International Journal of Computer Applications Technology and Research
Volume 5–Issue 12, 725-729, 2016, ISSN:-2319–8656

Review on Security Techniques using Cloud Computing

Supreet Kaur
Guru Nanak Dev University, Amritsar, Punjab.
India

Sonia Sharma
Guru Nanak Dev University, Amritsar, Punjab.
India

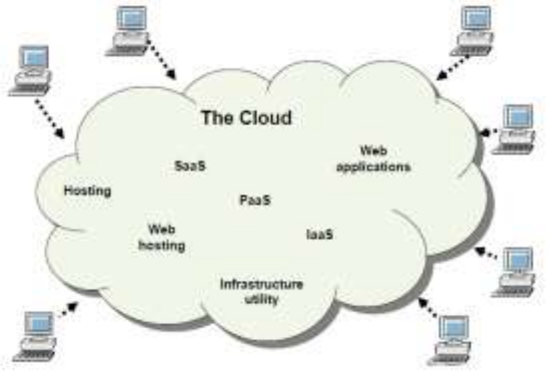
Abstract: Cloud Computing is the nascent technology which is based on Pay-Per-Use Model. Cloud computing is emerging as a model of "Everything as a Service" (XaaS). Cloud Computing is computing paradigm where applications, data bandwidth and IT services are provided over the Internet. Cloud Computing is a relatively new computing model that provides on demand business and IT services over the Internet. The main concerns in adapting Cloud Computing is its security, different security risks that affects the cloud environment in the area of confidentiality, Integrity and computing on data is thoroughly investigated.

Keywords: Cloud Computing, Security, Cloud Security Reference Model, Principal Security Dangers to Cloud Computing, Identity Management, SSL Overview

1. INTRODUCTION

The term cloud is "A network that delivers requested virtual resources as a service."**[D]** Cloud Computing refers to application and services that run distributed network using "Virtualized Resources" and accessed by common Internet Protocol and networking standard. **[K]** The need of cloud computing are as :

a) Cloud Computing is a compelling paradigm.
b) Making internet the ultimate resource of all computing needs. **[I]**



NIST Model stands for the US National Institute for Standards and Technology. It has the set of working definition that separate cloud computing into service model and the deployment model. **[F]** NIST Cloud Model does not address a intermediately services such as transaction or service brokers, provisioning and interoperability services that from the basis for many cloud computing.

1.3 CLOUD VULNERABILITIES

In computer security, a vulnerability is a weakness which allows an attacker to reduce a system's information assurance. Vulnerability is the intersection of three elements :

1.1 CHARACTERISTICS OF CLOUD COMPUTING

a) **On-Demand Self-Service :** Computing resources can be gathered and used at anytime without the need for manual interaction with cloud service providers.
b) **Poor of Virtualized Resources :** It focuses on delivering IT services through resource pool.
c) **Broad Network Access :** The available resources can be accessed over a network using "Heterogeneous Devices" such as Laptops or Mobile Phones.
d) **Measured Service :** Resource usage is measured using appropriate metrics such monitoring storage usage, CPU Hours, bandwidth usage etc.
e) **Rapid Elasticity :** A user can quickly obtain more resources by scaling out from the cloud. **[B]**

1.2 CLOUD SUPPORT TECHNIQUES

Cloud computing has leveraged a collection of existing techniques, such as Data Center Networking (DCN), Virtualization, distributed storage, MapReduce, web applications and services, etc. There are techniques are the followings :

I. **Modern of data center :** It provides massive computation and storage capability by composing thousands of machines with DCN techniques.

II. **Virtualization :** With virtualization, multiple OSs can core side on the same physical machine without interfering each other.

III. **MapReduce :** MapReduce is a programming framework that supports distributed computing on mass data sets. **[C]**

a) A system susceptibility or flaw.
b) An attacker access to the flaw.
c) An attacker capability to exploit the flaw. **[E]**

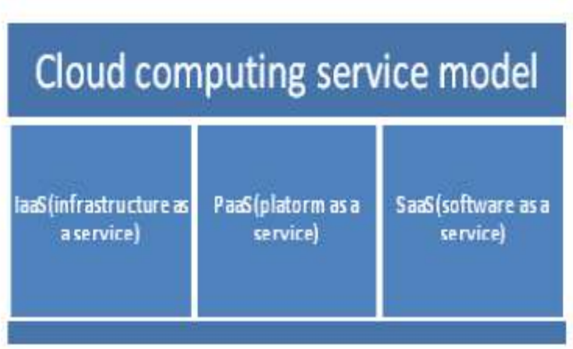
1.4 MODEL OF CLOUD COMPUTING

There are two Model of Cloud Computing are the followings :

a) Service Model
b) Deployment Model

a) Service Model :

It tells us what are services the cloud is providing. These types are as the followings :

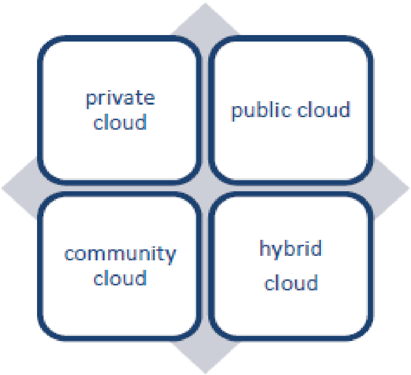

Cloud computing service model
IaaS(infrastructure as a service)
PaaS(platorm as a service)
SaaS(software as a service)

i. **Software as a Service (SaaS) :** It manages the user data and interaction. It does not require client installation just a browser or other client device and network connectivity. **[D]**

ii. **Platform as a Service (PaaS) :** It provides virtual machines, operating system application services, deployment framework, transaction and control structure.

iii. **Infrastructure as a Service (IaaS) :** It provides virtual machines, virtual storage and virtual infrastructure and other hardware access as resources that client can provision. **[A]**

b) **Deployment Model :** It tells us where the cloud is located and The National Institute of Standards and Technology (NIST) defines four cloud deployment types :



i. **Private Clouds :**

The cloud infrastructure is used solely by the organization that owns it. May reside in-house or off premises. There are two types of Private Clouds are the following :

a) **Private Internal Clouds :** The organization acquires the necessary hardware and maintains it for itself.

b) **Private External Clouds :** The organization pays a cloud provider to provide this as a service. **[L]**

ii. **Public Clouds :**

Service Provider lets clients access the cloud via the Internet.

iii. **Community Clouds :**

Used and Controlled by a group of organizations with a shared interest.

iv. **Hybrid Clouds :**

Composed of two or more clouds (private, public or community) that remain unique entities, but that can interoperate using standard or proprietary protocols. **[A]**

1.5 ADVANTAGES OF CLOUD COMPUTING

a) Cloud Computing environment are scalable system and a customized software stack.
b) In addition to the IT industry, even small scale business can adopt this environment model.
c) **Reduced setup costs :** The cost involved in setting up a data center are not very high. **[G]**
d) **Lower Cost :** Because lower cost operate at higher efficiencies and with greater utilization significant cost reduction are often encounter. **[G]**
e) **Outsourced IT management :** Capabilities required and how outsourcing vendors are developing them.

1.6 DISADVANTAGES OF CLOUD COMPUTING

1. **Network Failure :** It can result in loss to the company by causing extensive time delays.
2. **Quality of Service :** It is a key determining factor in the efficiency of a cloud network. **[J]**
3. Cloud Computing is a "**stateless system**" in order for a communication service on distributed system.
4. All cloud computing application suffers from the "**inherent latency**" i.e. intrinsic in the WAN connectivity.
5. The lack of state allow messages to travel over different routes and data to arrive out of sequence and communication to succeed even when the system is faulty.

2. SECURITY IN CLOUD COMPUTING

Cloud Computing presents an added level of risk because essential services are often outsourced to a third party. Cloud Computing shifts much of the control over data and operations from the client organizations to it cloud provider :

a) Clients must establish a trust relationship with the providers and understand the risks.
b) A trust but verify relationship is critical. **[D]**

Security areas to focus on include :

I. Recognizing Security risks
II. Carrying out required security tasks
III. Managing user identity
IV. Using detection and forensics programs
V. Encryption data
VI. Creating a security plan **[H]**

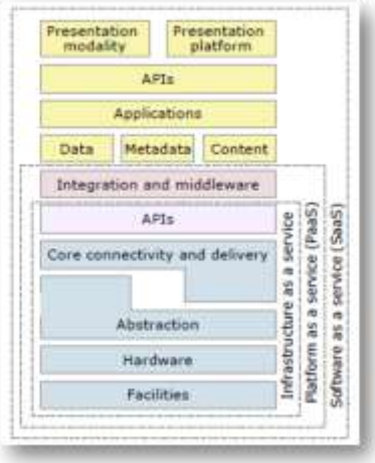
2.1 CLOUD SECURITY REFERENCE MODEL

Integration of Security into cloud reference model. The relationship and dependencies between these are important to fully grasp the security risks to cloud computing :

a) IaaS is the base of all cloud services.
b) PaaS is layered on top of IaaS.
c) SaaS is built upon PaaS. **[D]**

Layered Architectures inherit capabilities :

I. These capabilities includes operations and functionality.
II. Unfortunately, they also inherit risks, including security risks. **[D]**


Presentation modality
Presentation platform
APIs
Applications
Data
Metadata
Content
Integration and middleware
APIs
Core connectivity and delivery
Abstraction
Hardware
Facilities
Infrastructure as a service
Platform as a service (PaaS)
Software as a service (SaaS)

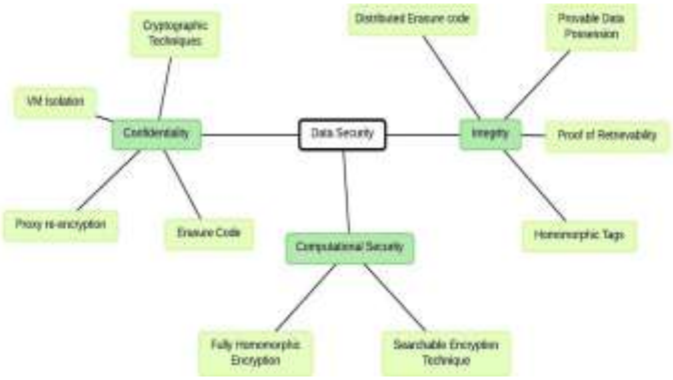
I. **Infrastructure as a Service (IaaS) in Security :** IaaS provides few application features but tremendous flexibility. This opens up the application layer and middleware layer requiring the cloud provider to focus the security, capability on the Operating System (OS) and underlying infrastructure.

II. **Platform as a Service (PaaS) in Security :** PaaS provides a layer in which developer works providing them the freedom to create functionality. The increase flexibility, removes additional security layering that was providing in SaaS. PaaS Security comprises of two types :
   a) Security of the PaaS platform itself, normally provided by cloud service provider.
   b) Security of customer applications deployed on a PaaS platform. **[C]**

III. **Software as a Service (SaaS) in Security :** SaaS in Security is a software deployment model where applications are remotely hosted by the service provider and made available to customers on demand, over the Internet. SaaS is rapidly emerging as the dominant delivery model.

2.2 IMPORTANT TYPES OF CLOUD DATA SECURITY

There are important three types of Cloud Data Security are as follows :



a) **Confidentiality :** Confidentiality refers to any authorized parties having access to protected data. **[C]**
b) **Integrity :** Integrity refers that data can be modified only by authorized parties or in authorized ways. **[C]**
c) **Computational Cloud Security :** The fundamental service enabled within the cloud paradigm is computation outsourcing. Users can make use of unlimited computing resources in a pay-per-use model. **[C]**

2.3 CLOUD SECURITY RISKS

Security Risks is very complicated area of Cloud Computing for three reasons are as follows :

a) Security is a trusted to the cloud provider; therefore, if the provider has not done a good job, there may be problems.
b) Security is difficult to monitor, so problems may not be apparent until there is a problem.
c) Measuring the quality of the cloud provider's security approach may be difficult because many cloud provider's do not expose their infrastructure to customers. **[D]**

2.4 PRINCIPAL SECURITY DANGERS TO CLOUD COMPUTING

The principal security dangers to cloud computing include dangers that currently exist in pre-cloud computing. These includes are :

a) Virtualization and Multi-Tenancy
b) Non-Standard and Vulnerable APIs
c) Internal Security Breaches
d) Data Corruption or Loss
e) User Account and Service Hijacking **[D]**

a) **Virtualization and Multi-Tenancy :** Virtualization and Multi-Tenancy architects make this possible.

Virtualization and Multi-Tenancy were not designed with strong isolation in place :

I. Hypervisors have extended these risks, potentially exposing the Operating System.
II. Creating an environment where attackers can gain accept at the OS level (hypervisors) and higher level services (functionality and data).

b) **Non-Standard and Vulnerable APIs :** Application Programming Interfaces are the software interfaces that cloud provider offer, allowing customers access into their services. Cloud API's are not standardized, forcing user of multiple cloud providers to maintain multiprogramming interfaces, Increasing complexity and security risk.

c) **Internal Security Breaches :** The IT industry has well documented that over 70% of security violation are internal :

I. This threat is amplified in Cloud Computing as both IT providers and consumers are under a single management domain.

d) **Data Corruption or Loss :** Data Corruption in an amplified since the cloud provider is a source for companies data, not the company itself. These operational characteristics of cloud environment at PaaS and SaaS layers, amplify the threat of data loss or leakage increase.

e) **User Account and Service Hijacking :** User Account and Service Hijacking occurs when attackers obtains your cloud services information and uses it to take over your cloud access. If attackers gain access to cloud user's traditional make an eavesdrop on activities and transaction, manipulate or steal data, return falsified data and redirect clients to illegitimate sites.

2.5 REDUCING CLOUD SECURITY BREACHES

The following steps offer a guideline to reduce cloud security breaches :

1. Implement security best practices including human processes.
2. Implement OS security best practices such as patch management.
3. Implement application & API systems security best practices.
4. Implement strong encryption, SSL, digital signatures & certificate practices.
5. Ensure that auditing & logging is being used to monitor activities.
6. Ensure that strong disaster recovery process exist.
7. Transparency in information & internal management practice.
8. Understanding the human resources requirements.
9. Have a clear level of escalation & notification of a breach, ensuring that you are in the loop if an internal breach occurs with the cloud provider (with your data or another customer’s). **[D]**

2.6 IDENTITY MANAGEMENT

Identity Management is a broad administrative area that deals with identifying individuals in a system & controlling access to the resources in that system by placing restrictions on the established identities of the individuals. **[D]** The benefits of Identity Management are as follows :

a) Improved User Productivity
b) Improved Customer and Partners Services
c) Reduced help desk costs
d) Reduced IT costs

The Aspects of Identity Management are as follows :

a) Centrally locate the data
b) Integrating
c) Strength Authentication
d) Provisioning
e) Single sign-on
f) Security Administration
g) Analyzing Data

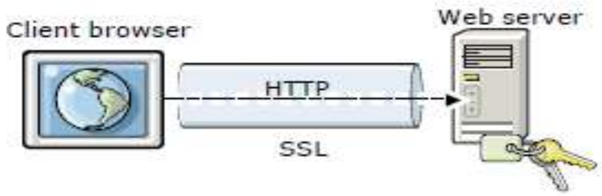
2.7 SSL OVERVIEW

SSL stands for Secure Sockets Layer. SSL provides connection security through :

a) **Communication Privacy :** The data on the connection can be encrypted.

b) **Communication Integrity :** The protocol includes a built-in integrity check.

c) **Authentication :** The client knows who the server is. **[D]**



SSL is the standard security technology for establishing an encrypted link between a web server & a browser. This link ensures that all data passed between the web server & browsers remains private and integral. SSL is an industry standard & is used by millions of websites in the protection of their online transactions with their customers.

Solve the following security problems :

I. Tampering
II. Impersonation
III. Eavesdropping

2.8 CLOUD SECURITY CHALLENGES

a) Indirect administrative accountability.
b) Proprietary cloud vendor implementations can't be examined.
c) Loss of Physical Control.
d) Possibility for massive outages.
e) Encryption needs for cloud computing :

i. Encrypting access to the cloud resource control interface.
ii. Encrypting administrative access to OS instances.
iii. Encrypting access to applications.
iv. Encrypting application data at rest. **[H]**

3. CONCLUSION

Cloud Computing is the promising paradigm for delivered IT services as computing utilities. Cloud are designed to provide services to external user; provider need to be compensated for sharing their resources and capabilities. Security Challenges and the privacy of data are the major obstacles for the success of Cloud Computing. We have performed a systematic review of security issues for cloud environment where we enumerated the main cloud threats and vulnerabilities.

4. REFERENCES

[A] Krishan Kant Lavania , Yogita Sharma , Chandresh Bakliwal, "Review on Cloud Computing Model", International Journal on Recent and Innovation Trends in Computing and Communication

[B] Navdeep Kaur, A Review Paper on various scheduling techniques in cloud computing

[C] Balasubramanian, Review of on various data security issues in cloud computing environment and it solutions

[D] IBM, Fundamentals of Cloud Computing (Student Notebook), WebSphere Education

[E] Cloud Computing Architecture & its Vulnerabilities with Presentation, Vinay Dwivedi (Visual Information Processing and Embedding Systems)

[F] https://en.wikipedia.org/wiki/Cloud_computing

[G] https://www.cse.unr.edu/~mgunes/cpe401/cpe401sp12/lect15_cloud.ppt

[H] https://en.wikipedia.org/wiki/Cloud_computing_security

[I] https://www.youtube.com/watch?v=ae_DKNwK_ms

[J] https://www.tutorialspoint.com/cloud_computing/

[K] https://www.youtube.com/watch?v=bsIZ_-8u4fE

[L] www.thbs.com/downloads/Cloud-Computing-Overview.pdf

International Journal of Computer Applications Technology and Research
Volume 5–Issue 12, 730-732, 2016, ISSN:-2319–8656

Smartphone Remote Detection and Wipe System using SMS

Nilesh Dorge
Department of Computer Engineering, Savitribai Phule Pune University, Pune, India

Atish Pawar
Department of Computer Engineering, Savitribai Phule Pune University, Pune, India

Suraj Khandbale
Department of Computer Engineering, Savitribai Phule Pune University, Pune, India

Abhijit Jachak
Department of Computer Engineering, Savitribai Phule Pune University, Pune, India

Shubham Nirmal
Department of Computer Engineering, Savitribai Phule Pune University, Pune, India

Prof. Jitendra Musale
Department of Computer Engineering, Savitribai Phule Pune University, Pune, India

Abstract: The project based on mobile application which functions on an Android operating system. The objective of this which enable the user to locate the mobile phone in a silent mode to General mode when it is misplaced as well as if it is lost and wipe the data from the device. To create an account the user needs to provide his /her mobile number, a password and 4 trustworthy numbers this completes the registration process. The application, which is still in a deactivation mode, will operate only when the phone is misplaced and the user sends the set password/ pass code from one of the 4 trustworthy numbers to one's own mobile number. This will change the profile of the misplaced phone i.e. switch it from the silent mode to the sound mode. It will also send an acknowledgement to the trustworthy number from which the user has sent the message. Furthermore, it will also provide the location with and also if mobile is lost then we can take back up from another mobile by using same application, we can also wipe the data remotely by sending the message.

Keywords: ***Real-time, location tracking, Android.***

1. INTRODUCTION

In this paper we are focusing on three major topics GPS, Mode conversion (silent to general) and Data Wipe & Recovery. Rapid evaluation of wireless technologies has provided a platform to support system in the domain of location tracking. Mobile devices such as phones are common and internet access is possible everywhere in daily. Worldwide out of 100% people who uses smart phone Probably 80% and above uses android operating system according to International Corporation market researches. Android is common with its open source nature and working capabilities on in expensive mobile devices. Location tracking is continuously monitoring a device by using obtained coordinates with GPS. Nowadays smart phone is mostly used by every human , sometimes some people face problem in finding the mobile are controlled by the smart phones, this Application helps to find the mobile when it is misplaced. This application is useful, when it is in silent and forgotten where it is placed.As enabling all the users to receive advantages and satisfaction, the smartphone have been applied in a variety range and it expand a range of security threat. Specially, security threat of the android phone by loss or stolen may cause the user data disclosure such as credit cards, login IDs, contacts, message, photos etc. To prevent these problems, network operators should provide some security by which intruder can not use mobile devices by and also need to support the remote lock and wipe services which delete users' data as in the state of factory reset.

