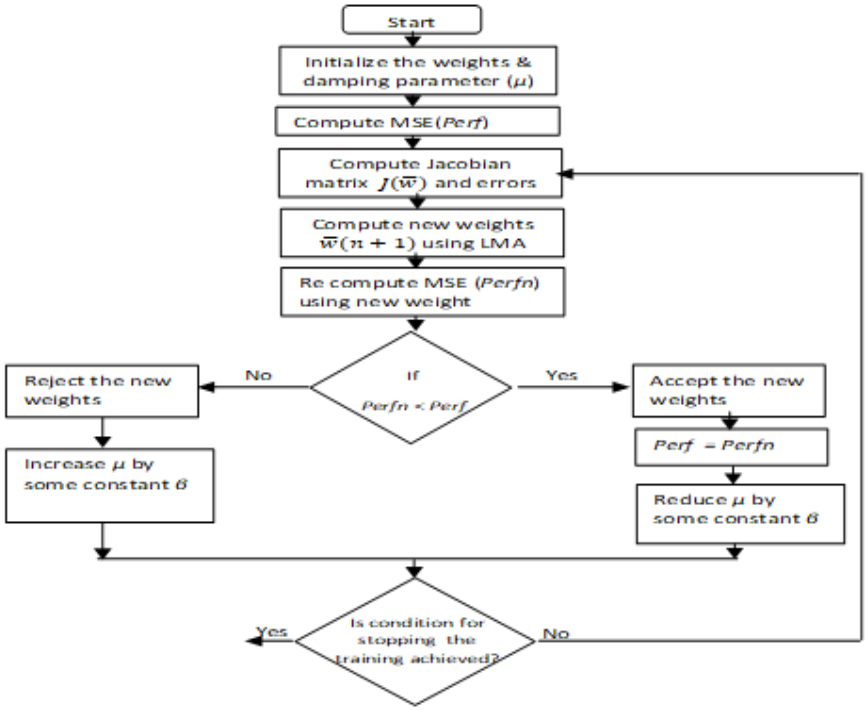


Figure 3 Flow-chart of implementation of the LM algorithm

A method for calculation of $\vec{w}(n+1)$ using Eq. (5) requires both forward and backward calculations. First, feed forward calculations which are made to determine the error at the output layer. The elements of the Jacobian matrix are then obtained by propagating this error back through the network which can be computed by a simple modification to the standard back

propagation algorithm [6]. The back propagation process has to be repeated for every output separately in order to obtain consecutive rows of the Jacobian matrix.

2.2 ARTIFICIAL NEURAL NETWORK COMMITTEE MACHINE (ANNCM)

Committee machines with static structure, the outputs of several predictors (expert) are combined by a mechanism that does not involve the input signal with *ensemble* and *boosting* methods. Figure (4) shows a number of differently trained neural networks (i.e., experts), which share a common input and whose individual outputs are combined using rules such as averaging, voting etc., to produce an overall output. Such a technique is referred to as an *ensemble averaging* method. This method is most popular [23]. Ensemble averaging creates a group of networks (experts); each with low bias and high variance, then combines them to a new network with low bias and low variance. Further the idea behind such network is to fuse knowledge acquired by experts in order to arrive at an overall decision that is superior to that of any of the individual experts ([12], [23], [7]).

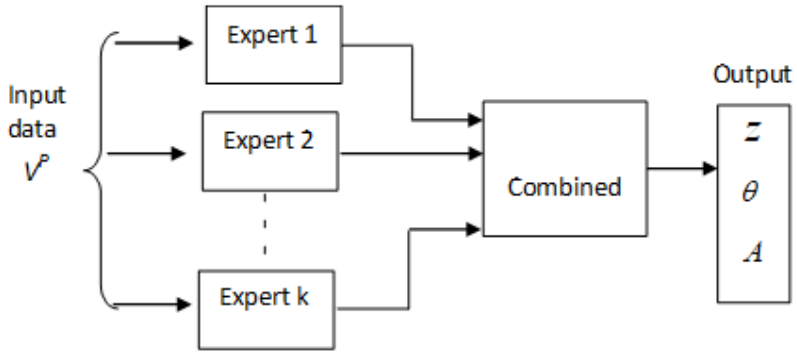


Figure 4 ANN Committee machine

3. MAGNETIC EFFECT DUE TO A 2-D HORIZONTAL CIRCULAR CYLINDER

The vertical magnetic effect $V(x)$ due to a 2-D horizontal circular cylinder extending infinitely along the Y-axis with its normal section parallel to the X-Z plane (Figure (5)) at a point $'x'$ is given by [36]:

$$V(x) = A\left[\frac{(z^2 - x^2)\sin\theta - 2xz\cos\theta}{(x^2 + z^2)^2}\right] \quad \ldots (6)$$

where,

z- is the depth to the centre of the cylinder,

θ - is the inclination of magnetic vector,

A-is the constant term comprising the radius (R) and the intensity of the magnetic field (I) and is given as $A = 2\pi R^2 I$.

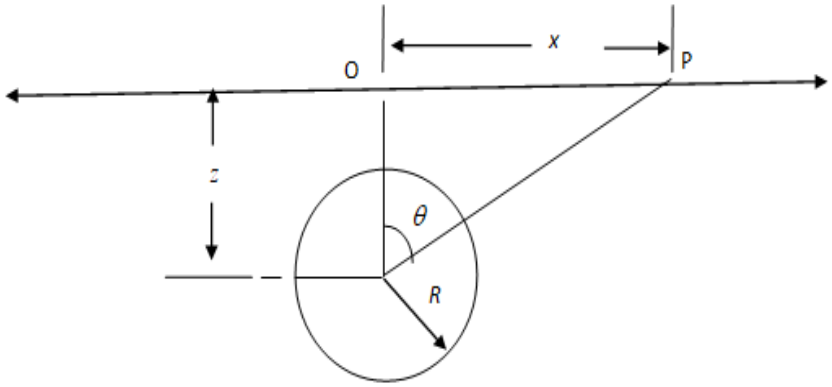


Figure 5 Geometry of the 2D Horizontal Circular Cylinder

The inversion of magnetic effect due to a 2-D horizontal circular cylinder is achieved by ANNCM which consists of phase-I and phase-II and is discussed in the following subsections. In phase-I coarse values of parameters are obtained whereas in phase-II fine values of parameters are obtained. We call, phase-I and phase-II as coarse and fine application. Flow chart of phase-I and phase-II is shown in Figure (6).

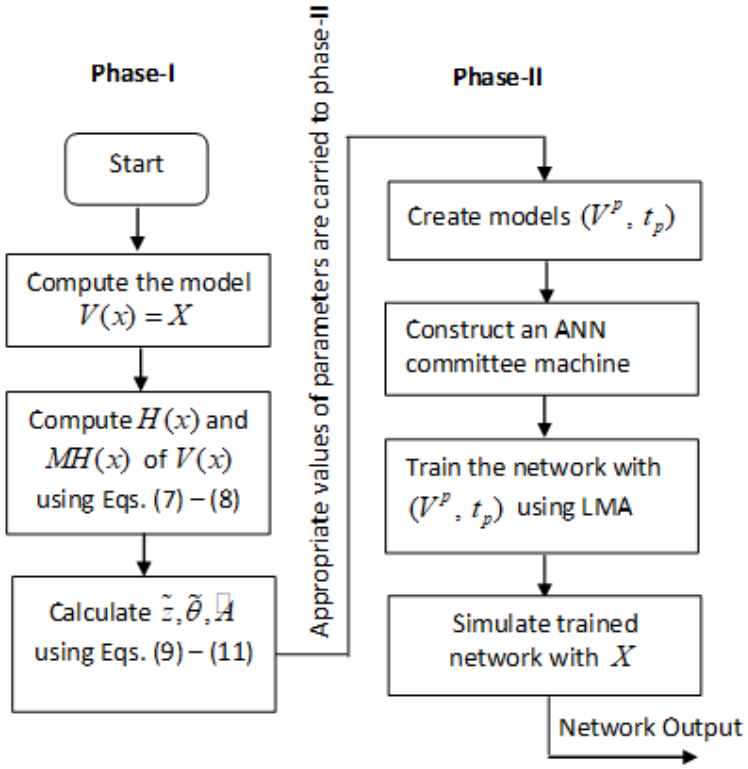


Figure 6 Flow-chart of implementation of the Phase-I and Phase-II

3.1 PHASE-I

In phase-I, Hilbert and modified Hilbert transform are used for the analysis of vertical magnetic anomaly $(V(x) = X)$ generated with the model parameters (z_o, θ_o, A_o). The Hilbert transform $H(x)$ and the modified Hilbert transform $MH(x)$ of the vertical magnetic anomaly $V(x)$ due to an inclined sheet are computed by [31]:

$$H(x) = -A\left[\frac{(z^2 - x^2)\cos\theta + 2xz\sin\theta}{(x^2 + z^2)^2}\right] \quad \ldots (7)$$

$$MH(x) = -A\left[\frac{(z^2 - x^2)\cos\theta - 2xz\sin\theta}{(x^2 + z^2)^2}\right] \quad \ldots (8)$$