2. ANDROID BASED ENERGY AWARE REAL-TIME LOCATION TRACKING SYSTEM.

EWAREL adopts a client-server scheme. User client is an android application that performs real-time background tracking and synchronizing to server in a given interval while internet connection is present .It stores location changes when internet access in not possible and synchronize as soon as mobile device is connected to internet. Monitoring client is also an android application to accomplish tracking of people to monitor. Monitoring application alerts when one of the clients does not send location data at predefined interval to server and becomes unreachable.

3. AUTO MODE CONVERSION

3.1 EXISTING SYSTEM

In the previous projects the message sent will be sent as a normal text message to the other mobile .But there is no application developed to change the modes of the mobile automatically by receiving a text message. For mobile recovery we have applications, which use Global Positioning System technology. If we want to change the mode we have to change it manually, this is the main disadvantage. So we have proposed a new system to overcome the problems in the existing system

3.2 PROPOSED SYSTEM

In this project we will send a text message to mobile, it will check the message and it will help in converting the modes. There may be a situation where we cannot convert manually when the mobile is misplaced then it can convert from silent mode to general mode.

Fig 2:Mode conversion from silent to general by sending a message

4. THE REMOTE LOCK AND WIPE SYSTEM

4.1 THE REMOTE LOCK AND WIPE SYSTEM

It consists of a remote control module on a server and a command handling module on a smartphone (see Fig.1). The commands are sent by text message push notification message. For example, when the users send a lock command message to the smartphone via the remote control module, the remote handling module enables the password locking function to lock the smartphone. Similarly, by sending a wipe command message, all personal data is remotely deleted.

4.2 COMMAND INTEGRITY PROBLEM

The remote lock and wipe service will be very useful when the smartphone is lost or stolen. However, there might be the case that the wrong user misuses this function by sending such commands to the normal users in order to interrupt the service. Thus, it is very important to check if the command is originated from the trusted server. In other words, the integrity of the commands must be checked. The traditional way to provide the integrity checking is simply to apply a digital signature scheme such as RSA or DSA signature. Note that RSA or DSA signature requires the key size of 1024 bits (i.e, 128 bytes) long to protect against active attackers over the wireless network. An obstacle here is that the command integrity checking should be provided only with the SMS message which is 80 bytes long

When the SMS command notification is sent, the remote control module creates a secret key from the password using PBKDF. Using HMAC function with the secret key, the message authentication code (MAC) is generated on the command message along with the timestamp which is added to protect against the well-known reply attack. Then, the command message is sent with the MAC to the designated smartphone.

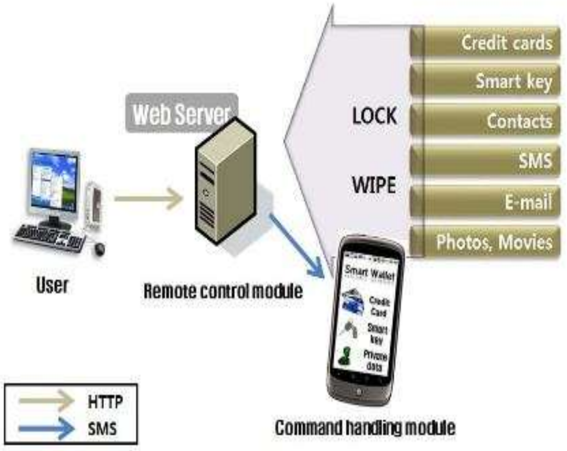


Fig. 1. Remote Lock and Wipe System

In server when the app is started by the user, user can perform 3 actions edit secret code ,start service and add contacts .Edit secret code will give permission to edit the secret code, add contacts will allow to add the contact numbers for client, start service will start the application. Now, client will start the application, after that user will enter the secret code, enter the receiver number and user will send the text message. When server receives the text message, it will check with secret code and perform different operations like tracking the location of device ,mode conversion (silent to General mode) And data wipe and recovery.

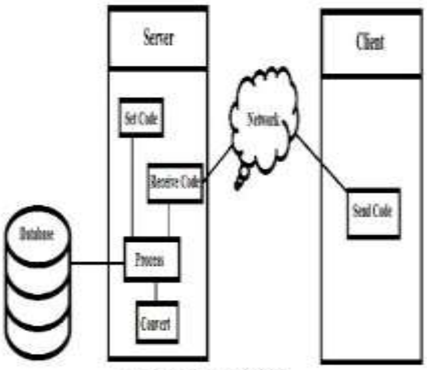


Fig 3: System Architecture

5. CONCLUSION

This application is developed in android platform and this provides higher curiosity to the android users. This system provides the location of device and efficient method for finding out the smart phones where it is placed(in silent mode) it is helpful in changing from silent mode to General Mode and also The remote lock and wipe service is necessary to protect against the private data disclose. At the same time, it must prevent the unknown user from launching DOS attacks that sends such commands to the normal users intentionally.

6. REFERENCES

[1]http://compnetworking.about.com/od/basicnetworkingconcepts/a/networktypes.html

[2]http://homeguides.sfgate.com/advantages-smart-house 8670.html

[3]http://en.wikipedia.org/wiki/Mobilesecurity

[4]Smartphone OS Market Share http://www.idc.com/prodserv/smartphone os-marketshare.jspast accessed January 15th, 2105.

[5]K.Jones and L. Liu, What where wiAn analysis of millions of wi access points,PORTABLE07. IEEE International Conference on Portable Information Devicespp. 25,29,2007

[6]Y . Cheng,Y. Chawathe, A. LaMarca, and J. Krumm, Accuracy characterization for metropolitan scale Wi - Fi localization, in Proceedings of the 3rd international conference on Mobile systems, applications, and services. ACM p.245 , 2005.

[7]Y.F .Chang, C.S. Chen, and H. Zhou, " Smart Phone For Mobile Commerce",Computer Standards Interfaces ,Vol . 31, Issue 4, June, 2009

[8]Sha k G. Punfa and Richard P. Mislan , "Smartphone Device Analysis", Small Scale Digital Device Forensics Journal , Vol. 2, No. 1, June, 2008.

[9]A . Menezes , P. van Oorschot , and S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996

[10]RSA Laboratories, "PKCS 5 v2.0: Password - Based Crypto - graphy Standard",March, 1999.

Empirical Investigation of Instant Messaging Security in a Virtual Environment

Peter S. Nyakomitta
School of informatics and Innovative systems
Jaramogi Oginga Odinga University of Science and Technology
Kenya

Dr. Solomon Ogara
School of informatics and Innovative systems
Jaramogi Oginga Odinga University of Science and Technology
Kenya

Dr. Silvance Abeka
School of informatics and Innovative systems
Jaramogi Oginga Odinga University of Science and Technology
Kenya

***Abstract* -** Use of instant messaging services is becoming increasingly popular with Internet based systems like America Online's Instant Messaging (AIM), Microsoft's MSN Messenger, Yahoo! Messenger, WhatsApp, Viber, Kakaotalk, Skype and face book instant messenger. These tools support any process where quick response and rapid problem solving are needed, and where faster communication than emails or telephones is useful. More and more people are enjoying the convenience and simplicity provided by the real-time messaging systems in their day-to-day life. Moreover, the instant messaging services have also found applications in business. In this application domain, the instant messaging services are employed for communicating with customers and partners, offering customer support, receiving real-time alerts, as well as management and project coordination. Despite their heavy utilization, public instant messaging systems have been criticized for having a number of security weaknesses. These weaknesses originate from the facts that the instant messaging clients are always on, those logs can contain sensitive information, and that the communication goes through an externally controlled server. Most of the instant messaging services were never intended for secure communication in the first place. The rapid growth in the number of public instant messaging users has therefore created a new security concern for information technology managers. In this paper, a prototype instant messaging was developed and employed to investigate some of the security challenges in instant messaging applications. The results indicated that upon following the TCP stream, the instant messages were in plain text in the sending and receiving communication devices interfaces and therefore prone to eavesdropping. As such, the researchers propose a port-based algorithm that would scramble the data packets at the end devices, requiring the users to input decryption keys for the data to be transformed into human readable format.

Keywords: ***Instant Messaging (IM), Plain Text, prototype, Client, Server, Security***

1. INTRODUCTION

The potential threats when using instant messaging services include the spread of malicious code, instant messaging software vulnerabilities, leakage of sensitive information, monitoring and retention issues, and lack of accountability. In their study, Weissbrot and Alison (2016) explain that the enterprise usage of instant is growing in both volume and importance. The use of these tools benefit their users in facilitating faster decision-making process, higher productivity and lower telecommunication costs. At the same time as, the instant messaging threats such as viruses are rapidly gaining attention as attackers begin to shift their focus from better-protected email systems to these networks.

Moreover, Mark (2015) report that spam messages can also be spread through the instant messaging tools. The spam that a user receives via these services is referred to as spim. Popular instant messaging clients have, just like any other software application, have a history of common security vulnerabilities. This means that installing an instant messaging client has the potential of introducing new vulnerabilities to a computer system. Confidentiality, which deals with the protection of organizational data or government data from illegal access, is a major concern when using a public instant messaging service for communication. This so according to Vinnie and Belvin, (2012), because in public instant messaging networks, communications exchanged between users are normally routed through instant messaging server farms which are controlled by the service providers themselves. In situations where client instant messaging software has a peer-to-peer capability, users can communicate with each other without passing through these servers.

Lin et al., (2016) note that no matter which mode is being utilized for communication purposes, this traffic is vulnerable to eavesdropping because most public clients do not possess any encryption capability. The consequences are that it is possible for sensitive information to be read or sniffed by unauthorized users. The situation can be even worse when public instant messaging services are used to communicate with individuals outside an organization. This may lead to the leakage of sensitive organizational classified data.

11. LITERATURE REVIEW

In this section, the researcher discuss on the literature concerning the related work to the research, which is empirical investigation of instant messaging security in a virtual environment.

2.1. RELATED WORK

In his research work, Wendell (2013) found out that the protocols employed by public instant messaging services are often considered rogue protocols. This because they were specifically designed to evade standard security controls. The consequences are that not only can instant messaging clients be configured to connect through SOCKS or web proxy servers, but as Green (2014) point out, the protocols are also capable of finding their way out through the firewall on their own. They can do this by determining an open port such as transmission control protocol (TCP) port 80, or by tunneling their traffic inside the hypertext transfer protocol (HTTP) requests. These practices make these traffic unrecognizable from standard web traffic. Additionally, the scripting and file transfer capabilities of instant messaging systems might expose an organization to leaks of sensitive information.

According to Andreas and Buchenscheit (2014), most of the productive features provided by the instant messaging applications are only one side of the coin. This is because these applications have vulnerabilities that related to the underlying instant messaging technology. Consequently, these vulnerabilities expose user communications to a number of security threats. Therefore, communication via these applications is regarded insecure. To start with, Jagwani (2016) note that all of the messages and connection information are retained on the application providers' servers. This means that the information communicated across the network is controlled by the provider of the instant messaging utility.

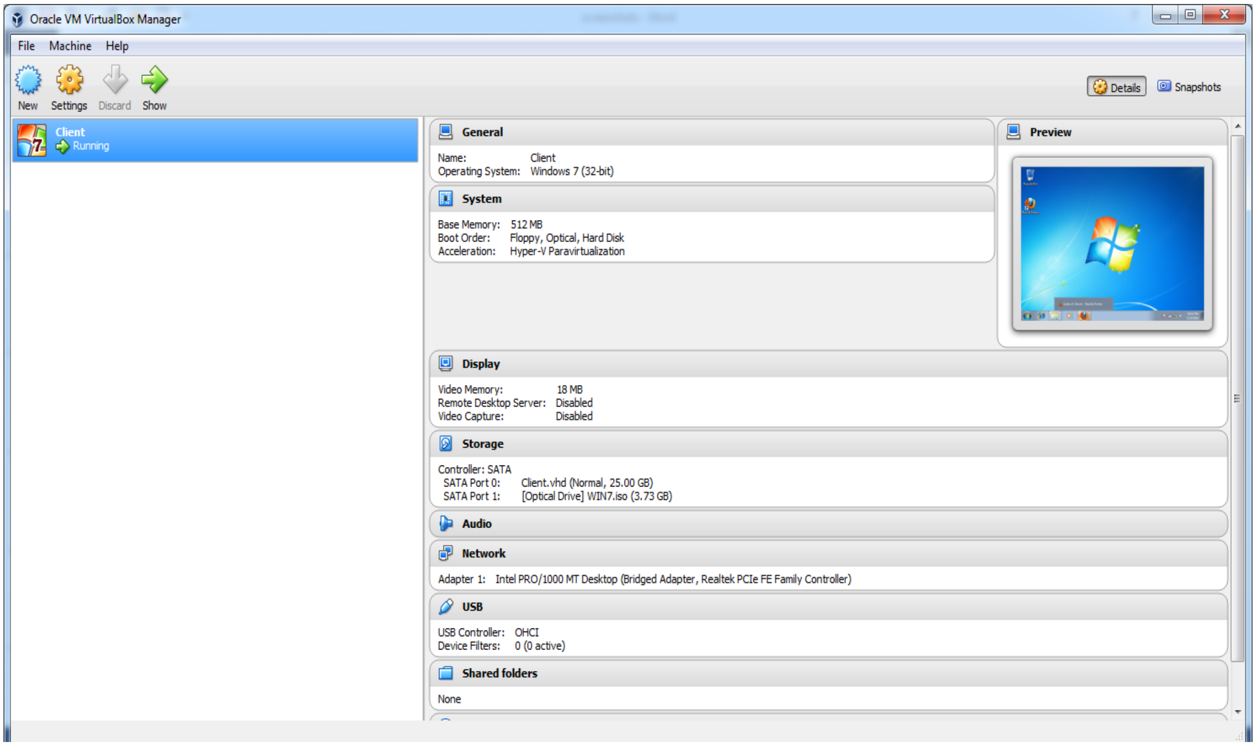
Another serious challenge with instant messaging connections is that the messaging process normally happens in plain text (Frosch et al., 2016). Consequently, this renders them susceptible to eavesdropping. Additionally, instant messaging client software quite often requires the user to expose their open user datagram ports. This gives rise to the threat posed by potential security vulnerabilities of user datagram protocol (UDP) ports. As Smith (2013) points out, UDP is vulnerable to spoofing and denial of service attacks. On the other hand, it is not feasible to spoof an address across the internet using transmission control protocol (TCP). This is because the three way handshake will never complete. Moreover, there are features of instant messaging applications that are threats to security. Such features include presence and status broadcasting, interoperability with others, maintaining a lists of all desired contacts, use of third party servers to provide chat functionality to messenger clients and keeping a log of messages and other events/ activities (Fahrnberger, 2014).

111. RESEARCH METHODOLOGY

This section presents the way the study was carried out. It involves the various steps that the researcher adopted in studying the research problem as well the logic behind those steps which include the research prototyping approach and the procedure is provided as outlined below.

3.1 Prototyping Approach

In this paper, a model that could help demonstrate the security challenges in instant messaging applications in a virtualized environment was developed. The hypervisor was chosen to be Oracle VM Virtual box. This hypervisor is a cross-platform virtualization application, meaning that it installs on the existing Intel or AMD-based computers,. It supports operating systems such as Windows, Mac, Linux or Solaris. It serves to extend the capabilities of the existing computer so that it can run multiple operating systems inside multiple virtual machines at the same time. Using this hypervisor, one can install and run as many virtual machines as he likes. The only practical limiting factors are disk space and memory. Moreover, it is very simple to use and very powerful. This is because it can run on any platform, ranging from small embedded systems or desktop class machines to datacenter deployments and even Cloud environments. This made it the best choice for this paper. Figure 1 shows the interface for VirtualBox hypervisor.



General
Name: Client
Operating System: Windows 7 (32-bit)

System
Base Memory: 512 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Hyper-V Paravirtualization

Display
Video Memory: 18 MB
Remote Desktop Server: Disabled
Video Capture: Disabled

Storage
Controller: SATA
SATA Port 0: Client.vhd (Normal, 25.00 GB)
SATA Port 1: [Optical Drive] WIN7.iso (3.73 GB)

Audio

Network
Adapter 1: Intel PRO/1000 MT Desktop (Bridged Adapter, Realtek PCIe FE Family Controller)

USB
USB Controller: OHCI
Device Filters: 0 (0 active)

Shared folders
None

Figure 1: VirtualBox Interface

3.2 Procedure

After the installation of the VirtualBox hypervisor, a client operating system was installed inside this hypervisor. This client was chosen to be Windows Professional, 32 bit.

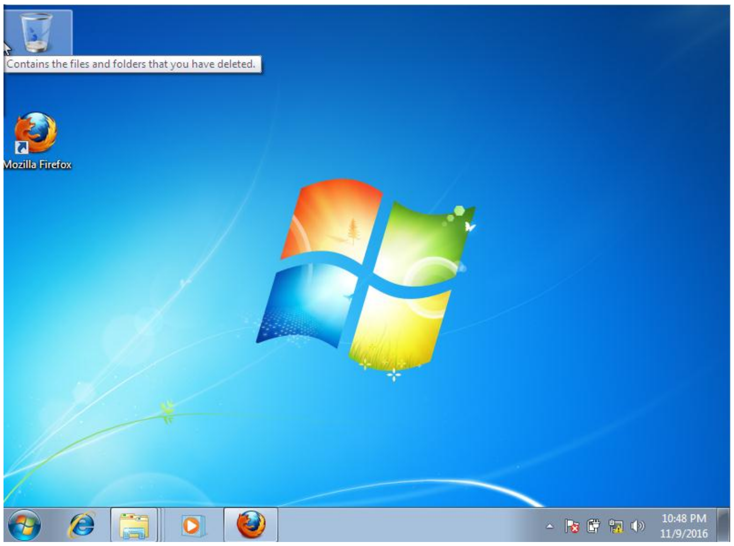


Figure 2: Client Operating System

The 32 – bit client was selected to avoid bus speed mismatch because the underlying host operating system ran 32 –bit. After this, Wamp server was installed inside this client operating system. This was meant to provide a respository for the instant messages through its *PhPMyadmin* feature as shown in Figure 3.

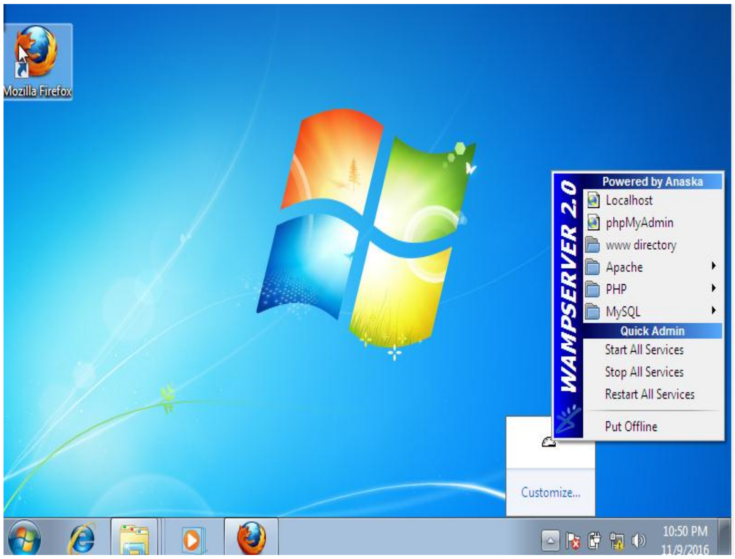


Figure 3: Wamp Server Installation

All the *PhP* coding was done and saved in the *www* folder of *wamp* server. The same *wamp* server was also installed on another machine that was required to achieve a two way instant messaging exchange. An Ethernet cable was then connected between the machines that required exchanging the instant messages and appropriate internet protocol (IP) addresses were assigned using class C subnet. Figure 4 show the typical set up that was employed.