Sundararajan and Srinivas [35] reported in literature that the Hilbert transform and its modified version intersect exactly over the origin (centre of the subsurface target). From the equations of vertical magnetic anomaly $V(x)$ and the modified Hilbert transform $MH(x)$, the depth to the top of the sheet $(\tilde{z})$, the inclination $(\tilde{\theta})$ and the constant term $(\tilde{A})$ are given as:

$$\tilde{z} = -\frac{x_1 + x_2}{2} \quad \ldots (9)$$

$$\tilde{\theta} = \tan^{-1}\left[\frac{2zxMH(x) - (z^2 - x^2)V(x)}{(z^2 - x^2)MH(x) - 2zxV(x)}\right] \quad \ldots (10)$$

$$\tilde{A} = z^2\sqrt{V(0)^2 + MH(0)^2} \quad \ldots (11)$$

where x_1 and x_2 are the abscissa of the points of intersection of $V(x)$ and $MH(x)$.

The appropriate values $\tilde{z}, \tilde{\theta}, \tilde{A}$ of parameters obtained by equations (9) – (11) will be carried to phase-II in order to increase their accuracy.

3.2 PHASE-II

Phase-II can be implemented in stepwise as hereunder.

Step -1: Create models to train an ANN in the following way.

Let $Z = \{\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_{m_1}\}$, $I = \{\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_{m_2}\}$ and $C = \{\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_{m_3}\}$, and be three sets of parameters that are selected in a small neighborhood of $\tilde{z}$, $\tilde{\theta}$ and $\tilde{A}$ respectively. Then the number of models of type $(\tilde{z}_i, \tilde{\theta}_j, \tilde{A}_k)$ is $m_1 \times m_2 \times m_3$, where $\tilde{z}_i \in Z$, $\tilde{\theta}_j \in I$ and $\tilde{A}_k \in C$.

For simplicity, rename each $(\tilde{z}_i, \tilde{\theta}_j, \tilde{A}_k)$ as (z_p, θ_p, A_p) for $p = 1, 2, \ldots\ldots, P$, where $P = m_1 \times m_2 \times m_3$. Let $V^p(x_j)$ be vertical magnetic effect due to a 2-D horizontal circular

cylinder generated by Eq. (6) at a point x_j with the model parameters (z_p, θ_p, A_p) where $1 \le p \le P, 1 \le j \le S_0$. Let $V^p = (V^p(x_1),...,V^p(x_j),...,V^p(x_{S_0}))$, $t_p = (z_p, \theta_p, A_p)$ and

$$T = \{(V^1, t_1), (V^2, t_2), ..., (V^P, t_P)\} \quad \ldots (12)$$

Realization of an inversion estimation of parameters of the anomaly (X) is achieved by training an ANNCM with the models $(V^p,\ t_p)$, where $(V^p,\ t_p) \in T$, $(p = 1, 2, ..., P)$.

Step-2: In this step, first design an artificial neural network committee machine with suitable number of experts (MLPs) and in turn each will be trained in batch mode with the models $(V^p,\ t_p), (p = 1, 2, ..., P)$ using LM algorithm.

4. THEORETICAL MODELS

The vertical magnetic anomaly $V(x)$ due to a 2D horizontal circular cylinder of theoretical model-I is generated using Eq. (6) with input parameters $(z = 12, \theta = 40$ and $A = 800)$ consisting of 51 samples with 2 units as sampling. The appropriate values of parameters obtained in phase-I are: $\tilde{z} = 8.96$, $\tilde{\theta} = 36.24^{\circ}$ and $\tilde{A} = 485.26$. The range of parameters and number of steps that were used in phase-II to generate ANN models $(V^p,\ t_p)$ are given in Table (1). The ANNCM with five MLPs (Figure (4)) of same topology (i.e., number layers, number of neurons in each layer are same) with different initial weights is used to invert the model-I by assigning 51 samples $V^p = \left[V^p(x_1)\ \ ...\ \ ...\ \ ...\ \ V^p(x_{51})\right]^T$ to the input layer. Ten neurons with hyperbolic tangent transfer functions are used for hidden layer. Three neurons with linear transfer functions are used for output layer to extract the required parameters (z, θ, A). While training the networks the set $T = \{(V^1, t_1), (V^2, t_2), ..., (V^P, t_P)\}$ is randomly divided into three subsets namely training, validation and testing sets, each are containing 70%, 15% and 15% models respectively. The performance of each MLP is calculated using Eq. (4) and weights are adjusted according to Eq. (5). Output of ANNCM is computed by *ensemble averaging* method and given in Table (2). The vertical magnetic anomaly $V(x)$ and the ANNCM inversion response are shown in Figure (7). The Hilbert transform $H(x)$ and the modified Hilbert transform $MH(x)$ of the vertical magnetic anomaly $V(x)$ are computed and shown in Figure (8). The well trained network can invert any data that falls within the training range in almost no time.