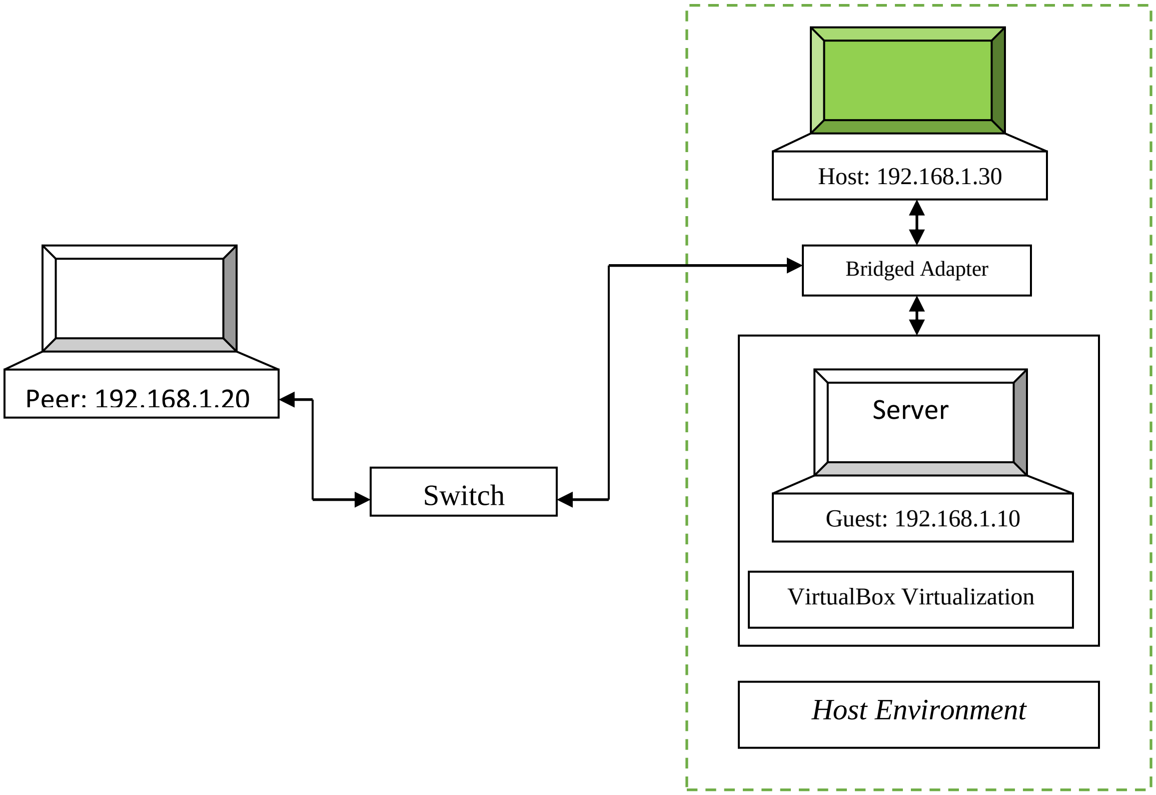


Figure 4: Prototyping Setup

In this paper, the Host operating system refers to the operating system of the physical computer on which VirtualBox was installed. On the other hand, the guest operating system was the operating system that is running inside the virtual machine. Virtual machine (VM) refers to the special environment that VirtualBox creates for the guest operating system while it is running. In other words, the guest operating system is run in a virtual machine. Basically, a virtual machine could be shown as a window on the computer’s desktop, but depending on which of the various frontends of VirtualBox is in use, it can be displayed in full screen mode or remotely on another computer.

Technically, VirtualBox regards a virtual machine as a set of parameters that determine its behavior. They include hardware settings (how much memory the virtual machine should have, what hard disks VirtualBox should virtualize through which container files, or what CDs are mounted) as well as state information (whether the virtual machine is currently running, saved, or its snapshots).

Figure 4 shows that the hypervisor was hosted in address *192.168.1.30*. This hypervisor in turn hosted a client of IP address *192.168.1.10*. In order for the guest operating system to communicate with the host operating system, a bridged adapter was employed as shown in Figure 4. In bridged networking, VirtualBox utilizes a device driver on the host system that filters data from the physical network adapter.

For this reason, this driver is called a net filter driver.

This permits VirtualBox to capture data from the physical network and inject data into it. The effect of this is the creation of a new network interface in software. Ideally, when the guest operating system is utilizing such a new software interface, it looks to the host system as though the guest were physically connected to the interface using a network cable. Effectively, the host can send data to the guest through that interface and receive data from it. This means that one can set up routing or bridging between the guest and the rest of your network.

To enable bridged networking, the *Settings* dialog of a virtual machine was opened; from there navigation was done to the *Network* page. Finally, the selection of *Bridged network* was accomplished in the drop down list for the *Attached to* field as shown in Figure 5.

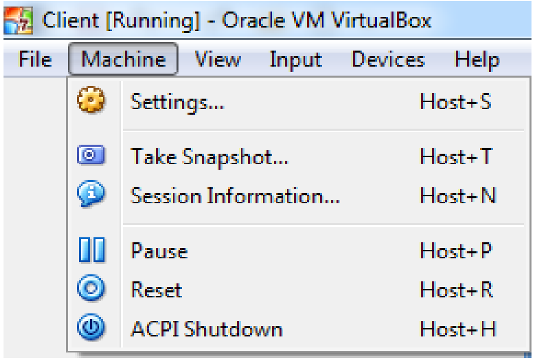
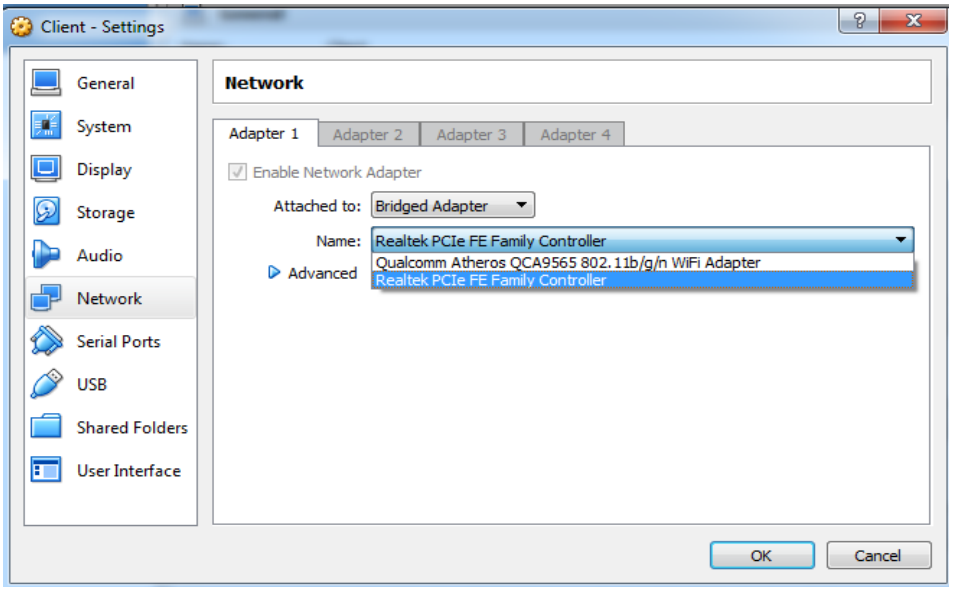


Figure 5: VirtualBox Network Setting

To finish the configuration, the desired host interface was selected from the list at the bottom of the page, which contains the physical network interfaces of the systems. As Figure 6 demonstrates, two network interfaces were detected: *RealTek PCIe FE Family Controller* and *Qualcom Atheros QCA9565802.11b/n Wifi Adapter*. Howver, since the latter adapter was for the wireless connections, the former interface *RealTek PCIe FE Family Controller* was selected.



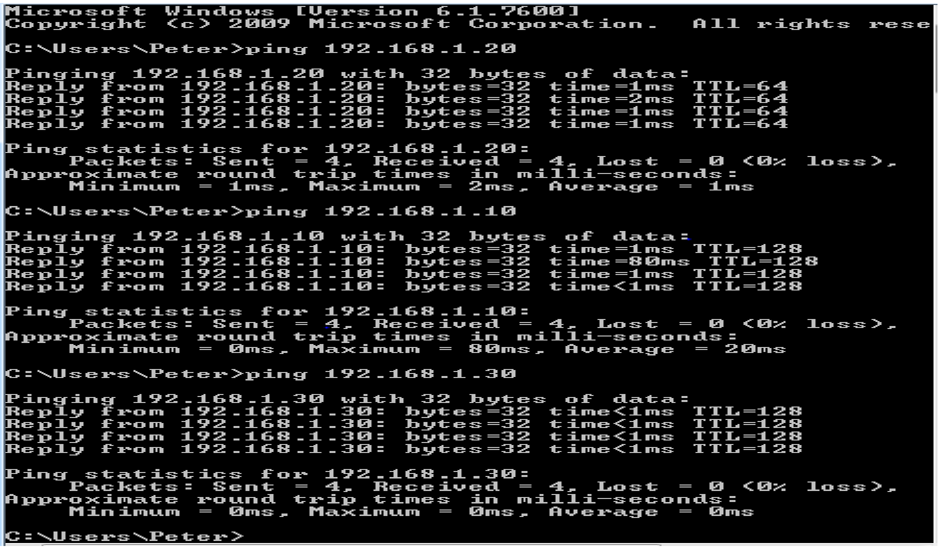
☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: Realtek PCIe FE Family Controller

Figure 6: Bridged Host-Client Connections

To test the effectiveness of these configurations, the internet control message protocol (ICMP) code number 8, called packet internet groper (PING) utility was employed. Figure 7 shows the results of the PING utility against IP address of the guest running in Oracle VirtualBox VM.



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights rese

C:\Users\Peter>ping 192.168.1.20

Pinging 192.168.1.20 with 32 bytes of data:
Reply from 192.168.1.20: bytes=32 time=1ms TTL=64
Reply from 192.168.1.20: bytes=32 time=2ms TTL=64
Reply from 192.168.1.20: bytes=32 time=1ms TTL=64
Reply from 192.168.1.20: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\Users\Peter>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:
Reply from 192.168.1.10: bytes=32 time=1ms TTL=128
Reply from 192.168.1.10: bytes=32 time=80ms TTL=128
Reply from 192.168.1.10: bytes=32 time=1ms TTL=128
Reply from 192.168.1.10: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 80ms, Average = 20ms

C:\Users\Peter>ping 192.168.1.30

Pinging 192.168.1.30 with 32 bytes of data:
Reply from 192.168.1.30: bytes=32 time<1ms TTL=128
Reply from 192.168.1.30: bytes=32 time<1ms TTL=128
Reply from 192.168.1.30: bytes=32 time<1ms TTL=128
Reply from 192.168.1.30: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.30:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Peter>
```

Figure 7: Connectivity Testing

As this figure shows, there was successful communication from the peer instant messaging machine to the virtualized instant messaging guest peer. This meant that all was set for the actual instant message communication between the virtualized guest and the non-virtualized peer.

1V. RESULTS AND DISCUSSION

This section presents and discusses the results from the empirical investigation of Instant Messaging security in a virtualized environment as explained below.

4.1 Client-Server Interface

Each of the instant message communicating entity had the hyperlinks shown in Figure 8. This consisted of the *post Chat*, *View Incoming Chat* and *View Outgoing Chat* hyperlinks. The *Post Chat* link was utilized to send an instant message to the receiver while the *View Incoming Chat* was employed to retrieve the instant message directed towards the user. Finally, the *View Outgoing Chat* link was used to retrieve the instant messages that the user has sent.

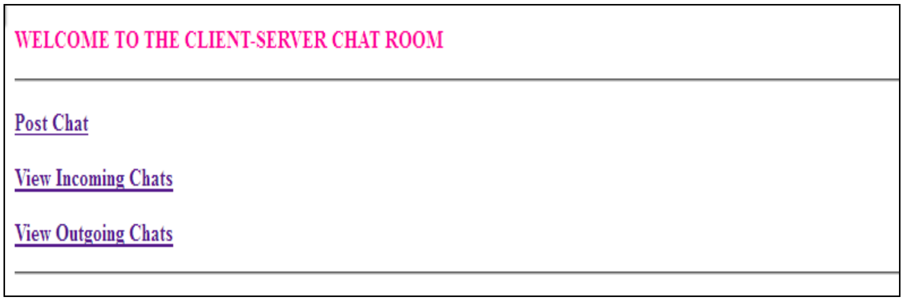
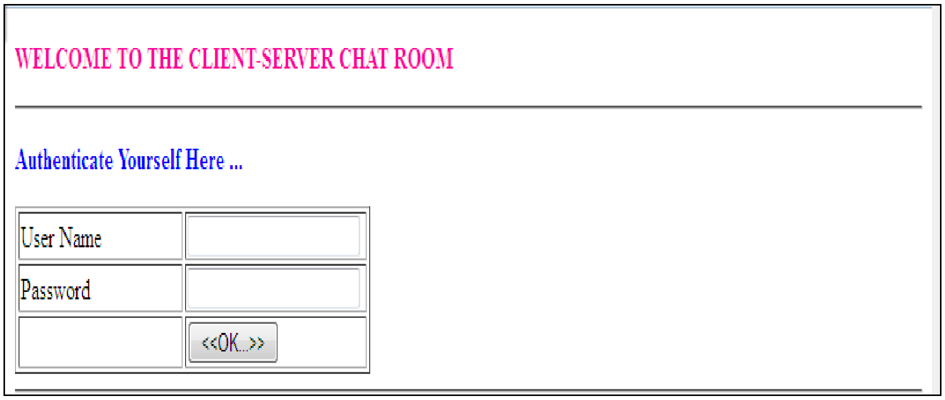


Figure 8: Main Chat Interface

To post a chat, one clicks the *'Post Chat'* link that then load the interface shown in Figure 9. On this interface, the user authenticates himself by entering a valid user name and password. Afterwards, he clicks the *'OK'* button to load the next interface.

4.2. Authentication Interface

The design represent an interface used by the. users to authenticate themselves when they log into the system.



WELCOME TO THE CLIENT-SERVER CHAT ROOM

Authenticate Yourself Here ...

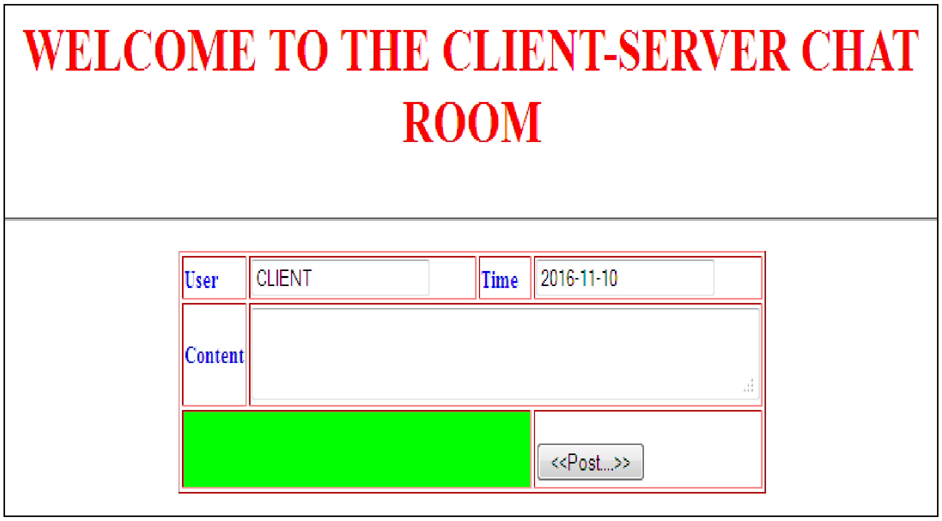
User Name	
Password	

Figure 9: Authentication Interface

The username and password are analogous to the password or the unique pattern that the user draws to unlock the phone so as to access the instant messaging interface. Upon the entry of the correct credentials, the user is presented with the interface shown in Figure 10.

4.3. Composing Interface

In order to compose a message that will be instantly posted from the peers to the server for processing, figure 10 represents the design.



WELCOME TO THE CLIENT-SERVER CHAT ROOM

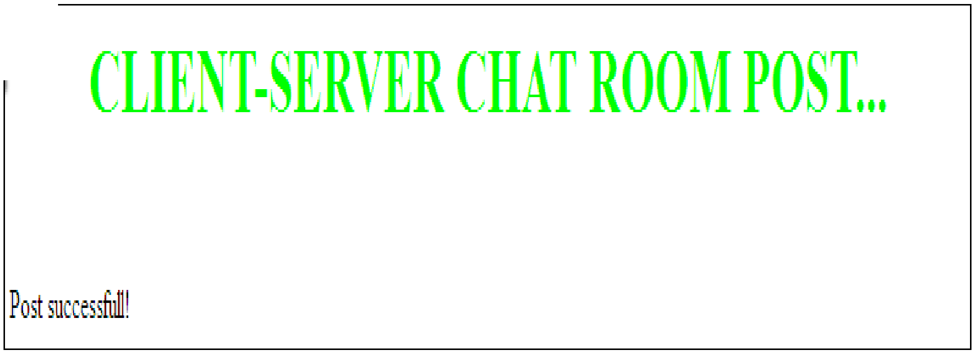
User	CLIENT	Time	2016-11-10
Content			

Figure 10: Chat Posting Interface

In this interface, the user types his instant chat message and clicks the *'OK'* command button. This will in turn load the interface shown in Figure 11.

4.4. Posting Response

This serves to inform the user that indeed the chat he posted has been delivered to the recipient, otherwise an error message will be displayed. To retrieve the incoming instant messages, one simply clicks on *"View Incoming Chats"* link of Figure 8. Upon doing this, the interface on Figure 12 will be displayed.



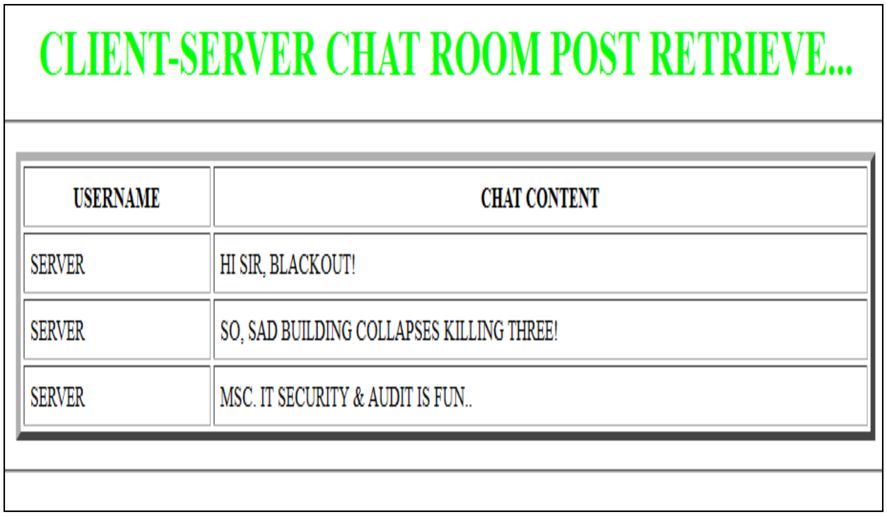
CLIENT-SERVER CHAT ROOM POST...

Post successfull!

Figure 11: Instant Message Posting Successful

4.5. Incoming Message

This process clearly depicts what happens with normal instant chat message retrieval applications.



CLIENT-SERVER CHAT ROOM POST RETRIEVE...

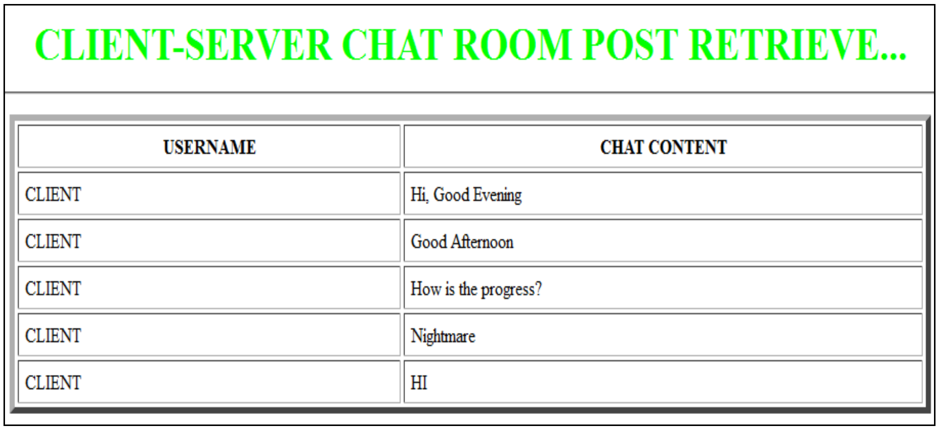
USERNAME	CHAT CONTENT
SERVER	HI SIR, BLACKOUT!
SERVER	SO, SAD BUILDING COLLAPSES KILLING THREE!
SERVER	MSC. IT SECURITY & AUDIT IS FUN..

Figure 12: Incoming Chats

Typically, one clicks on the sender's contact, which then loads the chat interface from which the incoming chats can be read and post messages. To retrieve the chats sent, one clicks the *"View Outgoing Chats"* and this displays the information shown in Figure 13.

4.6. Outgoing Message

The design demonstrate the content of the outgoing messages. On both Figure12 and Figure 13, the name of the sender and receiver are displayed on the *"Username"* column. The actual chat is displayed under the *"Chat Content"* column.



CLIENT-SERVER CHAT ROOM POST RETRIEVE...

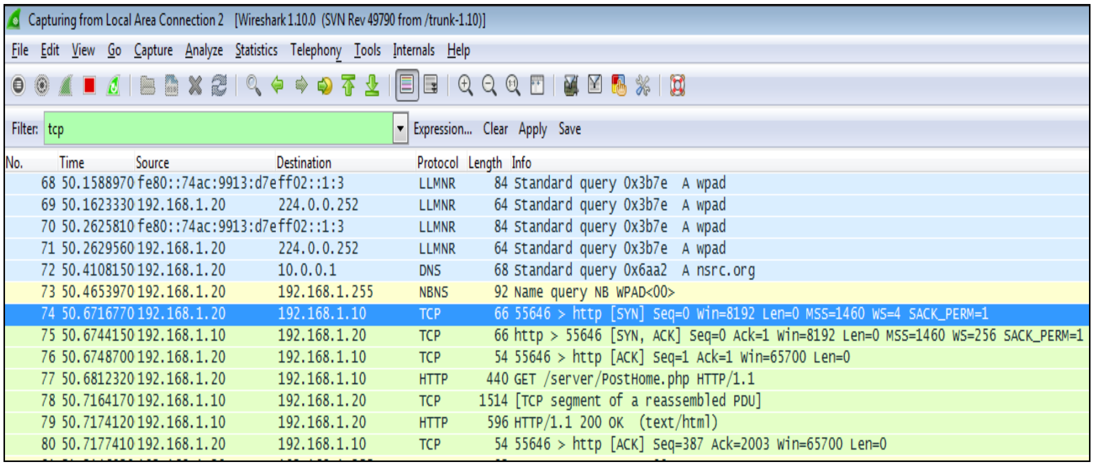
USERNAME	CHAT CONTENT
CLIENT	Hi, Good Evening
CLIENT	Good Afternoon
CLIENT	How is the progress?
CLIENT	Nightmare
CLIENT	HI

Figure 13: Outgoing Chat Retrieval

Clearly, these instant message communications are in plain text, raising the susceptibility to eavesdropping. Further traffic analysis using the Wireshark software was conducted to determine whether the instant message communication can be intercepted. The results obtained are as shown in Figure 14.

4.7 TCP Three Way Handshake communication

The figure illustrates the synchronization process of the device during communication which describes a Three-way handshake. The TCP allows one side to establish a connection as a client and the other side to accept the connection or reject it as a serve. It shows that there was a TCP communication between the peer (IP address 192.168.1.20) and the virtualized guest (IP address 192.168.1.10).


Capturing from Local Area Connection 2   [Wireshark 1.10.0 (SVN Rev 49790 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: tcp