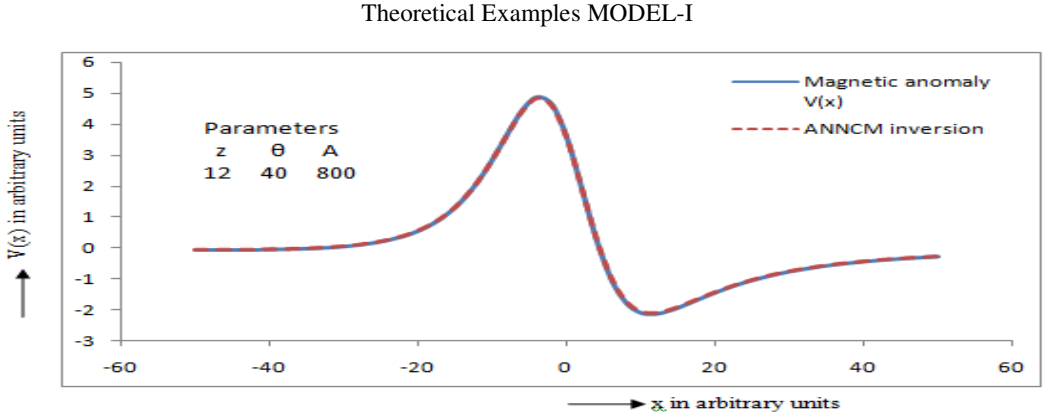


Figure 7 Vertical magnetic anomaly V(x) and ANNCM inversion response of model-I

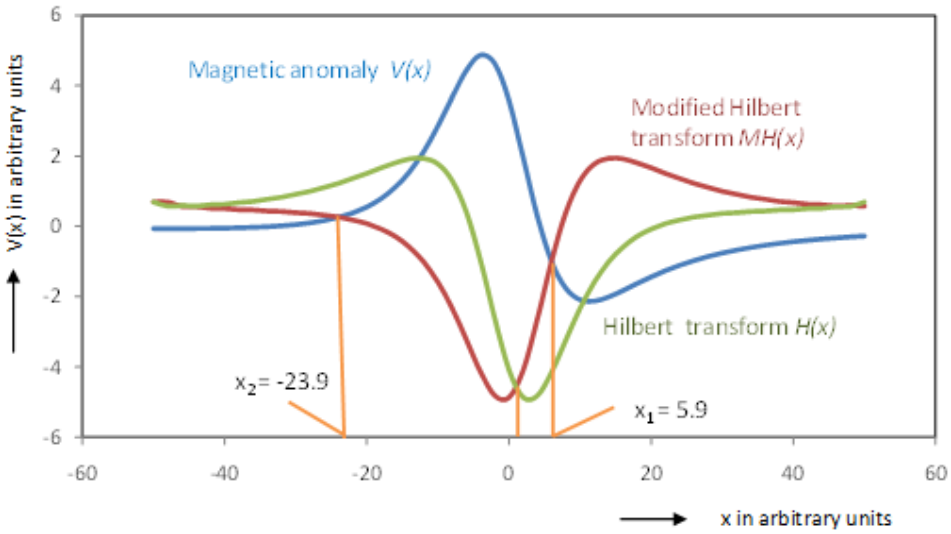


Figure 8 Vertical magnetic anomaly the Hilbert transform and modified Hilbert transform of model-I

4.1 Effect of Random Noise

Ten percent of Gaussian random noise is added to the vertical magnetic anomaly $V(x)$ of model-I [Figure (7)] and is shown in Figure (9). As in the case of noise free analysis, of magnetic anomalies, the values of parameters obtained during phase-I are $\tilde{z} = 9.61$, $\tilde{\theta} = 36.47^{\circ}$ and $\tilde{A} = 424.51$. The range of parameters, number of steps and the number of ANN models that are generated in phase-II are given in Table (1). The ANNCM inversion response is shown in Figure (9). The Hilbert transform $H(x)$ and the modified Hilbert transform $MH(x)$ of the noisy vertical magnetic anomaly $V(x)$ are computed and shown in Figure (10). The result of the ANNCM inversion parameters is given in Table (2).