```
No.  Time       Source                 Destination   Protocol Length Info
  68 50.1588970 fe80::74ac:9913:d7eff02::1:3         LLMNR      84 Standard query 0x3b7e  A wpad
  69 50.1623330 192.168.1.20           224.0.0.252   LLMNR      64 Standard query 0x3b7e  A wpad
  70 50.2625810 fe80::74ac:9913:d7eff02::1:3         LLMNR      84 Standard query 0x3b7e  A wpad
  71 50.2629560 192.168.1.20           224.0.0.252   LLMNR      64 Standard query 0x3b7e  A wpad
  72 50.4108150 192.168.1.20           10.0.0.1      DNS        68 Standard query 0x6aa2  A nsrc.org
  73 50.4653970 192.168.1.20           192.168.1.255 NBNS       92 Name query NB WPAD<00>
  74 50.6716770 192.168.1.20           192.168.1.10  TCP        66 55646 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
  75 50.6744150 192.168.1.10           192.168.1.20  TCP        66 http > 55646 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
  76 50.6748700 192.168.1.20           192.168.1.10  TCP        54 55646 > http [ACK] Seq=1 Ack=1 Win=65700 Len=0
  77 50.6812320 192.168.1.20           192.168.1.10  HTTP      440 GET /server/PostHome.php HTTP/1.1
  78 50.7164170 192.168.1.10           192.168.1.20  TCP      1514 [TCP segment of a reassembled PDU]
  79 50.7174120 192.168.1.10           192.168.1.20  HTTP      596 HTTP/1.1 200 OK  (text/html)
  80 50.7177410 192.168.1.20           192.168.1.10  TCP        54 55646 > http [ACK] Seq=387 Ack=2003 Win=65700 Len=0
```

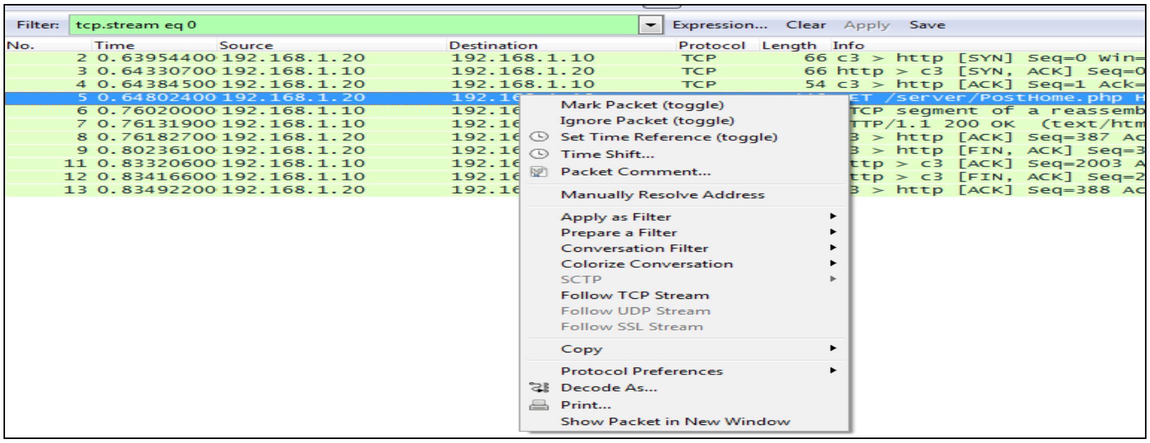
Figure 14: Instant Message Communication Interception

The sequence number of the initial packet was 0. The guest then respondent by acknowledging the request for connection, with ACK=1. The peer and the virtualized guest then proceeded to exchange data using the hypertext transfer protocol (HTTP).

4.8 Transmission Control Protocol Stream

It's a window that details the "request" sent and the "response" received. This process was done for both the server and the client. The idea was to capture the live packet as it was being transmitted from the client to the server and from the server to the client. Since the address of the client was 192.168.1.20 and that of the server was 192.168.1.10, the following communication shown in Figure 16 was intercepted and live capture carried out.

To demonstrate that the instant messages were in plain text in both sending and transmitting devices, the TCP stream was followed as shown in Figure 15.



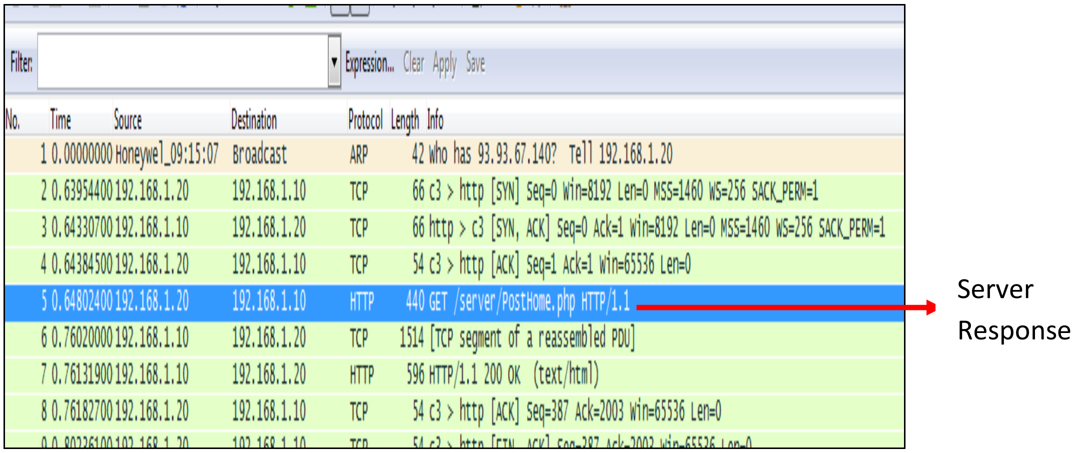
Filter: tcp.stream eq 0

```
No.   Time        Source        Destination   Protocol Length Info
    2 0.63954400 192.168.1.20  192.168.1.10  TCP        66 c3 > http [SYN] Seq=0 Win=
    3 0.64330700 192.168.1.10  192.168.1.20  TCP        66 http > c3 [SYN, ACK] Seq=0
    4 0.64384500 192.168.1.20  192.168.1.10  TCP        54 c3 > http [ACK] Seq=1 Ack=
    5 0.64802400 192.168.1.20  192.16                      ET /server/PostHome.php H
    6 0.76020000 192.168.1.10  192.16                      TCP segment of a reassemb
    7 0.76131900 192.168.1.10  192.16                      TTP/1.1 200 OK  (text/htm
    8 0.76182700 192.168.1.20  192.16                      3 > http [ACK] Seq=387 Ac
    9 0.80236100 192.168.1.20  192.16                      3 > http [FIN, ACK] Seq=3
   11 0.83320600 192.168.1.10  192.16                      ttp > c3 [ACK] Seq=2003 A
   12 0.83416600 192.168.1.10  192.16                      ttp > c3 [FIN, ACK] Seq=2
   13 0.83492200 192.168.1.20  192.16                      3 > http [ACK] Seq=388 Ac
```

Figure 15: TCP Stream Following

4.9 HTTP Client-Sarver Communication

The figure depict response packet capture from the server through hypertext transfer protocol.



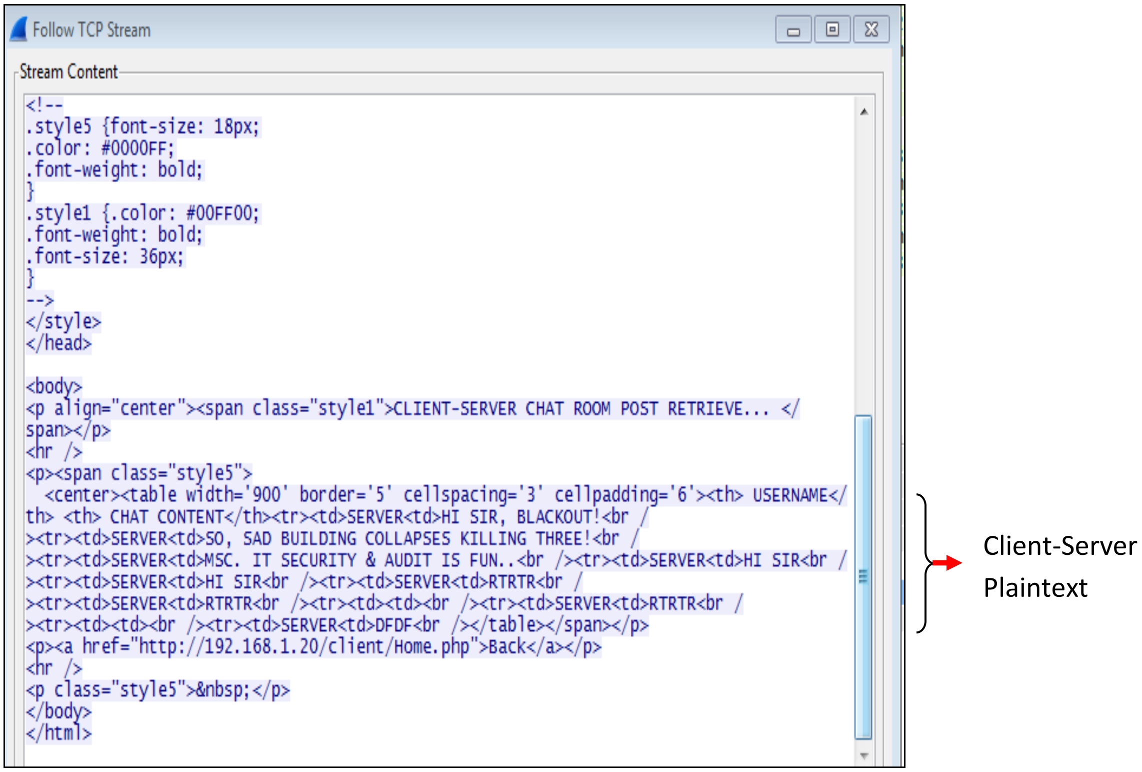
```
No. Time       Source              Destination  Protocol Length Info
  1 0.00000000 Honeywel_09:15:07   Broadcast    ARP      42 Who has 93.93.67.140?  Tell 192.168.1.20
  2 0.63954400 192.168.1.20        192.168.1.10 TCP      66 c3 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
  3 0.64330700 192.168.1.10        192.168.1.20 TCP      66 http > c3 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
  4 0.64384500 192.168.1.20        192.168.1.10 TCP      54 c3 > http [ACK] Seq=1 Ack=1 Win=65536 Len=0
  5 0.64802400 192.168.1.20        192.168.1.10 HTTP    440 GET /server/PostHome.php HTTP/1.1
  6 0.76020000 192.168.1.10        192.168.1.20 TCP    1514 [TCP segment of a reassembled PDU]
  7 0.76131900 192.168.1.10        192.168.1.20 HTTP    596 HTTP/1.1 200 OK  (text/html)
  8 0.76182700 192.168.1.20        192.168.1.10 TCP      54 c3 > http [ACK] Seq=387 Ack=2003 Win=65536 Len=0
  9 0.80236100 192.168.1.20        192.168.1.10 TCP      54 c3 > http [FIN, ACK] Seq=387 Ack=2003 Win=65536 Len=0
```

Figure 16: Client – Server Communication Interception

Upon following this TCP stream, valuable information related to this packet was obtained as demonstrated by Figure 17.

4.10 Plaintext Client-Server Communication

This figure shows the TCP packet in pain text. Note that Figure 17 essentially displays the contents of the table in Figure 12. To do so, it uses the HTML tags for table creation *(<table>)*, table rows *(<tr>)*, table data *(<td>)*.



```
<!--
.style5 {font-size: 18px;
.color: #0000FF;
.font-weight: bold;
}
.style1 {.color: #00FF00;
.font-weight: bold;
.font-size: 36px;
}
-->
</style>
</head>

<body>
<p align="center"><span class="style1">CLIENT-SERVER CHAT ROOM POST RETRIEVE... </
span></p>
<hr />
<p><span class="style5">
  <center><table width='900' border='5' cellspacing='3' cellpadding='6'><th> USERNAME</
th> <th> CHAT CONTENT</th><tr><td>SERVER<td>HI SIR, BLACKOUT!<br /
><tr><td>SERVER<td>SO, SAD BUILDING COLLAPSES KILLING THREE!<br /
><tr><td>SERVER<td>MSC. IT SECURITY & AUDIT IS FUN..<br /><tr><td>SERVER<td>HI SIR<br /
><tr><td>SERVER<td>HI SIR<br /><tr><td>SERVER<td>RTRTR<br /
><tr><td>SERVER<td>RTRTR<br /><tr><td><td><br /><tr><td>SERVER<td>RTRTR<br /
><tr><td><td><br /><tr><td>SERVER<td>DFDF<br /></table></span></p>
<p><a href="http://192.168.1.20/client/Home.php">Back</a></p>
<hr />
<p class="style5">&nbsp;</p>
</body>
</html>
```

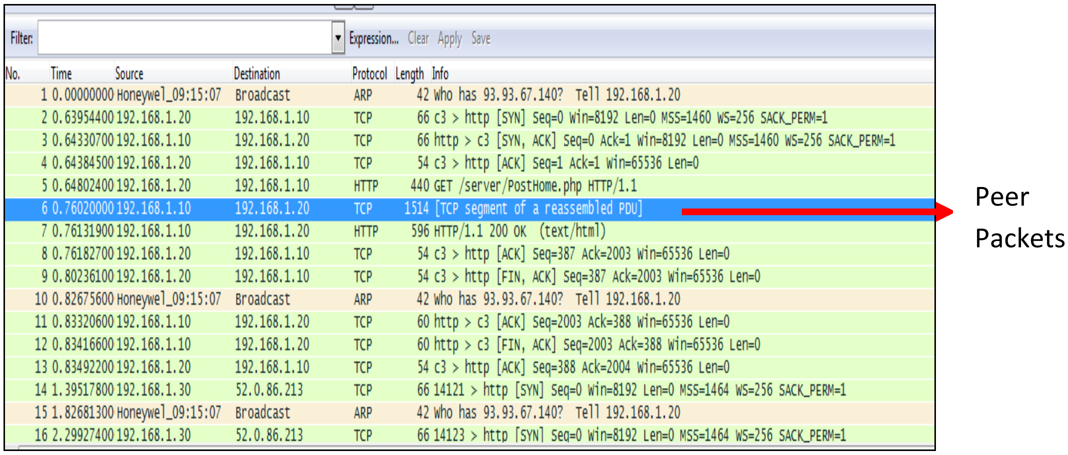
Figure 17: Plaintext Client-Server Communication

Towards the end of this TCP stream, the uniform resource locator (URL) of the machine towards which this request is directed to is given: *http://192.168.1.20/client/home.php* , which was the address of the client machine connecting to the server.

Obviously, this plain text packet reveals enough information that may facilitate further attacks such as spoofing attacks and denial of services (DOS) since the IP addresses and the URLs are evident from this capture. To investigate the server-client communication, the packet in Figure 18 was intercepted and followed.

4.11 Server - Client Communication Interception

To demonstrate the insecure request, the analysis was done from one the peers under domain 192.168.1.20 as shown in figure 18.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	Honeywel_09:15:07	Broadcast	ARP	42	Who has 93.93.67.140? Tell 192.168.1.20
2	0.63954400	192.168.1.20	192.168.1.10	TCP	66	c3 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
3	0.64330700	192.168.1.10	192.168.1.20	TCP	66	http > c3 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
4	0.64384500	192.168.1.20	192.168.1.10	TCP	54	c3 > http [ACK] Seq=1 Ack=1 Win=65536 Len=0
5	0.64802400	192.168.1.20	192.168.1.10	HTTP	440	GET /server/PostHome.php HTTP/1.1
6	0.76020000	192.168.1.10	192.168.1.20	TCP	1514	[TCP segment of a reassembled PDU]
7	0.76131900	192.168.1.10	192.168.1.20	HTTP	596	HTTP/1.1 200 OK (text/html)
8	0.76182700	192.168.1.20	192.168.1.10	TCP	54	c3 > http [ACK] Seq=387 Ack=2003 Win=65536 Len=0
9	0.80236100	192.168.1.20	192.168.1.10	TCP	54	c3 > http [FIN, ACK] Seq=387 Ack=2003 Win=65536 Len=0
10	0.82675600	Honeywel_09:15:07	Broadcast	ARP	42	Who has 93.93.67.140? Tell 192.168.1.20
11	0.83320600	192.168.1.10	192.168.1.20	TCP	60	http > c3 [ACK] Seq=2003 Ack=388 Win=65536 Len=0
12	0.83416600	192.168.1.10	192.168.1.20	TCP	60	http > c3 [FIN, ACK] Seq=2003 Ack=388 Win=65536 Len=0
13	0.83492200	192.168.1.20	192.168.1.10	TCP	54	c3 > http [ACK] Seq=388 Ack=2004 Win=65536 Len=0
14	1.39517800	192.168.1.30	52.0.86.213	TCP	66	14121 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1464 WS=256 SACK_PERM=1
15	1.82681300	Honeywel_09:15:07	Broadcast	ARP	42	Who has 93.93.67.140? Tell 192.168.1.20
16	2.29927400	192.168.1.30	52.0.86.213	TCP	66	14123 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1464 WS=256 SACK_PERM=1

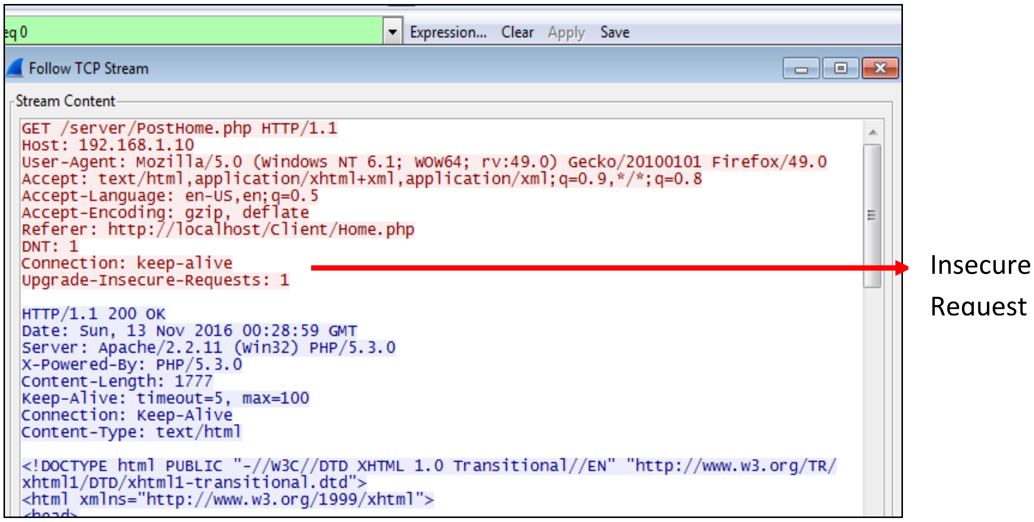
Peer
Packets

Figure 18: Server - Client Communication Interception

Once again, upon TCP stream following of this packet, the information shown in Figure 19 was obtained.

4.12 Plaintext Server-Client Communication

The information obtained showed that the response was using the "GET" form submission method as indicated by the first line. Note that both the client and server request are marked as being *'Insecure-Request'* as demonstrated by line 10, since the request are in plain text.



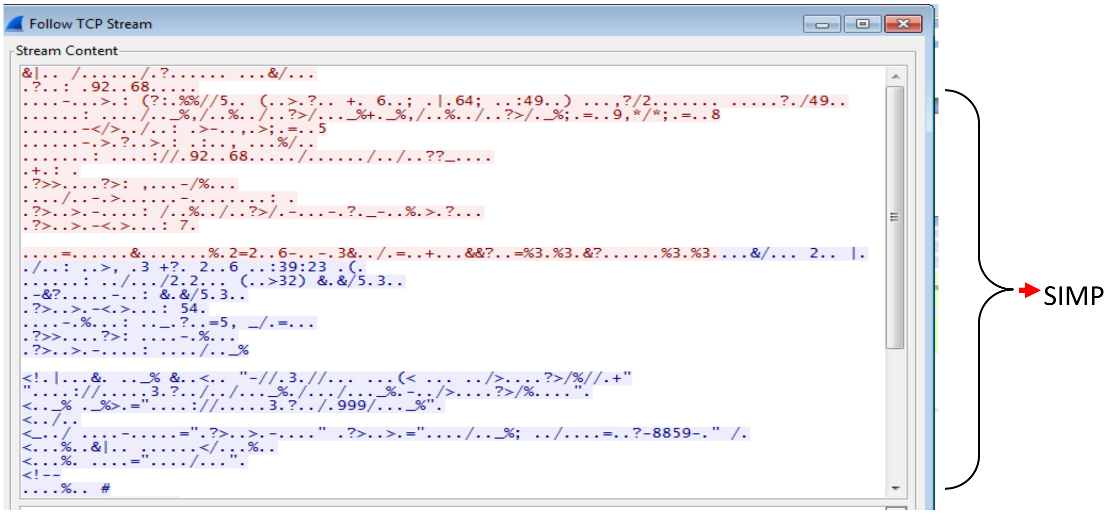
```
GET /server/PostHome.php HTTP/1.1
Host: 192.168.1.10
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/Client/Home.php
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Sun, 13 Nov 2016 00:28:59 GMT
Server: Apache/2.2.11 (Win32) PHP/5.3.0
X-Powered-By: PHP/5.3.0
Content-Length: 1777
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

Insecure
Request

Figure 19: Plaintext Server-Client Communication

4.13 Scrambled Instant Message Packet

The proposed port-based algorithm would scramble the packets such that the interception of this packet makes no sense to the intruder. Figure 20 shows such kind of TCP packet scrambling in a typical instant message.



```
&|.. /....../.?...... ...&/...
.?..: .92..68.....
....-...>.: (?:.%%//5.. (..>.?.. +. 6..; .|.64; ..:49..) ...,?/2....... .....?./49..
......: ..../.._%,/..%../..?>/..._%+._%,/..%../..?>/._%;.=..9,*/*;.=..8
......-</>../..: .>-..,.>;.=..5
......-.>.?..>.: .:.., ...%/..
.......: ....://.92..68..../....../../..??_....
.+.: .
.?>>....?>: ,...-/%...
..../..-.>......-........: .
.?>..>.-....: /..%../..?>/.-...-.?._-..%.>.?...
.?>..>.-<.>...: 7.

....=......&.......%.2=2..6-..-.3&../.=..+...&&?..=%3.%3.&?......%3.%3....&/... 2.. |.
./..: ..>, .3 +?. 2..6 ..:39:23 .(.
......: ../.../2.2... (..>32) &.&/5.3..
.-&?.....-..: &.&/5.3..
.?>..>.-<.>...: 54.
....-.%...: .._.?..=5, _/.=...
.?>>....?>: ....-.%...
.?>..>.-....: ..../.._%

<!.|...&. ..._% &..<.. "-//.3.//... ...(< ... ../>....?>/%//.+"
"....://.....3.?../../..._%./.../..._%.-../>....?>/%....".
<.._% ._%>.="....://.....3.?../.999/..._%".
<../..
<_../ ....-.....=".?>..>.-...." .?>..>.="..../.._%; ../....=..?-8859-." /.
<...%..&|.. ......</...%..
<...%. ....="..../...".
<!--
....%.. #
```

Figure 20: Typical Scrambled Instant Message Packet

This figure confirms the fact that if instant messages could be scrambled, then t its content is meaningless to the human readers. In such a case, a decryption key will be required to turn this data into human readable format.

V. CONCLUSIONS

This paper sought to demonstrate the fact that instant messaging applications have numerous vulnerabilities, one of them being the transmission of messages in plaintext. To illustrate this, a prototype was developed in Java programming language using its networking sockets. The prototype was run in an Oracle VirtualBox VM environment. The results that were obtained demonstrate clearly that the instant messages are in plaintext in both the sending and receiving machines.

Moreover, data capture that was performed using Wireshak further reveals that the instant messages were not immune from interception and remote monitoring. Therefore, the researchers recommend secure protocols, strong authentication and message encryption at both the receiver and sender so that these messages are in human unreadable format in both terminals. In this way, eavesdropping and remote monitoring of the communication can be thwarted.

REFERENCES

1] Weissbrot & Alison (2016). *Car Service APIs Are Everywhere, But What's In It For Partner Apps?* AdExchanger. ad exchanger.

2] Mark (2015). Private, Partner or Public: Which API Strategy Is Best For Business?. Programmable Web.

3] Z. Vinnie & G. Belvin (2012). *Silent circle instant messaging protocol.*

4] S. Lin Z. Hao; X. Tao; L. Mingshu (2016). *An Empirical Study on Evolution of API Documentation.* International Conference on Fundamental Approaches to Software Engineering. Springer Berlin Heidelberg.

5] O. Wendell (2013*). Cisco CCENT/ CCNA ICND1 100-101 Official Cert Guide*. Pearson Education. pp. Ch. 1

6] M. Green (2014). *Noodling about IM protocols.*

7] Andreas & Buchenscheit (2014). *Privacy implications of presence sharing in mobile messaging applications.* Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia. ACM.

8] P. Jagwani (2016). *Analyzing Instant Messaging Applications for Threats : WhatsApp Case Study*. Department of Computer Science, Aryabhatta College, University of Delhi, Delhi (India).

9] T. Frosch C. Mainka, C. Bader, F. Bergsma, J. Schwenk, T. Holz (2016). *How Secure is TextSecure.* Ruhr University Bochum.

10] B. Smith (2013). *UDP vs TCP security*

11] *G.* Fahrnberger (2014). *SIMS: A Comprehensive Approach for a Secure Instant Messaging Sifter.*

International Journal of Computer Applications Technology and Research
Volume 5 –Issue 12, 748-759, 2016, ISSN:-2319–8656

Time Series Forecasting Using Novel Feature Extraction Algorithm and Multilayer Neural Network

Raheleh Rezazadeh
Master student of software Engineering, Ferdows
Islamic Islamic Azad University,
Ferdows, Iran

Hooman Kashanian
Department of Computer science and software Engineering
Islamic Azad University,
Ferdows, Iran

Abstract: Time series forecasting is important because it can often provide the foundation for decision making in a large variety of fields. A tree-ensemble method, referred to as time series forest (TSF), is proposed for time series classification. The approach is based on the concept of data series envelopes and essential attributes generated by a multilayer neural network... These claims are further investigated by applying statistical tests. With the results presented in this article and results from related investigations that are considered as well, we want to support practitioners or scholars in answering the following question: Which measure should be looked at first if accuracy is the most important criterion, if an application is time-critical, or if a compromise is needed? In this paper demonstrated feature extraction by novel method can improvement in time series data forecasting process.

Keyword: time series data, neural network, forecasting

1. INTRODUCTION

Classical statistics and data analysis primarily address items that can be described by a classic variable that takes either a real value (for a quantitative variable) or a category (for a nominal variable). However, observations and estimations in the real world are usually not sufficiently complete to represent classic data exactly. In the stock market, for instance, stock prices have their daily (or weekly, or monthly) bounds and vary in each period (be it a day, week, or month). Representing the variations with snapshot points (e.g., the closing price) only reflects a particular number at a particular time; it does not properly reflect its variability during the period. This problem can be eased if the highest and lowest prices per period are considered, giving rise to interval-valued data. Interval-valued data is a particular case of symbolic data in the field of symbolic data analysis (SDA) [14]. SDA states that symbolic variables (lists, intervals, frequency distributions, etc.) are better suited than single-valued variables for describing complex real-life situations [2]. It should be noted that interval-valued data in the field of SDA does not come from noise assumptions but rather from the expression of variation or aggregation of huge databases into a reduced number of groups [22]. When considering a chronological sequence of interval-valued data, interval time series (ITS) arises quite naturally. Modeling and forecasting of ITS has the advantage of taking into account the variability and/or uncertainty, and it reduces Therefore, tools for ITS forecasting are very much in demand. According to the existing literature, the main methodologies available for ITS forecasting fall roughly into two categories in relation to the method in which interval data is handled, i.e., splitting single-valued methods or interval-valued methods. For the first category, the lower and upper bounds of interval data are treated as two independent single-valued parts, such as the autoregressive integrated moving average (ARIMA) employed in [3]. For the second category, the lower and upper bounds of interval data are treated using interval arithmetic as interval-valued data, as in the interval Holt's exponential smoothing method (HoltI) [22], the vector auto regression/vector error correction model [11,16], multilayer perceptron (MLP) [22], and interval MLP (IMLP) [21]. Interested readers are referred to [1] for a recent survey of the presented methodologies and techniques employed for ITS forecasting. In this study, we propose to take the form of complex numbers to represent interval data, i.e., by denoting the lower and upper bounds of the interval as real and imaginary parts of a complex number, respectively, thus allowing us to use the complex-valued neural network (CVNN) for ITS prediction. The CVNN is a type of neural network in which weights, threshold values, and input and output signals are all complex numbers. The activation function as well as its derivatives have to be ''well behaved'' everywhere in the complex plane [20]. CVNNs exhibit very desirable characteristics in their learning, self-organizing, and processing dynamics. This, together with the widespread use of analytic signals, gives them a significant advantage in practical applications in diverse fields of engineering, where signals are routinely analyzed and processed in time/space, frequency, and phase domains. A significant number of studies have demonstrated that CVNNs have better capabilities than real-valued neural networks for function approximation [17,20] and classification tasks [16,17]. Due to the localization ability and simple architecture of radial basis function (RBF) neural networks in the real domain, the complex-valued RBF neural network is gaining interest among researchers. Notable earlier work includes that of Chen et al. [9], who investigated a complex-valued RBF neural network with complex-valued weights and a real-valued activation function using several learning algorithms. Other related studies can be found in Jianping et al. [20] and Deng et al. [13]. Complex-valued RBF neural networks typically use a Gaussian activation function that maps complex-valued inputs to a real-valued hyper-dimensional feature space at the hidden layer. However, as the mapping is done at the hidden layer, the input is not efficiently transmitted to the output [15], which results in inaccurate phase approximation [12]. To overcome the limitations, researchers have started to develop fully complex-valued regression methods (or classifiers) for solving real valued function approximation (or classification) problems. Recently, a fully complex-valued RBF neural network (FCRBFNN) using a hyperbolic secant function as the activation function was derived by Savitha et al. [7]. Their experimental study clearly showed that the FCRBFNN can outperform other complex-valued RBF networks from the literature for function approximation [4]. In view of the FCRBFNN's advantages in processing complex-valued signals, it will be interesting to investigate the possibility of forecasting the lower and upper bounds of ITS in the form of complex intervals using the FCRBFNN. Another issue considered in this study is the evolution of structure (or topology) and parameters (e.g., scaling factors and weights) of the FCRBFNN. In general, the learning steps of a neural network are as follows. First, a network structure is determined with a predefined number of inputs, hidden nodes, and outputs. Second, an algorithm is chosen to realize the learning process. In [30], for instance, the number of hidden nodes was first determined by the K-means clustering algorithm, and then a fully complex-valued gradient descent learning algorithm was used to tune the FCRBFNN. Since the gradient descent algorithm, may get stuck in local optima and is highly dependent on the starting points, research efforts have been made on using evolutionary computation methods to design and evolve neural networks [10,12,11,9,3]. Following this line of research, in this study, we use multilayer neural network (MNN) [21].

2. Time Series Data Models and Forecasting

Time series forecasting, or time series prediction, takes an existing series of data $x_{t-n}, \ldots, x_{t-2}, x_{t-1}, x_t$ and forecasts the $x_{t+1}, x_{t+2}, \ldots$ data values. The goal is to observe or model the existing data series to enable future unknown data values to be forecasted accurately. Examples of data series include financial data series (stocks, indices, rates, etc.), physically observed data series (sunspots, weather, etc.), and mathematical data series (Fibonacci sequence, integrals of differential equations, etc.). The phrase “time series” generically refers to any data series, whether or not the data are dependent on a certain time increment. Throughout the literature, many techniques have been implemented to perform time series forecasting. This paper will focus on two techniques: neural networks and k-nearest-neighbor. This paper will attempt to fill a gap in the abundant neural network time series forecasting literature, where testing arbitrary neural networks on arbitrarily complex data series is common, but not very enlightening. This paper thoroughly analyzes the responses of specific neural network configurations to artificial data series, where each data series has a specific characteristic. A better understanding of what causes the basic neural network to become an inadequate forecasting technique will be gained. In addition, the influence of data preprocessing will be noted. The forecasting performance of k-nearest-neighbor, which is a much simpler forecasting technique, will be compared to the neural networks’ performance. Finally, both techniques will be used to forecast a real data series. Time series Models and forecasting methods have been studied by various people and detailed analysis can be found in [**Error! Reference source not found.**, **Error! Reference source not found.**,**Error! Reference source not found.**]. Time Series Models can be divided into two kinds. Univariate Models where the observations are those of single variable recorded sequentially over equal spaced time intervals. The other kind is the Multivariate, where the observations are of multiple variables. A common assumption in many time series techniques is that the data are stationary. A stationary process has the property that the mean, variance and autocorrelation structure do not change over time. Stationarity can be defined in precise mathematical terms, but for our purpose we mean a flat looking series, without trend, constant variance over time, a constant autocorrelation structure over time and no periodic fluctuations. There are a number of approaches to modeling time series. We outline a few of the most common approaches below. Trend, Seasonal, Residual Decompositions: One approach is to decompose the time series into a trend, seasonal, and residual component. Triple exponential smoothing is an example of this approach. Another example, called seasonal loess, is based on locally weighted least squares. Frequency Based Methods: Another approach, commonly used in scientific and engineering applications, is to analyze the series in the frequency domain. An example of this approach in modeling a sinusoidal type data set is shown in the beam deflection case study. The spectral plot is the primary tool for the frequency analysis of time series.

Autoregressive (AR) Models: A common approach for modeling univariate time series is the autoregressive (AR) model equation (1):

$$X_t = \delta + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + A_t \quad (1)$$

where Xt is the time series, at is white noise, and

$$\delta = \left(1 - \sum_{i=1}^{p} \phi_i\right)\mu \quad (2)$$

with μ denoting the process mean.

An autoregressive model is simply a linear regression of the current value of the series against one or more prior values of the series. The value of p is called the order of the AR model. AR models can be analyzed with one of various methods; including standard linear least squares techniques. They also have a straightforward interpretation. Moving Average (MA): Models another common approach for modeling univariate time series models is the moving average (MA) model:

$$X_t = \mu + A_t - \theta_1 A_{t-1} - \theta_2 A_{t-2} - \ldots - \theta_q A_{t-q} \quad (3)$$

where Xt is the time series, μ is the mean of the series, $\mathbf{A_{t-i}}$ are white noise, and 1, ... , q are the parameters of the model. The value of q is called the order of the MA model.

That is, a moving average model is conceptually a linear regression of the current value of the series against the white noise or random shocks of one or more prior values of the series. The random shocks at each point are assumed to come from the same distribution, typically a normal distribution, with location at zero and constant scale. The distinction in this model is that these random shocks are propagated to future values of the time series. Fitting the MA estimates is more complicated than with AR models because the error terms are not observable. This means that iterative non-linear fitting procedures need to be used in place of linear least squares. MA models also have a less obvious interpretation than AR models. Note, however, that the error terms after the model is fit should be independent and follow the standard assumptions for a univariate process.

Box-Jenkins Approach: The Box-Jenkins ARMA model is a combination of the AR and MA models:

$$\mathrm{X}_t = \delta + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \\ A_t - \theta_1 A_{t-1} - \theta_2 A_{t-2} - \ldots - \theta_q A_{t-q} \qquad (4)$$

where the terms in the equation have the same meaning as given for the AR and MA model [**Error! Reference source not found.**].

The Box-Jenkins model assumes that the time series is stationary. Box and Jenkins recommend differencing non-stationary series one or more times to achieve stationarity. Doing so produces an ARIMA model, with the "I" standing for "Integrated". Some formulations transform the series by subtracting the mean of the series from each data point. This yields a series with a mean of zero. Whether you need to do this or not is dependent on the software you use to estimate the model. Box-Jenkins models can be extended to include seasonal autoregressive and seasonal moving average terms. Although this complicates the notation and mathematics of the model, the underlying concepts for seasonal autoregressive and seasonal moving average terms are similar to the non-seasonal autoregressive and moving average terms. The most general Box-Jenkins model includes difference operators, autoregressive terms, moving average terms, seasonal difference operators, seasonal autoregressive terms, and seasonal moving average terms. As with modeling in general, however, only necessary terms should be included in the model.

2.1 Steps in the Time Series Forecasting Process:

The goal of a time series forecast is to identify factors that can be predicted. This is a systematic approach involving the following steps and show in Figure (1).

Step 1: Hypothesize a form for the time series model.

Identify which of the time series components should be included in the model.

Perform the following operations.

Collect historical data.

Graph the data vs. time.

Hypothesize a form for the time series model.

Verify this hypothesis statistically.

Step 2: Select a forecasting technique. A forecasting technique must be chosen to predict future values of the time series.

The values of input parameters must be determined before the technique can be applied.

Step 3: Prepare a forecast.

The appropriate data values must be substituted into the selected forecasting model.

The forecast may be affected by Number of past observations used.

Initial forecast value used.

The following flowchart highlights the systematic development of the modeling and forecasting phases:

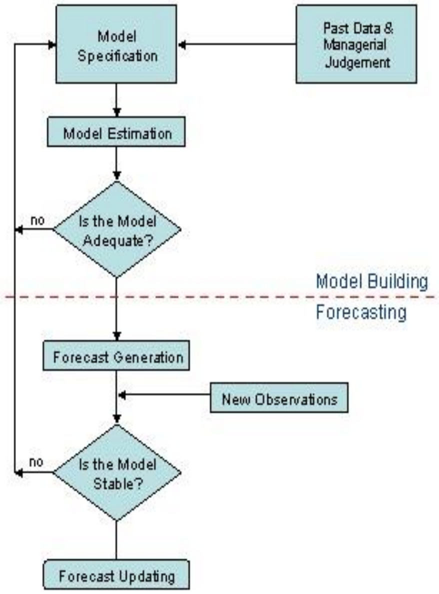


Figure (1): time series forecasting architecture

Stationary Forecasting Models: In a stationary model the mean value of the time series is assumed to be constant.