Theoretical Example with noise MODEL-II

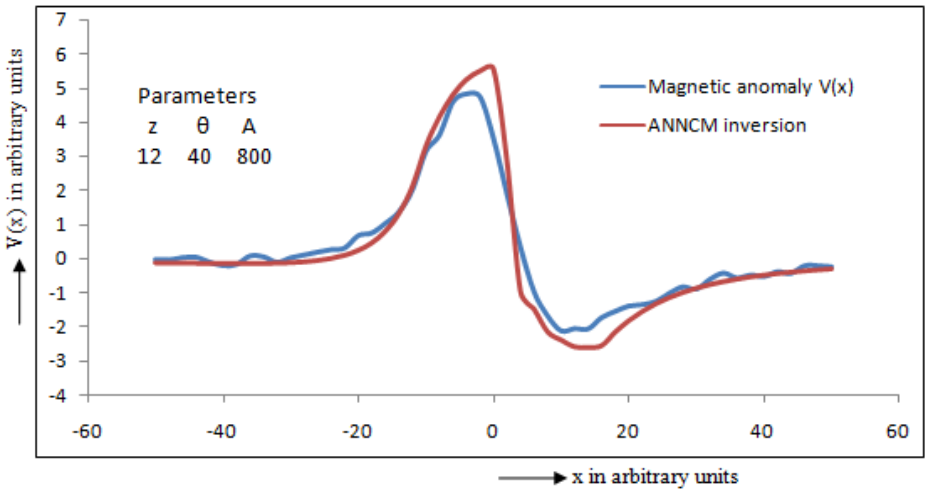


Figure 9 Noisy vertical magnetic anomaly and ANNCM inversion response of model-II

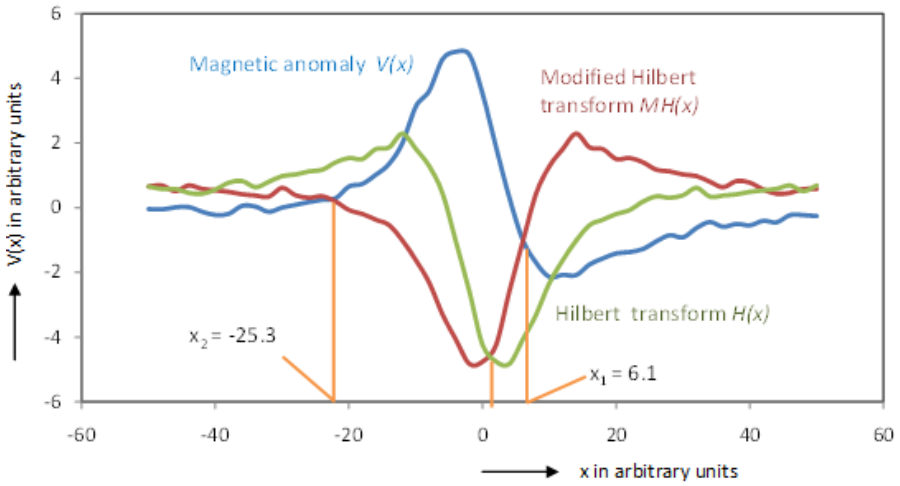


Figure 10 Vertical magnetic anomaly V(x) the Hilbert transform H(x) and modified Hilbert transform MH(x) of model-II.

Table 1 Range of input parameters and number of ANN generated models (Phase-II)

Examples	z	θ	A	**Number of ANN models**
Model-I	6 – 12 (5)	31^{o} – 41^{o} (6)	405 – 565 (9)	**270**
Model-II (with noise)	6.6 – 12.6 (5)	31.4^{o} – 41.4^{o} (6)	374 – 474 (10)	**300**

	Parameters	z*	θ	A*
Theoretical Model-I	Assumed values	12.00	$40^{o}00'$	800.00
	ANNCM processed parameters	12.05	$40^{o}02'$	799.60
	Error in percentage	0.41	0.08	0.05
Theoretical Model with Noise	Assumed Values	12.00	$40^{o}00'$	800.00
	ANNCM processed parameters	10.85	$41^{o}17'$	872.93
	Error in percentage	9.58	3.20	9.11

Table 2 Theoretical examples (* in arbitrary units)

5. FIELD EXAMPLE

The applicability of the proposed technique is demonstrated on an observed vertical magnetic data near Karimnagar district, Telangana, India (Srinivas 1998) and is shown in Figure (11). The

total length of the profile (182.60 meters) was digitized into 60 equal parts at an interval of 3.0433m. The quality of the data is determined by the signal to noise ratio (*SNR*) and is given as:

$$SNR = \frac{m}{s},$$

where m is the mean and s is the standard deviation of the data. If the ratio is less than 3, the data is assumed to be very poor quality. If the ratio is greater than 3, then the level of noise is negligible and the data shall be considered clean. Signal to noise ratio of the field data is calculated and is given by:

$$SNR = \frac{5017.1}{646.8136} = 7.7567$$

The appropriate values of parameters obtained in phase-I are $\tilde{z} = 24.38m,\ \tilde{\theta} = 73^{o},$ $\tilde{A} = 46592000$ Three hundred training models were created by assigning different values to (z , θ , A) in a close range of $(\tilde{z}, \tilde{\theta}, \tilde{A})$ which were used in phase-II are as follows:

- the depth z (19m – 29m), with five points in this range
- the inclination θ (70^{o} – 80^{o}), with six points in this range
- the constant A (46562000 – 46622000), with ten points in this range;

ANNCM inversion response compared with the field data are shown in Figure (11). The Hilbert transform $H(x)$ and the modified Hilbert transform $MH(x)$ of the vertical magnetic anomaly $V(x)$ are computed and shown in Figure (12). The estimated parameters are given in Table (3). Results shown are better and agree well with other inversion methods (Table 3).

Methods	**z (in meters)**	θ	A
Gradient method [26]	23.23	$56^{o}00'$	….
Modified Hilbert transform Technique [31]	21.4	$46^{o}00'$	….
Present Artificial Neural Network Committee Machine	22.66	$78^{o}18'$	47493187.04

Table 3 Field Example (Vertical magnetic anomaly, near Karimnagar, Telangana, India)

6. RESULTS AND DISCUSSION

During training network, 670 training models were used for both theoretical and field data for which Levenberg-Marquart algorithm is very much suitable. From Table (2), it is observed that the results in general agree with the assumed values. However, the addition of random noise level to the magnetic anomaly and subsequent analysis show a marginal variation implying that the effect of such noise is almost negligible in the present method. The method with LM algorithm shows the best performance in extraction of parameters of a model. Hence, determination of the depth and inclination of various structures from magnetic data can be solved effectively.

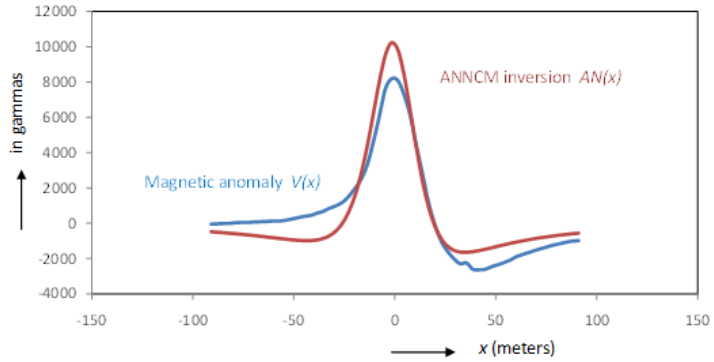


Figure 11 Field Example Vertical magnetic anomaly over a narrow band of quartz magnetite, near Karimnagar District, Telangana, India.

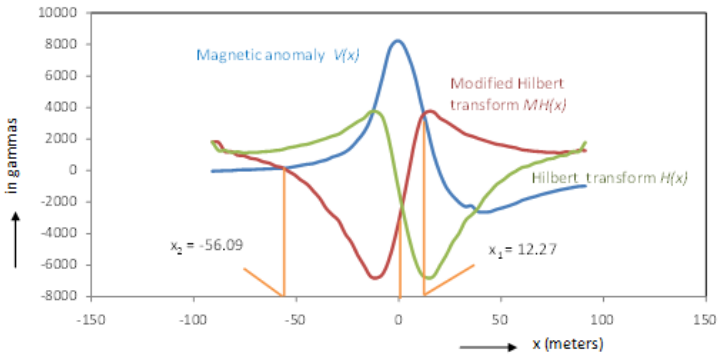


Figure 12 Vertical magnetic anomaly the Hilbert transform and modified Hilbert transform of field data.

7. CONCLUSIONS

The accuracy of ANNCM inversion results is fairly good. The ANN committee machine analysis of magnetic inversion is simple and elegant and the method is effective even in the presence of noise. In addition, it is independent of analytical nature of the data.

REFERENCES

[1] Alamdar K, Kamkare-Rouhani A & Ansari A H, (2015) "Interpretation of the magnetic data from anomaly 2c of Soork iron ore using the combination of the Euler deconvolution and TDX filter", Arabian Journal of Geosciences, Vol. 8, No.8, pp 6021-6035.

[2] Amjadi A & Naji J, (2013) "Application of genetic algorithm optimization and least square method for depth determination from residual gravity anomalies", International Research Journal of Applied and Basic Sciences, Science explorer publications, Vol. 5, No.5, pp 661- 666.

[3] Edwin K P, Chong & Stanislaw H Zak,(2001) An introduction to optimization, Second edition, A Willey-Interscience Publication.

[4] Fitian R Al-Rawi, (2009) "Magnetic depth estimation of dyke- like bodies by using Fraser filter", Journal of Al-Anbar University for Pure Science, Vol. 3, No.1, pp 89-97.

[5] Guoqing Ma & Lili Li., (2013) "Direct analytic signal (DAS) method in the interpretation of magnetic data", Journal of Applied Geophysics, Vol. 88, pp 101-104.

[6] Hagan M T & Menhaj M B., (1994) "Training feedforward networks with the Marquardt algorithm", IEEE Transactions on Neural Networks, Vol. 5, No.6, pp 989-993.

[7] Haykin S., (1999) Neural networks, A Comprehensive Foundation, 2nd edition, Prentice Hall, New Jersey, 842pages.

[8] Ibrahim Kara, Mustafa Ozdemir & Ali Ismet Kanli, (2003) "Magnetic interpretation of horizontal cylinders using displacement of the maximum and minimum by Upward Continuation", Journal of the Balkan Geophysical Society, Vol. 6, No.1, pp 16-20.

[9] Kaftan I, Sindirgi P & Akdemir O, (2014) "Inversion of self potential anomalies with multilayer perceptron neural networks", Pure and Applied Geophysics, Vol.171, No.8, pp 1939-1949.

[10] Karlik B & Olgac A V, (2011) "A performance analysis of various activation functions in generalized MLP architectures of neural networks", International Journal of Artificial Intelligence and Expert Systems, Vol.1, No.4, pp 111-122.

[11] Keating P & Pilkington M, (2004) "Euler deconvolution of the analytic signal and its application to magnetic interpretation", Geophysical Prospecting, Vol.52, pp 165-182.

[12] Krasnopolsky V M, (2007) "Reducing uncertainties in neural network Jacobians and improving accuracy of neural network emulations with ensemble approaches", Neural Networks, Vol.20, No.4, pp 454-461.

[13] Lashin A & Din S S El, (2013) "Reservoir parameters determination using artificial neural networks: Ras Fanar field, Gulf of Suez, Egypt", Arabian Journal of Geosciences, Vol.6, No.8, pp 2789-2806.

[14] Levenberg K, (1944) "A method for the solution of certain nonlinear problems in least squares", Quarterly of Applied Mathematics, Vol.2, pp 164-168.

[15] Li Y & Oldenburg D W, (2003) "Fast inversion of large scale magnetic data using wavelet transforms and a logarithmic barrier method", Geophysical Journal International, Vol.152, pp 251-265.

[16] Mansour A Al-Garni, (2009) "Interpretation of some magnetic bodies using neural networks inversion", Arabian Journal of Geosciences, Vol.2, No.2, pp 175-184.

[17] Mansour A Al-Garni, (2011) "Spectral analysis of magnetic anomalies due to a 2-D horizontal circular cylinder, A Hartley transforms technique", SQU Journal for Science, Vol.16, pp 45-56.

[18] Marquardt D, (1963) "An algorithm for least-squares estimation of nonlinear parameters", Journal of the Society for Industrial and Applied Mathematics, Vol.11, No.2, pp 431-441.

[19] McCulloch W S & Pitts W, (1943) "A logical calculus for ideas imminent in nervous activity", Bulletin of Mathematical Biophysics, Vol. 5, pp 115-133.

[20] Mohan N L, Babu L, Sundararajan N & Seshagiri Rao S V, (1990) "Analysis of magnetic anomalies due to some two dimensional bodies using the Mellin transform", Pure and Applied Geophysics, Vol.133, pp 403-428.

[21] Moreau F, Gilbert D, Holschneider M & Saracco G, (1999) "Identification of sources of potential fields with continuous wavelet transform, Basic theory", J. Geophys. Res, Vol.104, pp 5003-5013.

[22] Murthy K S R & Mishra D C, (1980) "Fourier transform of the general expression for the magnetic anomaly due to long horizontal cylinder", Geophysics, Vol.45, pp 1091-1093.

[23] Naftaly U, Intrator N & Horn D, (1997) "Optimal ensemble averaging of neural networks", Network: Computation in Neural Systems, Vol.8, pp 283-296.

[24] Parker Gay Jr. S, (1965) "Standard curves for magnetic anomalies over long horizontal cylinders", Geophysics, Vol.30, pp 818-828.