2.2 Feature Extraction Algorithm

The proposed feature extraction scheme processes the magnetic, angular rate, and accelerometer signals provided by the MARG sensors in order to excerpt 1. the orientation of the person w.r.t. the earth frame, and 2. the acceleration in the person frame, P a. In contrast to other feature extraction schemes [4, 7], we consider that angular rate measurements provided by gyroscopes are not valuable signals any longer for the classification algorithms, since their information is incorporated to the orientation of the person. Therefore, the main goal consists in computing P E^q, i.e., the orientation of the earth frame (E) relative to the person frame (P). The proposed algorithm makes use of quaternion property 2., decomposing the estimation of P E^q as a concatenation of the estimation of the orientation of Ez w.r.t. to P z, P E^qz, followed by the estimation of the orientation of the plane Exy w.r.t. the plane P xy, P E^qxy, i.e.,

$$ {}^{P}_{E}\wedge q = {}^{P}_{E}\wedge q_{xy} \otimes {}^{P}_{E}\wedge q_{z} \qquad (5)$$

where${}^{P}_{E}\wedge q_{z}$ is also decomposed as

$$ {}^{P}_{E}\wedge q_{z} = {}^{S}_{E}\wedge q \otimes {}^{P}_{S}\wedge q_{z} \qquad (6)$$

Algorithm 1 summarizes the process to compute P E^q[n], the orientation of the earth frame w.r.t. the person frame. The calculation is performed for the N available samples of magnetic field, angular rate, and acceleration measurements acquired by the MARG sensor. Note that _, the key parameter of the sensor orientation algorithm [6] must be selected at the beginning, and it plays a key role in the performance of the classification algorithm.

Algorithm 1 Pseudocode of person orientation algorithm

```
Select β
for n = 1: N do
    Compute $^{S}_{E}{}^\wedge q[n]$ with the algorithm of [6] and β
    Detect whether the person is walking
    if walking then
        Update $^{P}_{S}{}^\wedge q\ _{Z}[n]$
        Update $^{P}_{E}{}^\wedge q\ _{XY}[n]$
    else
        $^{P}_{S}{}^\wedge q\ _{Z}[n] = {}^{P}_{S}{}^\wedge q\ _{Z}[n-1]$
        $^{P}_{E}{}^\wedge q\ _{XY}[n] = {}^{P}_{E}{}^\wedge q\ _{XY}[n-1]$
    end if
    $^{P}_{E}{}^\wedge q[n] = {}^{P}_{E}{}^\wedge q\ _{XY}[n] \otimes {}^{S}_{E}{}^\wedge q[n] \otimes {}^{P}_{S}{}^\wedge q\ _{Z}[n]$
end for
```

2.3
2.4
2.5

Table 1Beginning parameters for heuristically trained neural networks.

Heuristic Algorithm Training **Update Frequency = 50, Change Frequency = 10, Decrement = 0.05**						
Architecture	**Learning Rate**	**Epochs Limit**	**Error Limit**	**Data Series** **O = original** **L = less noisy** **M = more noisy** **A = ascending**	**Training Set Data Point Range (# of Examples)**	**Validation Set Data Point Range (# of Examples)**
35:20:1	0.3	500,000	1x10-10	O, L, M, A	0 – 143 (109)	144 – 215 (37)
35:10:1	0.3	500,000	1x10-10	O, L, M, A	0 – 143 (109)	144 – 215 (37)
35:2:1	0.3	500,000	1x10-10	O, L, M, A	0 – 143 (109)	144 – 215 (37)
25:20:1	0.3	250,000	1x10-10	O	0 – 143 (119)	144 – 215 (47)
25:10:1	0.3	250,000	1x10-10	O	0 – 143 (119)	144 – 215 (47)

Table 2 beginning parameter

Simple Method Training						
Architecture	**Learning Rate**	**Epochs Limit**	**Error Limit**	**Data Series** **O = original** **L = less noisy** **M = more noisy** **A = ascending** **S = sunspots**	**Training Set Data Point Range (# of Examples)**	**Validation Set Data Point Range (# of Examples)**
35:20:1	0.1	100,000	1x10-10	O, L, M, A	0 – 143 (109)	144 – 215 (37)
35:10:1	0.1	100,000	1x10-10	O, L, M, A	0 – 143 (109)	144 – 215 (37)
35:2:1	0.1	100,000	1x10-10	O, L, M, A	0 – 143 (109)	144 – 215 (37)
30:30:1	0.05	100,000	1x10-10	S	0 – 165 (136)	-

Often, two trained networks with identical beginning parameters can produce drastically different forecasts. This is because each network is initialized with random weights and biases prior to training, and during training, networks converge to different minima on the training error curve. Therefore, three candidates of each network configuration were trained. This allows another neural network forecasting technique to be used: forecasting by committee. In forecasting by committee, the forecasts from the three networks are averaged together to produce a new forecast. This may either smooth out noisy forecasts or introduce error from a poorly performing network. The coefficient of determination will be calculated for the three networks' forecasts and for the committee forecast to determine the best candidate.

3. Multi Neural Network Parameters and Procedure Architectures

To test how feed-forward neural networks respond to various data series, a network architecture that could accurately learn and model the original series is needed. An understanding of the feed-forward neural network is necessary to specify the number of network inputs. The trained network acts as a function: given a set of inputs it calculates an output. The network does not have any concept of temporal position and cannot distinguish between identical sets of inputs with different actual outputs. For example, referencing the original data series, if there are ten network inputs, among the training examples (assuming the training set covers an entire period) there are several instances where two examples have equivalent **input** vectors but different **output** vectors. One such instance is shown in Figure . This may "confuse" the network during training, and fed that **input** vector during forecasting, the network's output may be somewhere in-between the **output** vectors' values or simply "way off"!

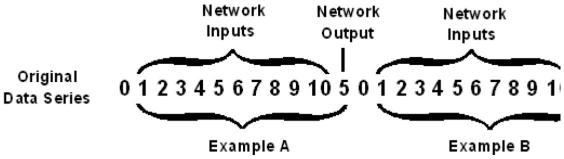


Figure 2 One instance in the original data series where two examples have equivalent **input** vectors but different **output** vectors.

Inspection of the original data series reveals that a network with at least twenty-four inputs is required to make unambiguous examples. (Given twenty-three inputs, the largest ambiguous example **input** vectors would be from zero-based data point 10 to 32 and from 34 to 56.) Curiously, considerably more network inputs are required to make good forecasts, as will be seen in Section **Error! Reference source not found.**.

Next, through trial-and-error the number of hidden layers was found to be one and the number of units in that layer was found to be ten for the artificial data series. The number of output layer units is necessarily one. This network, called 35:10:1 for shorthand, showed excellent forecasting performance on the original data series, and was selected to be the reference for comparison and evaluation. To highlight any effects of having a too-small or too-large network for the task, two other networks, 35:2:1 and 35:20:1, respectively, are also included in the evaluation. Also, addressing the curiosity raised in the previous paragraph, two more networks, 25:10:1 and 25:20:1, are included.

It is much more difficult to choose the appropriate number of network inputs for the sunspots data series. By inspecting the data series and through trial-and-error, thirty network inputs were selected. Also through trial-and-error, one hidden layer with thirty units was selected. Although this network seems excessively large, it proved to be the best neural network forecaster for the sunspots data series. Other networks that were tried include 10:10:1, 20:10:1, 30:10:1, 10:20:1, 20:20:1, and 30:20:1.

4. Training

By observing neural network training characteristics, a heuristic algorithm was developed and implemented in FORECASTER. The parameters for the heuristic are set within the Training Parameters dialog (see Section **Error! Reference source not found.**). The heuristic requires the user to set the learning rate and epochs limit to higher-than-normal values (e.g., 0.3 and 500,000, respectively) and the error limit to a lower-than-normal value (e.g., 1×10^{-10}). The heuristic also uses three additional user-set parameters: the number of training epochs before an application window (view) update (update frequency), the number of updates before a learning rate change (change frequency), and a learning rate change decrement (decrement). Finally, the heuristic requires the data series to be partitioned into a training set and validation set. Given these users set parameters, the heuristic algorithm is:

```
for each view-update during training
    if the validation error is higher than the lowest value seen
        increment count
        if count equals change-frequency
            if the learning rate minus decrement is greater than zero
                lower the learning rate by decrement
                reset count
                continue
            else
                stop training
```

The purpose of the heuristic is to start with an aggressive learning rate, which will quickly find a coarse solution, and then to gradually decrease the learning rate to find a finer solution. Of course, this could be done manually by observing the validation error and using the Change Training Parameters dialog to alter the learning rate. But an automated solution is preferred, especially for an empirical evaluation.

In the evaluation, the heuristic algorithm is compared to the “simple” method of training where training continues until either the number of epochs grows to the epochs limit or the total squared error drops to the error limit. Networks trained with the heuristic algorithm are termed “heuristically trained”; networks trained with the simple method are termed “simply trained”.

Finally, the data series in Section **Error! Reference source not found.** are partitioned so that the training set is the first two periods and the validation set is the third period. Note that “period” is used loosely for less noisy, noisier, and ascending, since they are not strictly periodic.

5. Simply Trained Neural Networks with Thirty-Five Inputs

Figure graphically shows the one-period forecasting accuracy for the best candidates for simply trained 35:2:1, 35:10:1, and 35:20:1 networks. Figure (a), (b), and (c) show forecasts from networks trained on the original, less noisy, and more noisy data series, respectively, and the forecasts are compared to a period of the original data series. Figure (d) shows forecasts from networks trained on the ascending data series, and the forecasts are compared to a fourth "period" of the ascending series. In Figure (c) the 35:2:1 network is not included.

Figure graphically compares metrics for the best candidates for simply trained 35:2:1, 35:10:1, and 35:20:1 networks. Figure (a) and (b) compare the total squared error and unscaled error, respectively, and (c) compares the coefficient of determination. The vertical axis in (a) and (b) is logarithmic. The number of epochs and training time are not included in Figure because all networks were trained to 100,000 epochs.

Finally, Table lists the raw data for the charts in Figure and other parameters used in training. Refer to **Error! Reference source not found.** for more parameters.

In this case, it seems the heuristic was inappropriate. Notice that the heuristic allowed the 35:2:1 network to train for 28,850 epochs more than the simple method. Also, the total squared error and unscaled error for the heuristically trained network were lower, but so was the coefficient of determination it was much lower. Again, the forecasts for the 35:10:1 and 35:20:1 networks are near perfect, and are indistinguishable from the original data series.

In Figure (b) the 35:2:1 network forecast is worse than in **Error! Reference source not found.** (b), whereas the 35:10:1 and 35:20:1 forecasts are about the same as before. Notice that the 35:10:1 forecast is from the committee, but does not appear to smooth the data series' sharp transitions.

In Figure (c), the 35:2:1 network is not included because of its poor forecasting performance on the more noisy data series. The 35:10:1 and 35:20:1 forecasts are slightly worse than before.

In Figure (d), the 35:2:1 network is included and its coefficient of determination is much improved from before. The 35:10:1 and 35:20:1 network forecasts are decent, despite the low coefficient of determination for 35:10:1. The forecasts appear to be shifted up when compared to those in **Error! Reference source not found.** (d).

Finally, by evaluating the charts in Figure and data in Table some observations can be made:

The total squared error and unscaled error are higher for noisy data series with the exception of the 35:10:1 network trained on the noisier data series. It trained to extremely low errors, orders of magnitude lower than with the heuristic, but its coefficient of determination is also lower. This is probably an indication of overfitting the noisier data series with simple training, which hurt its forecasting performance.

The errors do not appear to correlate well with the coefficient of determination.

In most cases, the committee forecast is worse than the best candidate's forecast.

There are four networks whose coefficient of determination is negative, compared with two for the heuristic training method.

(a)

(b)

(c)

(d)

Figure 3 The one-period forecasting accuracy for the best candidates for simply trained 35:2:1, 35:10:1, and 35:20:1 networks trained on the (a) original, (b) less noisy, (c) more noisy, and (d) ascending data series.

(a) (b)

(c)

Figure 4 Graphical comparison of metrics for the networks in Figure : (a) total squared error, (b) unscaled error, and (c) coefficient of determination. Note that the vertical axis is logarithmic for (a) and (b). The raw data are given in Table .

Table 3 Parameters and metrics for the networks in Figure .

Arch.	**Candidate**	**Data Series**	**Ending Learnin g Rate**	**Epochs**	**Total Squared Error**	**Unscaled Error**	**Coeff. of Determ. (1 period)**	**Training Time (sec.)**
35:2:1	b	Original	0.1	100000	0.0035864	14.6705	0.9111	99
35:10:1	c	Original	0.1	100000	3.02316e-006	0.386773	1.0000	286
35:20:1	a	Original	0.1	100000	2.15442e-006	0.376312	1.0000	515
35:2:1	b	Less Noisy	0.1	100000	0.0822801	84.8237	0.5201	99
35:10:1	committee (b)	Less Noisy	0.1	100000	0.00341762	17.6535	0.9173	287
35:20:1	b	Less Noisy	0.1	100000	0.00128001	10.8401	0.9453	531
35:2:1	b	More Noisy	0.1	100000	0.360209	203.893	-4.6748	100
35:10:1	a	More Noisy	0.1	100000	2.47609e-009	0.0166514	-0.0056	282
35:20:1	c	More Noisy	0.1	100000	0.000106478	3.65673	-1.5032	519
35:2:1	a	Ascending	0.1	100000	0.023301	59.7174	0.4091	98
35:10:1	a	Ascending	0.1	100000	0.000421792	8.67945	-0.3585	280
35:20:1	b	Ascending	0.1	100000	0.000191954	5.87395	0.4154	529

and **Error! Reference source not found.** list beginning parameters for all neural networks trained with the heuristic algorithm and simple method, respectively. Parameters for trained networks (e.g., the actual number of training epochs) are presented in Section **Error! Reference source not found.** and Section **Error! Reference source not found.**.

Often, two trained networks with identical beginning parameters can produce drastically different forecasts. This is because each network is initialized with random weights and biases prior to training, and during training, networks converge to different minima on the training error curve. Therefore, three candidates of each network configuration were trained. This allows another neural network forecasting technique to be used: forecasting by committee. In forecasting by committee, the forecasts from the three networks are averaged together to produce a new forecast. This may either smooth out noisy forecasts or introduce error from a poorly performing network. The coefficient of determination will be calculated for the three networks' forecasts and for the committee forecast to determine the best candidate.

Table 1 Beginning parameters for heuristically trained neural networks.

Heuristic Algorithm Training **Update Frequency = 50, Change Frequency = 10, Decrement = 0.05**						
Architecture	**Learning Rate**	**Epochs Limit**	**Error Limit**	**Data Series** **O = original** **L = less noisy** **M = more noisy** **A = ascending**	**Training Set Data Point Range (# of Examples)**	**Validation Set Data Point Range (# of Examples)**
35:20:1	0.3	500,000	1x10^{-10}	O, L, M, A	0 – 143 (109)	144 – 215 (37)
35:10:1	0.3	500,000	1x10^{-10}	O, L, M, A	0 – 143 (109)	144 – 215 (37)
35:2:1	0.3	500,000	1x10^{-10}	O, L, M, A	0 – 143 (109)	144 – 215 (37)
25:20:1	0.3	250,000	1x10^{-10}	O	0 – 143 (119)	144 – 215 (47)
25:10:1	0.3	250,000	1x10^{-10}	O	0 – 143 (119)	144 – 215 (47)

Table 2 beginning parameter

Simple Method Training						
Architecture	**Learning Rate**	**Epochs Limit**	**Error Limit**	**Data Series** **O = original** **L = less noisy** **M = more noisy** **A = ascending** **S = sunspots**	**Training Set Data Point Range (# of Examples)**	**Validation Set Data Point Range (# of Examples)**
35:20:1	0.1	100,000	1x10^{-10}	O, L, M, A	0 – 143 (109)	144 – 215 (37)
35:10:1	0.1	100,000	1x10^{-10}	O, L, M, A	0 – 143 (109)	144 – 215 (37)
35:2:1	0.1	100,000	1x10^{-10}	O, L, M, A	0 – 143 (109)	144 – 215 (37)
30:30:1	0.05	100,000	1x10^{-10}	S	0 – 165 (136)	-

Often, two trained networks with identical beginning parameters can produce drastically different forecasts. This is because each network is initialized with random weights and biases prior to training, and during training, networks converge to different minima on the training error curve. Therefore, three

candidates of each network configuration were trained. This allows another neural network forecasting technique to be used: forecasting by committee. In forecasting by committee, the forecasts from the three networks are averaged together to produce a new forecast. This may either smooth out noisy forecasts or introduce error from a poorly performing network. The coefficient of determination will be calculated for the three networks' forecasts and for the committee forecast to determine the best candidate.

6. Multi Neural Network Parameters and Procedure Architectures

To test how feed-forward neural networks respond to various data series, a network architecture that could accurately learn and model the original series is needed. An understanding of the feed-forward neural network is necessary to specify the number of network inputs. The trained network acts as a function: given a set of inputs it calculates an output. The network does not have any concept of temporal position and cannot distinguish between identical sets of inputs with different actual outputs. For example, referencing the original data series, if there are ten network inputs, among the training examples (assuming the training set covers an entire period) there are several instances where two examples have equivalent **input** vectors but different **output** vectors. One such instance is shown in Figure . This may "confuse" the network during training, and fed that **input** vector during forecasting, the network's output may be somewhere in-between the **output** vectors' values or simply "way off"!

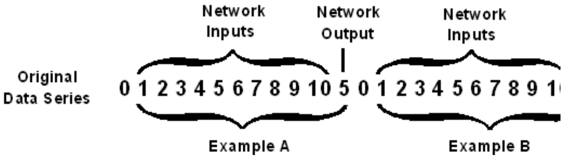


Figure 2 One instance in the original data series where two examples have equivalent **input** vectors but different **output** vectors.

Inspection of the original data series reveals that a network with at least twenty-four inputs is required to make unambiguous examples. (Given twenty-three inputs, the largest ambiguous example **input** vectors would be from zero-based data point 10 to 32 and from 34 to 56.) Curiously, considerably more network inputs are required to make good forecasts, as will be seen in Section **Error! Reference source not found.**.

Next, through trial-and-error the number of hidden layers was found to be one and the number of units in that layer was found to be ten for the artificial data series. The number of output layer units is necessarily one. This network, called 35:10:1 for shorthand, showed excellent forecasting performance on the original data series, and was selected to be the reference for comparison and evaluation. To highlight any effects of having a too-small or too-large network for the task, two other networks, 35:2:1 and 35:20:1, respectively, are also included in the evaluation. Also, addressing the curiosity raised in the previous paragraph, two more networks, 25:10:1 and 25:20:1, are included.

It is much more difficult to choose the appropriate number of network inputs for the sunspots data series. By inspecting the data series and through trial-and-error, thirty network inputs were selected. Also through trial-and-error, one hidden layer with thirty units was selected. Although this network seems excessively large, it proved to be the best neural network forecaster for the sunspots data series. Other networks that were tried include 10:10:1, 20:10:1, 30:10:1, 10:20:1, 20:20:1, and 30:20:1.

7. Training

By observing neural network training characteristics, a heuristic algorithm was developed and implemented in FORECASTER. The parameters for the heuristic are set within the Training Parameters dialog (see Section **Error! Reference source not found.**). The heuristic requires the user to set the learning rate and epochs limit to higher-than-normal values (e.g., 0.3 and 500,000, respectively) and the error limit to a lower-than-normal value (e.g., 1×10^{-10}). The heuristic also uses three additional user-set parameters: the number of training epochs before an application window (view) update (update frequency), the number of updates before a learning rate change (change frequency), and a learning rate change decrement (decrement). Finally, the heuristic requires the data series to be partitioned into a training set and validation set. Given these users set parameters, the heuristic algorithm is:

```
for each view-update during training
    if the validation error is higher than the lowest value seen
        increment count
        if count equals change-frequency
            if the learning rate minus decrement is greater than zero
                lower the learning rate by decrement
                reset count
                continue
            else
                stop training
```

The purpose of the heuristic is to start with an aggressive learning rate, which will quickly find a coarse solution, and then to gradually decrease the learning rate to find a finer solution. Of course, this could be done manually by observing the validation error and using the Change Training Parameters dialog to alter the learning rate. But an automated solution is preferred, especially for an empirical evaluation.