[25] Pourghasemi H R, Pradhan B & Gokceoglu C, (2012) "Application of fuzzy logic and analytical hierarchy process (AHP) to landslide susceptibility mapping at Haraz watershed, Iran", Natural Hazards., Vol.63, No.2, pp 965-996.

[26] Radhakrishna Murthy I V, Visweswara Rao C & Gopala Krishna C, (1980) "A gradient method for interpreting magnetic anomalies due to horizontal circular cylinders, infinite dykes and vertical steps", Journal of Earth System Science, Vol. 89, No. 1, pp 31-42.

[27] Rao B S R, RadhaKrishna Murthy I V & Visweswara Rao, (1973) "A direct method of interpreting gravity and magnetic anomalies, The case of a horizontal cylinder", PAGEOPH, Vol.102, pp 67-72.

[28] Rosenblatt F, (1958) "The perceptron: A probabilistic model for information storage and organization in the brain", Psychological Review, Vol. 65, pp 386-408.

[29] Saumen Maiti, Vinit C Erram, Gautam Gupta & Ram Krishna Tiwari, (2012) "ANN based inversion of DC resistivity data for groundwater exploration in hard rock terrain of western Maharashtra (India)". Journal of Hydrology, Vols.464–465: pp 294-308.

[30] Sibi P, Allwyn Jones S & Siddarth P, (2013) "Analysis of different activation functions using back propagation neural networks", Journal of Theoretical and Applied Information Technology, Vol.47, No.3, pp 1264-1268.

[31] Srinivas Y., (1998) Modified Hilbert transform- A tool to the interpretation of geopotential field anomalies, Thesis, Osmania University, Hyderabad, India.

[32] Srinivas Y, Stanley Raj A, Muthuraj D, Hudson Oliver D & Chandrasekar N, (2010) "An application of artificial neural network for the interpretation of three layer electrical resistivity data using feed forward back propagation algorithm", Current Development in Artificial Intelligence, Vol.1, No.1-3, pp 1-11.

[33] Sundararajan N, Mohan N L, Vijaya Raghava M S & Seshagiri Rao S V, (1985) "Hilbert transform in the interpretation of magnetic anomalies of various components due to thin infinite dyke", PAGEOPH, Vol.123, pp 557-566.

[34] Sundararajan N, Umashankar B, Mohan N L & Seshagiri Rao S V, (1989) "Direct interpretation of magnetic anomalies due to spherical sources-A Hilbert transform method", Geophysical Transactions, Vol.35, No.3, pp 507-512.

[35] Sundararajan N & Srinivas Y, (1996) "A modified Hilbert transform and its applications to self-potential interpretation", Journal of Applied Geophysics, Vol.36, pp. 137-143.

[36] Sundararajan N, Srinivas Y & Laxminarayana Rao T, (2000) "Sundararajan Transform – a tool to interpret potential field anomalies", Exploration Geophysics, Vol. 31, No. 4, pp 622 -628.

AUTHORS

Dr. M. Bhagwan Das post graduated in Mathematics from the Kakatiya University with gold medal followed M.Phil from University of Hyderabad and Ph.D in Mathematics from Osmania University, Hyderabad, India. He is currently head of the department, Mathematics at Sree Triveni Educational Institutions, Hyderabad. He worked on an inversion of geophysical problems using Neural Networks and Hilbert transformation with Prof. N. Sundararajan. His research is centered on development of new algorithms, Neural Networks, Wavelet Transforms, Fractals and their applications etc.

Dr. Narasimman Sundararajan graduated in Mathematics from the University of Madras followed by an M.Sc (Tech) and Ph.D in Geophysics from Osmania University, India. Began a career as a Research Scientist and later switched over to teaching in Osmania University where he became a Professor in 2004. Currently he is in the Department of Earth Sciences, Sultan Qaboos University, Oman. Published more than 90 research papers in the leading International journals besides a book and a couple of Book chapters and supervised several Ph.Ds in Geophysics as well as Mathematics. Brought out a few innovative tools for processing and interpreting of various geophysical data besides mathematical concept called "Sundararajan Transform". Implemented several research projects including one on Uranium exploration. Member of XIV Indian Scientific Expedition to Antarctica during 1994-95. Introduced a valid and viable approach to multidimensional Hartley transform in contrast with the definition of Prof R N Bracewell from Stanford University, USA. For his overall significant research contribution, Govt. of India has conferred upon him the National Award for Geosciences in 2007. His research interests are varied and wide including geophysical data processing, mineral and ground water exploration, earth quake hazard assessment studies etc. In 2015, Dr. Sundararajan joined as an Associate Editor of Arabian Journal of Geosciences (Springer) responsible for evaluating submission in the field of theoretical and applied geophysics.

AUTHOR INDEX