In the evaluation, the heuristic algorithm is compared to the "simple" method of training where training continues until either the number of epochs grows to the epochs limit or the total squared error drops to the error limit. Networks trained with the heuristic algorithm are termed "heuristically trained"; networks trained with the simple method are termed "simply trained".

Finally, the data series in Section **Error! Reference source not found.** are partitioned so that the training set is the first two periods and the validation set is the third period. Note that "period" is used loosely for less noisy, noisier, and ascending, since they are not strictly periodic.

8. Simply Trained Neural Networks with Thirty-Five Inputs

Figure graphically shows the one-period forecasting accuracy for the best candidates for simply trained 35:2:1, 35:10:1, and 35:20:1 networks. Figure (a), (b), and (c) show forecasts from networks trained on the original, less noisy, and more noisy data series, respectively, and the forecasts are compared to a period of the original data series. Figure (d) shows forecasts from networks trained on the ascending data series, and the forecasts are compared to a fourth "period" of the ascending series. In Figure (c) the 35:2:1 network is not included.

Figure graphically compares metrics for the best candidates for simply trained 35:2:1, 35:10:1, and 35:20:1 networks. Figure (a) and (b) compare the total squared error and unscaled error, respectively, and (c) compares the coefficient of determination. The vertical axis in (a) and (b) is logarithmic. The number of epochs and training time are not included in Figure because all networks were trained to 100,000 epochs.

Finally, Table lists the raw data for the charts in Figure and other parameters used in training. Refer to **Error! Reference source not found.** for more parameters.

In this case, it seems the heuristic was inappropriate. Notice that the heuristic allowed the 35:2:1 network to train for 28,850 epochs more than the simple method. Also, the total squared error and unscaled error for the heuristically trained network were lower, but so was the coefficient of determination it was much lower. Again, the forecasts for the 35:10:1 and 35:20:1 networks are near perfect, and are indistinguishable from the original data series.

In Figure (b) the 35:2:1 network forecast is worse than in **Error! Reference source not found.** (b), whereas the 35:10:1 and 35:20:1 forecasts are about the same as before. Notice that the 35:10:1 forecast is from the committee, but does not appear to smooth the data series' sharp transitions.

In Figure (c), the 35:2:1 network is not included because of its poor forecasting performance on the more noisy data series. The 35:10:1 and 35:20:1 forecasts are slightly worse than before.

In Figure (d), the 35:2:1 network is included and its coefficient of determination is much improved from before. The 35:10:1 and 35:20:1 network forecasts are decent, despite the low coefficient of determination for 35:10:1. The forecasts appear to be shifted up when compared to those in **Error! Reference source not found.** (d).

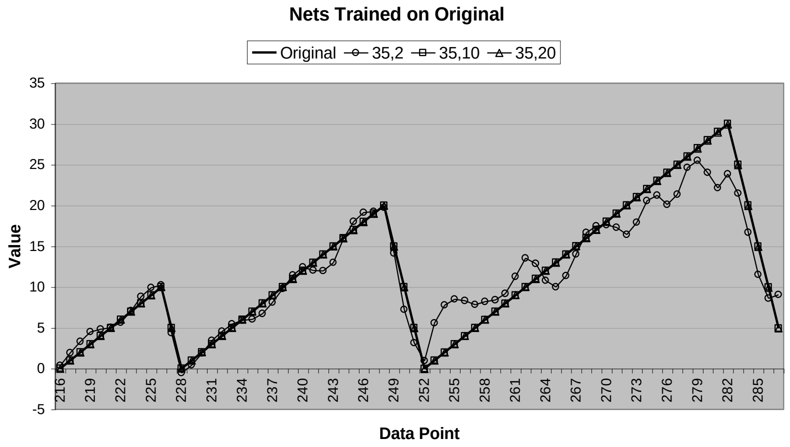
Finally, by evaluating the charts in Figure and data in Table some observations can be made:

The total squared error and unscaled error are higher for noisy data series with the exception of the 35:10:1 network trained on the noisier data series. It trained to extremely low errors, orders of magnitude lower than with the heuristic, but its coefficient of determination is also lower. This is probably an indication of overfitting the noisier data series with simple training, which hurt its forecasting performance.

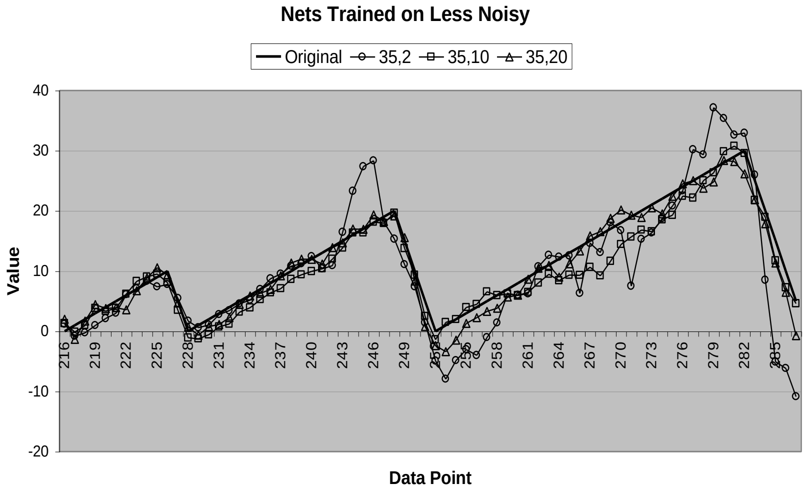
The errors do not appear to correlate well with the coefficient of determination.

In most cases, the committee forecast is worse than the best candidate's forecast.

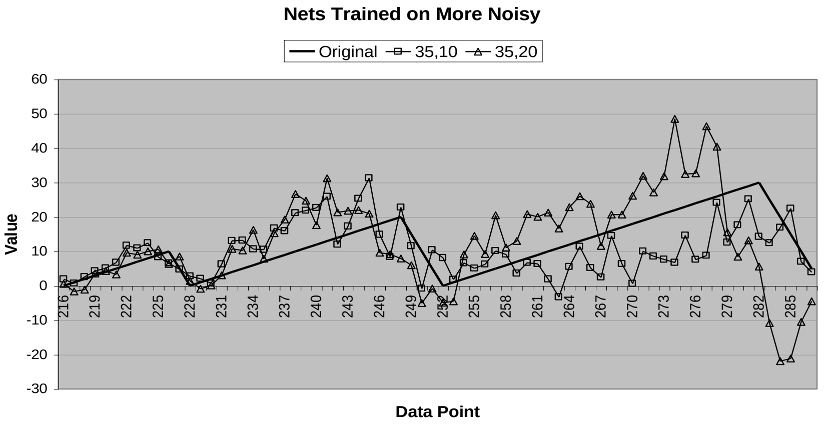
There are four networks whose coefficient of determination is negative, compared with two for the heuristic training method.


Nets Trained on Original
Original
35,2
35,10
35,20
Value
Data Point
35
30
25
20
15
10
5
0
-5
216
219
222
225
228
231
234
237
240
243
246
249
252
255
258
261
264
267
270
273
276
279
282
285

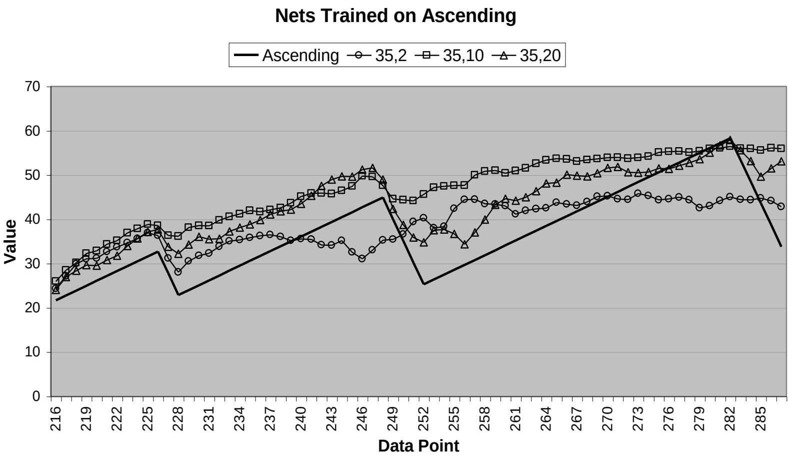
(a)


Nets Trained on Less Noisy
Original
35,2
35,10
35,20
Value
Data Point
40
30
20
10
0
-10
-20
216
219
222
225
228
231
234
237
240
243
246
249
258
261
264
267
270
273
276
279
282
285

(b)

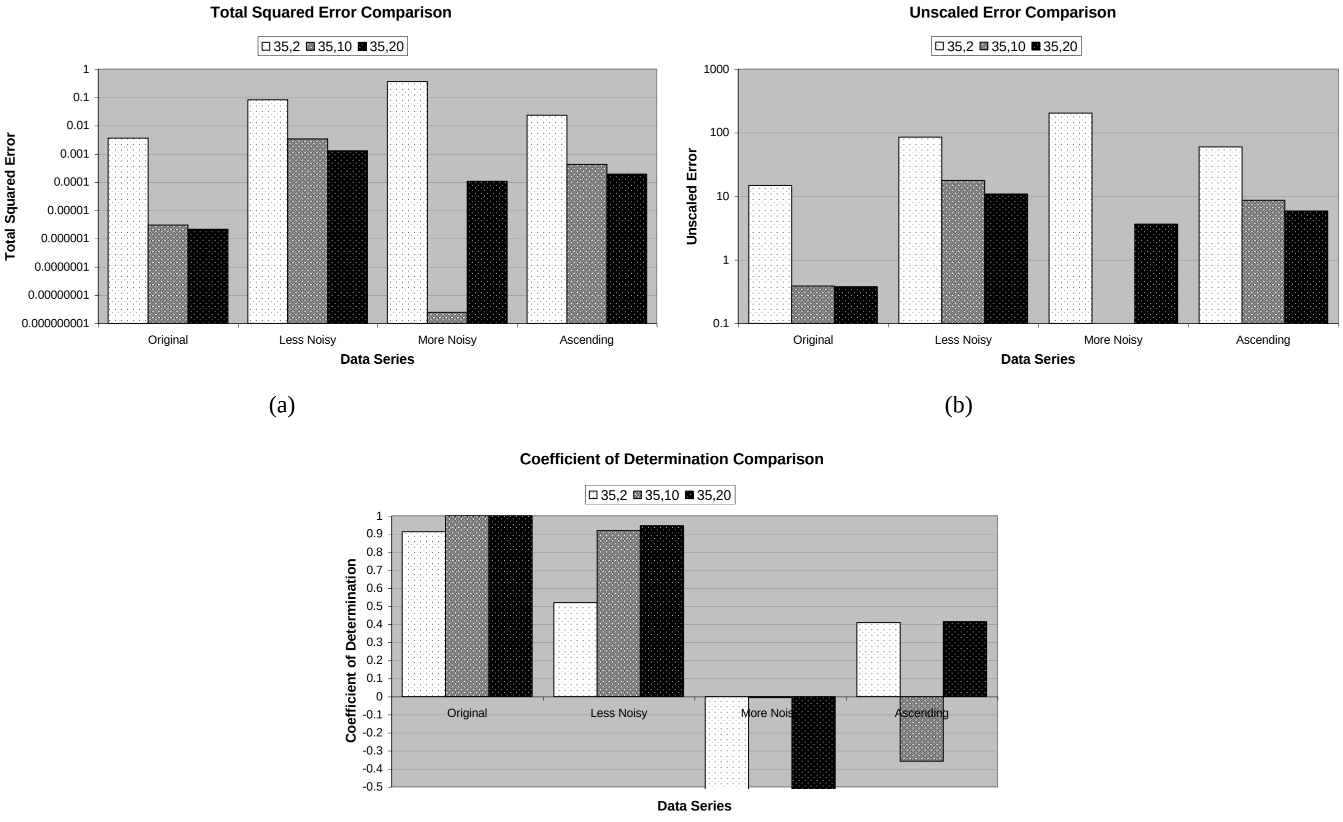

Nets Trained on More Noisy
Original
35,10
35,20
Value
Data Point
60
50
40
30
20
10
0
-10
-20
-30
216
219
222
225
228
231
234
237
240
243
246
249
255
258
261
264
267
270
273
276
279
282
285

(c)



(d)

Figure 3 The one-period forecasting accuracy for the best candidates for simply trained 35:2:1, 35:10:1, and 35:20:1 networks trained on the (a) original, (b) less noisy, (c) more noisy, and (d) ascending data series.



(a)

(b)

(c)

Figure 4 Graphical comparison of metrics for the networks in Figure : (a) total squared error, (b) unscaled error, and (c) coefficient of determination. Note that the vertical axis is logarithmic for (a) and (b). The raw data are given in Table .

Table 3 Parameters and metrics for the networks in Figure .

Arch.	Candidate	Data Series	Ending Learning Rate	Epochs	Total Squared Error	Unscaled Error	Coeff. of Determ. (1 period)	Training Time (sec.)
35:2:1	b	Original	0.1	100000	0.0035864	14.6705	0.9111	99
35:10:1	c	Original	0.1	100000	3.02316e-006	0.386773	1.0000	286
35:20:1	a	Original	0.1	100000	2.15442e-006	0.376312	1.0000	515
35:2:1	b	Less Noisy	0.1	100000	0.0822801	84.8237	0.5201	99
35:10:1	committee (b)	Less Noisy	0.1	100000	0.00341762	17.6535	0.9173	287
35:20:1	b	Less Noisy	0.1	100000	0.00128001	10.8401	0.9453	531
35:2:1	b	More Noisy	0.1	100000	0.360209	203.893	-4.6748	100
35:10:1	a	More Noisy	0.1	100000	2.47609e-009	0.0166514	-0.0056	282
35:20:1	c	More Noisy	0.1	100000	0.000106478	3.65673	-1.5032	519
35:2:1	a	Ascending	0.1	100000	0.023301	59.7174	0.4091	98
35:10:1	a	Ascending	0.1	100000	0.000421792	8.67945	-0.3585	280
35:20:1	b	Ascending	0.1	100000	0.000191954	5.87395	0.4154	529

9. REFERENCES

[1] M. Abdollahzade, A. Miranian, H. Hassani, and H. Iranmanesh, “A new hybrid enhanced local linear neuro-fuzzy model based on the optimized singular spectrum analysis and its application for nonlinear and chaotic time series forecasting,” Inf. Sci. (Ny)., vol. 295, pp. 107–125, 2015.

[2] J. Ares, J. A. Lara, D. Lizcano, and S. Suárez, “A soft computing framework for classifying time series based on fuzzy sets of events,” Inf. Sci. (Ny)., vol. 330, pp. 125–144, 2016.

[3] M.-Y. Chen and B.-T. Chen, “A hybrid fuzzy time series model based on granular computing for stock price forecasting,” Inf. Sci. (Ny)., vol. 294, pp. 227–241, 2015.

[4] T.-T. Chen and S.-J. Lee, “A weighted LS-SVM based learning system for time series forecasting,” Inf. Sci. (Ny)., vol. 299, pp. 99–116, 2015.

[5] Z. Chen, W. Zuo, Q. Hu, and L. Lin, “Kernel sparse representation for time series classification,” Inf. Sci. (Ny)., vol. 292, pp. 15–26, 2015.

[6] S.-H. Cheng, S.-M. Chen, and W.-S. Jian, “Fuzzy time series forecasting based on fuzzy logical relationships and similarity measures,” Inf. Sci. (Ny)., vol. 327, pp. 272–287, 2016.

[7] H. Deng, G. Runger, E. Tuv, and M. Vladimir, “A time series forest for classification and feature extraction,” Inf. Sci. (Ny)., vol. 239, pp. 142–153, 2013.

[8] L. N. Ferreira and L. Zhao, “Time series clustering via community detection in networks,” Inf. Sci. (Ny)., vol. 326, pp. 227–242, 2016.

[9] M. González, C. Bergmeir, I. Triguero, Y. Rodríguez, and J. M. Benítez, “On the stopping criteria for k-Nearest Neighbor in positive unlabeled time series classification problems,” Inf. Sci. (Ny)., vol. 328, pp. 42–59, 2016.

[10] X. Huang, Y. Ye, L. Xiong, R. Y. K. Lau, N. Jiang, and S. Wang, “Time series k-means: A new k-means type smooth subspace clustering for time series data,” Inf. Sci. (Ny)., vol. 367–368, pp. 1–13, 2016.

[11] A. Kattan, S. Fatima, and M. Arif, “Time-series event-based prediction: An unsupervised learning framework based on genetic programming,” Inf. Sci. (Ny)., vol. 301, pp. 99–123, 2015.

[12] M. Krawczak and G. Szkatuła, “An approach to dimensionality reduction in time series,” Inf. Sci. (Ny)., vol. 260, pp. 15–36, 2014.

[13] L. Liu, Y. Peng, S. Wang, M. Liu, and Z. Huang, “Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors,” Inf. Sci. (Ny)., vol. 340, pp. 41–57, 2016.

[14] A. Marszałek and T. Burczyński, “Modeling and forecasting financial time series with ordered fuzzy candlesticks,” Inf. Sci. (Ny)., vol. 273, pp. 144–155, 2014.

[15] S. Miao, U. Vespier, R. Cachucho, M. Meeng, and A. Knobbe, “Predefined pattern detection in large time series,” Inf. Sci. (Ny)., vol. 329, pp. 950–964, 2016.

[16] V. Novák, I. Perfilieva, M. Holčapek, and V. Kreinovich, “Filtering out high frequencies in time series using F-transform,” Inf. Sci. (Ny)., vol. 274, pp. 192–209, 2014.

[17] H. Pree, B. Herwig, T. Gruber, B. Sick, K. David, and P. Lukowicz, “On general purpose time series similarity measures and their use as kernel functions in support vector machines,” Inf. Sci. (Ny)., vol. 281, pp. 478–495, 2014.

[18] M. Pulido, P. Melin, and O. Castillo, “Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange,” Inf. Sci. (Ny)., vol. 280, pp. 188–204, 2014.

[19] T. Xiong, Y. Bao, Z. Hu, and R. Chiong, “Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms,” Inf. Sci. (Ny)., vol. 305, pp. 77–92, 2015.

[20] F. Ye, L. Zhang, D. Zhang, H. Fujita, and Z. Gong, “A novel forecasting method based on multi-order fuzzy time series and technical analysis,” Inf. Sci. (Ny)., vol. 367–368, pp. 41–57, 2016.

[21] H. Zhao, Z. Dong, T. Li, X. Wang, and C. Pang, “Segmenting time series with connected lines under maximum error bound,” Inf. Sci. (Ny)., vol. 345, pp. 1–8, 2016.

[22] S. Zhu, Q.-L. Han, and C. Zhang, “Investigating the effects of time-delays on stochastic stability and designing l1-gain controllers for positive discrete-time Markov jump linear systems with time-delay,” Inf. Sci. (Ny)., vol. 355, pp. 265–281, 2016.

International Journal of Computer Applications Technology and Research
Volume 5 –Issue 12, 760-763, 2016, ISSN:-2319–8656

Preprocessing Phase for Offline Arabic Handwritten Character Recognition

Rawia I. O. Ahmed
College of Computer Science and Information Technology Sudan University of Science and Technology Khartoum,
Sudan

Mohamed E. M. Musa
College of Computer Science and Information Technology Sudan University of Science and Technology Khartoum,
Sudan

Abstract: —In this paper we reviewed the importance issues of the optical character recognition, gives more emphases for OCR and its phases. We discuss the main characteristics of Arabic language, furthermore it focused on the pre-processing phase of the character recognition system. We described and implemented the algorithms of binarization, dots removing and thinning which will be used for feature extraction phase. The algorithms are tested using 47,988 isolated character sample taken from SUST/ ALT dataset and achieved better results. The pre-processing phase developed by using MATLAB software.

Keywords: optical character recognition; offline recognition; online recognition; handwritten, preprocessing.

1. INTRODUCTION

Over the past three decades, many studies have been concerned with the recognition of Arabic words. Offline handwritten Arabic characters recognition have received more attention in these studies, because of the need to Arabic document digitalization.

In this paper, preprocessing system for an isolated Arabic handwritten are design and tested by using SUST/ ALT dataset. it’s a new dataset developed and published by SUST/ALT (Sudan University of Science and Technology-Arabic Language Technology group) group. It contains numerals datasets, isolated Arabic character datasets and Arabic names datasets[1]. 40 common Arabic (especially in Sudan) males and females' name[2]. Each form written by one writer resulting 40,000 sample. it used for researching purpose.

The rest of the paper is organized as follows: the concepts of OCR approaches are described in Section 2. Then the main characteristics of Arabic language are discussed in Section 3. Then phases involved in OCR system are discussed as general in Section 4, These phases are: preprocessing, segmentation, feature extraction and classification. The proposed Preprocessing phase discussed in Section 5. conclusion and future work are presented in Section 6.

2. THE OPTICAL CHARACTER RECOGNITION APPROACHES

The Optical Character Recognition (OCR) is one of important tasks in computer area. It has many definitions, OCR defined as a process that attempts to turn a paper document into a fully editable form, which can be used in word processing and other applications as if it had been typed through the keyboard[3]. Also OCR was defined by Srihari et al. as the task of transforming text represented in the special form of graphical marks into its symbolic representation[4] .

The recognition of handwritten can be applied in many areas such as names of persons, companies, organizations, newspapers, letters, archiving and retrieving texts, proteins and genes in the molecular biology context, journals, books, bank chequs, personal signatures and digital recognition, etc.[5]. A recognition system can be either online or offline[6]. It is online if the data being captured during the writing process. It always captured by special pen on an electronic interface. Online recognition has several interesting characteristics: firstly, recognition is performed on one dimensional rather than two dimensional images, secondly, the writing line is represented by a sequence of dots which its location is a function of time[3].A recognition system is offline if its data scanned by scanner after writing process is over, such as any images scanned in by a scanner. In this case, only the image of the handwriting is available.

When we compared online handwriting recognition systems with offline systems, we found that offline systems are considered more difficult than online systems. This difficulty due to several reasons, out of which online handwriting recognition depends on temporal information, which facilitate the recognition system, but the temporal information is lacked in offline handwriting, it depends on passive images stored in files. This lemma makes offline systems less accurate than online systems. Furthermore, offline systems are more complex than online systems, because they depend on human writing which had more feature and characteristic specially for Arabic language. Table.1 summarizes the differences between online and offline recognition systems.

Error! Reference source not found. **The differences between online and offline recognition systems**

Criteria of recognition	on-line recognition system	off-line recognition system
Data Capture	during the writing process	scanned in by a scanner or camera.
Data Type	Temporal information	Not temporal information
Accuracy	More accurate	Less accurate
Complicity	Less complex	More complex

3. The MAIN CHARACTRERISTICS OF ARABIC LANGUAGE

Many studies have been conducted on recognition of Chinese, Japanese and Latin languages, but few were done on Arabic handwritten recognition[7]. One of the main reasons for this is that characteristics of Arabic language do not allow direct implementation of many algorithms used in other languages. The characteristics of Arabic language can be summarized as follows:

- Arabic language is represented in 28 characters and appears in different four shapes isolated, initial, medium or final.

- Arabic language is written from right to left, rather than from left to right this is useful for human reader rather than for the computer.

- Arabic characters of a word are connected a long baseline, and character position above and below the baseline. As seen in Figure.1 which illustrates the word "samah"; the character ˝Seen˝ appear above the baseline, while character ˝Meem˝ appears below the baseline.

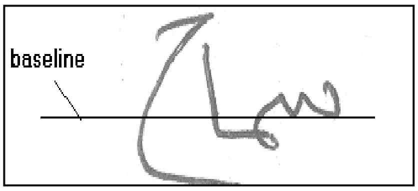


Figure.1 Arabic characters of a word are connected a long baseline

- Some Arabic character have the same shape and differ in the number of dots by which it will be identified, for example characters ث ,ت ,ب have the same shape but differ in number of dots, one dot in character Baa, two dots in character Taa, and three dots in character Thaa.

- Some Arabic character have the same shape and differ in the position of dots by which it will be identified, for example characters ن ,ب the two characters have the same shape and identify with one dot, but they differ in position of dot one is above the baseline (character Noon), and other under the base line (character Baa), this differentiation can change the meaning of a word.

- The width and high of Arabic characters are differ from one character to another.

- The shape of Arabic character varies per writer.

- Arabic writing is cursive, most of Arabic characters are connected from two sides; right and left, only six characters are connected from right side only, as shown in Figure.2.

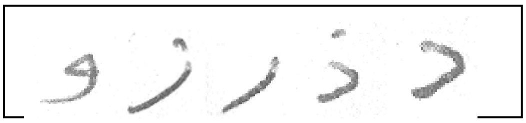


Figure.2 Arabic characters which can be connected from right to left

- Moreover, Arabic language has some diacritics called Tashkeel. The names of these Tashkeel: Fatha, Dhamma, Kasra, Sukun, Shadda, Fathatain, Kasratain, Dhammatain also combination of them are possible. These diacritics may change the meaning of specific word, for example: when we put Fatha diacritic on the word "حر" it became "حَر" which meaning "hot weather", when we put dhamma diacritics on the same word, it became "حُر" which meaning "free".

- Some Arabic words consists of more than one sub-words. A sub-word is the basic standalone pictorial block of the Arabic writing [8]. A brief details of Arabic handwritten characteristic were reviewed by Lorigo [9].

4. RECOGNITION SYSTEM PHASES

OCR systems either can be online or offline. There is no variation between phases of both systems. It depends on lexicon nature, and the recognition approach. The lexicon is a key point to the success of any OCR system. As the size of lexicon grows, the recognition efforts and the complexity are increased. So, the general phases of OCR can be described by six phases[10]. First, the data can be captured by several ways depending on the system (online or offline). Then the scanned text image may need to be passed through several preprocessing steps. After the preprocessing process, the text image may need to be segmented into lines, words, pieces of words, characters or pieces of character. To facilitate the recognition phase, useful features are extracted from the text image, then the valuable classifier methods were used to build the model. Finally, to improve the recognition rate, some post-processing operations may be applied on the model. But post processing phase can scarcely apply and limited to few systems as in [11, 12]. So, the general phases of OCR systems are: data capture, preprocessing, segmentation, feature extraction, classification and, post processing as shown in Figure .3

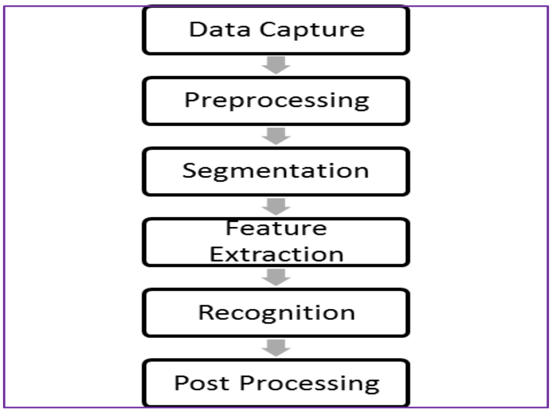


Figure.3 General recognition system phases

5. THE PROPOSED PREPROCESSING PHASE

Prior to the features extraction phase preprocessing phase must be done. Pre-processing of the handwritten character image is an important factor, to simplify the task of recognition. Usually several operations cans be performed in this phase. Since in SUST isolated characters dataset, some preprocessing method are done during the development stage[1],minimal number of preprocessing processes are used in this work. An image file of isolated handwritten character will first be introduced to the system as gray scale bmp image. Then obtained images are binarized to be in digital form. When the study focus on characters' body only, dots is removed from some characters. Thinning is very important process in OCR, therefore we applied it the binary images. The next sub sections give a brief detail of these operations.

5.1 Binarization

Binarization operation attempts to converted the gray scale image into a binary image based on threshold. So, the bitmap images are threshold and converted into 1s and 0s forms. Two types of thresholding are existing. These types are global and local thresholding. In global thresholding, threshold selection leads to a single threshold value for the entire image[13].This value is often based on an estimation of the background intensity level of the using an intensity histogram. In local thresholding different values are used for each pixel according to the local area information[14].

Since the proposed system implemented on simple isolated handwritten character images, where the characters can be to distinguish into background and foreground pixels, the global thresholding methods are sufficient for this type of images. Therefore we use Otsu’s method[15]. It applied on dataset to converted the image into 1s (background) and 0s(foreground).

5.2 Dots Removing

When the goal of the system is to design an offline handwritten recognition system to deal with isolated Arabic handwritten character body written by multiple writers, before recognition all dots need to be removed. Some of Arabic characters, may have three, two or one dots such as “ ب ,ت ,ث ” characters or may be without any dot such as “و , د , ح” characters. We removed dots from the (Baa, Faa, Noon and Yaa) characters images to extract the character's bodies only. When applying this operation some of data are loss during this process (about 601 samples), Table.2 illustrated the accuracy rate of this process.

Table .2 The accuracy rate of dots removing

character	Accuracy rate
Baa	90.28%
Fah	94.33%
Noon	95.39%
Yaa	77.38%

Samples are loss due to two reasons. The first reason, is the writing style of the writers, for examples some writers connected dots with the main body of characters "Baa, Fah and Yaa", or writing dot inside character "Noon" as shown in Figure.4a & Figure.4b.

Figure .4a

The original images

Figure .4b

Images after dots removing process

The second reason, due to unclear samples from the original dataset, Figure.5a displays some unclear samples from the original dataset, and Figure.5b displays the same samples after removing dots.

Figure .5a

Unclear samples from the original dataset

Figure .5b

The same samples after dots removing.

5.3 Thinning

Finally, the character body image is thinned by T.Y. Zhang and C. Y. Suen algorithm [16] to maintained the connectivity of skeleton and extracted the edges which is be the input to the feature extraction phase. Figure.6a displays an image for character "Baa" body before thinning, and Figure.6b displays the same image after thinning.

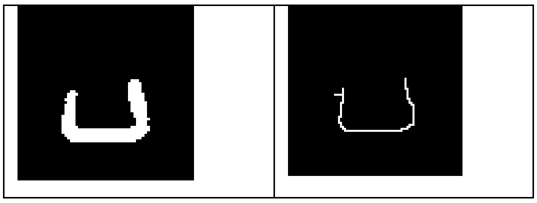


Figure .6a

Image Before Thinning

Figure .6b

Image After Thinning

6. CONCLUSION AND FUTURE WORK

In this paper, we present a review about optical character recognition and its importance, and its main approaches and techniques. Also, we list the characteristics of Arabic language, and focused in one of important phases in recognition systems which is preprocessing. Moreover, we described and implemented preprocessing algorithms to binarized, dots removing and thinning for Arabian characters. In the future, we will use the result from this phase to extract features and design recognition system.

7. REFERENCES

[1] Musa, M.E. Arabic handwritten datasets for pattern recognition and machine learning. in 2011 5th International Conference on Application of Information and Communication Technologies (AICT).

[2] Wahby, T.M., I.M. Osman, and M.E. Musa, On Finding the Best Number of States for a HMM-Based Offline Arabic Word Recognition System. 2011.

[3] Mori, S., H. Nishida, and H. Yamada, Optical character recognition. 1999: John Wiley & Sons, Inc.

[4] Srihari, S.N., A. Shekhawat, and S.W. Lam, *Optical character recognition (OCR).* 2003.

[5] Amin, A., Off-line Arabic character recognition: the state of the art. Pattern recognition, 1998.

[6] Khorsheed, M.S., Off-line Arabic character recognition– a review. Pattern analysis & applications, 2002.

[7] Al-Emami, S. and M. Usher, On-line recognition of handwritten Arabic characters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990.

[8] Abed, M.A., Freeman chain code contour processing for handwritten isolated Arabic characters recognition. Alyrmook University Magazine,Baghdad, 2012.

[9] Lorigo, L.M. and V. Govindaraju, Offline Arabic handwriting recognition: a survey. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2006.

[10] O'Gorman, L. and R. Kasturi, Document image analysis. Vol. 39. 1995: IEEE Computer Society Press Los Alamitos, CA.

[11] Amin, A., G. Masini, and J. Haton, Recognition of handwritten Arabic words and sentences. ICPR84, 1984.

[12] Amin, A. and J.F. Mari, Machine recognition and correction of printed Arabic text. IEEE Transactions on systems, man, and cybernetics, 1989.

[13] Gatos, B., I. Pratikakis, and S.J. Perantonis, Adaptive degraded document image binarization. Pattern recognition, 2005.

[14] Suliman, A., et al., Chain Coding and Pre-Processing Stages of Handwritten Character Image File. electronic Journal of Computer Science and Information Technology, 2011.

[15] Otsu, N., A threshold selection method from gray-level histograms. Automatica, 1975.

[16] Zhang, T. and C.Y. Suen, A fast parallel algorithm for thinning digital patterns. Communications of the ACM, 1984

International Journal of Computer Applications Technology and Research
Volume 5–Issue 12, 764-774, 2016, ISSN:-2319–8656

A Hybrid Approach for Personalized Recommender System Using Weighted TFIDF on RSS Contents

Rebecca A. Okaka
SCIT, JKUAT
Nairobi, Kenya

Waweru Mwangi
SCIT, JKUAT
Nairobi, Kenya

George Okeyo
SCIT, JKUAT
Nairobi, Kenya

Abstract: Recommender systems are gaining a great popularity with the emergence of e-commerce and social media on the internet. These recommender systems enable users' access products or services that they would otherwise not be aware of due to the wealth of information on the internet. Two traditional methods used to develop recommender systems are content-based and collaborative filtering. While both methods have their strengths, they also have weaknesses; such as sparsity, new item and new user problem that leads to poor recommendation quality. Some of these weaknesses can be overcome by combining two or more methods to form a hybrid recommender system. This paper deals with issues related to the design and evaluation of a personalized hybrid recommender system that combines content-based and collaborative filtering methods to improve the precision of recommendation. Experiments done using MovieLens dataset shows the personalized hybrid recommender system outperforms the two traditional methods implemented separately.

Keywords: recommender systems; collaborative filtering; content-based filtering; hybrid recommender system; vector space model; term frequency inverse document frequency.

1. INTRODUCTION

Changes in information seeking behavior can be observed globally [1]. Rapid increase in blogs and websites has led to an increase in information overload and it has become extremely difficult for users to locate current relevant information, with vague ideas on where to get information, users often get lost or feel uncertain when seeking information on their own, giving rise to the need for creating systems that are able to process the existing information on one side, and help users by suggesting products, services or articles that match their tastes and preferences on the other side. Recommender systems (RS) are promising tools to deal with these issues.

There are lots of taxonomies of RS. They can be divided according to the fact whether the created recommendation is personalized or non-personalized [2]. Some Research distinguishes three main categories of personalized RS: collaborative filtering (CF), content-based filtering (CBF), and hybrid filtering (HF) [3]. Adomavicius and Tuzhilin claim that these three categories are the most popular and significant recommendation methods. However, they pinpoint the shortcomings of these methods when used individually such as limited content analysis, new item problem, new user problem, sparsity, scalability etc, which leads to poor recommendation quality. They also propose possible improvements; such as combining two or more recommender filtering methods using different hybridization techniques to overcome the challenges of single recommender systems.

In CF, a user gets recommendations of items that he or she hasn't rated or liked before, but that were already positively rated by users in his or her neighborhood. In CBF, a user gets recommendations of items he or she had not seen or rated but similar to the ones he or she had rated or liked earlier. HF combines two or more filtering methods to overcome the limitations of each method. According to Tuzhilin et al, [4] the combination of two or more filtering methods proceeds in different ways; creating a unified model recommender system that brings all approaches together, utilizing some rules of one approach into a different approach and vice versa, separate implementation of algorithms and then joining results, developing one model that applies the characteristics of both methods.

The hybrid approach presented in this paper uses the weighted hybridization technique which probably is the most straight forward architecture for a hybrid system. Weighted hybridization technique was successfully used by the winners of the Netflix Prize competition [5]. Our approach involves separate implementation of algorithms then joining results, it is based on the idea of merging predicted ratings computed by individual recommenders to form a ranked list of items from which top (top k, k=5) items are selected and presented to the user as recommendations.
This hybrid approach combines CBF and CF methods, while CBF are able to make predictions on any item, CF only score an item if there are peer users who have rated it, the combination of these two methods therefore also helps eliminate the new item problem in CF and new user problem in CBF. This hybrid approach adapts the Vector Space Model (VSM) in both CBF and CF, uses ranking algorithm Term Frequency Inverse Document Frequency (TFIDF) and cosine similarity measure to find the relationships among users *U*, items *I* and attributes *A*.

Generally, in a recommender system, there exists a large number of *m* items *I*= $\{i_1, i_2....i_m\}$, which are described by a set of *l* attributes, *A*= $\{a_1, a_2....a_l\}$, where each item is described by one attribute or more, a number of *n* users, *U*= $\{u_1, u_2....u_n\}$ and for each user *u*, a set of rated items $IR_u = \{u_{i1}, u_{i2}, \dots, u_{in}\}$. For $u \epsilon U$ and $i \epsilon I$, the recommender system predicts the rating $r'_{u,i}$called the predicted rating of the user *u* on the item *i* such that$r'_{u,i}$is unknown. From this formulation, the main problem is predicting the rating a user would give an item he or she have not seen, then computing the accuracy of the predicted rating.

The main contribution of this work is that it provides a very straight forward hybrid architecture that can be used to improve recommendation precision as well as provide top most relevant items to users as recommendations. Because of the two methods used; content-based and collaborative filtering, the new user and new item problems is eliminated; the new user problem in content-based filtering is eliminated by collaborative filtering and the new item problem in collaborative filtering is eliminated by content-based filtering. This hybrid approach uses the most widely used effective information retrieval model, the VSM, and a very simple efficient ranking algorithm tfidf.

The rest of this paper is organized as follows; section 2 reviews related work. Section 3 presents the hybrid model and experimental results are presented in section 4. Section 5 presents conclusions and outlines of future research.

2. RELATED WORK

Hybrid recommender systems combine two or more recommender systems. Depending on the hybridization approach different types of systems can be found [6]. There have been some works on using boosting algorithms for hybrid recommendations [7, 8]. These works attempt to generate new synthetic ratings in order to improve recommendation quality. The personalized hybrid recommender system combines collaborative and content-based information.

Spiegel [9] proposed a framework that combines CBF, CF and demographic information for recommending information sources such as web pages or news articles. The author used home HTML pages to gather demographic information of users. The recommender system is tested on very few numbers of users and items which cannot guarantee the efficiency of the proposed system.

The author does not also give an explanation on how the model is built.

Melville [10] proposed a model in which content-based algorithm is used to enhance the existing user data then the collaborative filtering is used for rating prediction. But fails to justify how both approaches combined improves prediction accuracy. Another researcher [11] used a number of collaborative filtering algorithms such as Singular Value Decomposition (SVD), Asymmetric Factor Model and neighborhood based approaches to build a recommender system. The author shows that linearly combining these algorithms increases the accuracy of prediction, but the use of all these models leads to significant increase in training time.

Basu et al. [12] use Ripper, a rule induction system, to learn a function that takes a user and movie and predicts whether the movie will be liked or disliked. They combine collaborative and content information, by creating features such as comedies liked by user and users who liked movies of genre X. They however do not show how that approach improved recommendation quality.

Several other hybrid approaches are based on traditional CF, but also maintain a content-based profile for each user. These content-based profiles, rather than co-rated items, are used to find similar users. In Pazzani's approach [13], each user profile is represented by a vector of weighted words derived from positive training examples using the Winnow algorithm. Predictions are made by applying CF directly to the matrix of user profiles as opposed to the user ratings matrix. An alternative approach by; Fab [14] uses relevance feedback to simultaneously mold a personal filter along with a communal topic filter. Documents are initially ranked by the topic filter and then sent to a user's personal filter. The user's relevance feedback is used to modify both the personal filter and the originating topic filter. Good et al. [15] use collaborative filtering along with a number of personalized information filtering agents. Predictions for a user are made by applying CF on the set of other users and the active user's personalized agents.

The proposed hybrid approach adapts some interesting features of the above systems; the use of collaborative and content information. It however uses the VSM, tfidf and cosine similarity measure which are very simple efficient algorithms that enable item ranking based on weights. Prediction accuracy is computed by getting the deviation of the predicted rating from the actual rating. Other works on hybrid recommender systems can be found in [16].

3. THE HYBRID FILTERING MODEL

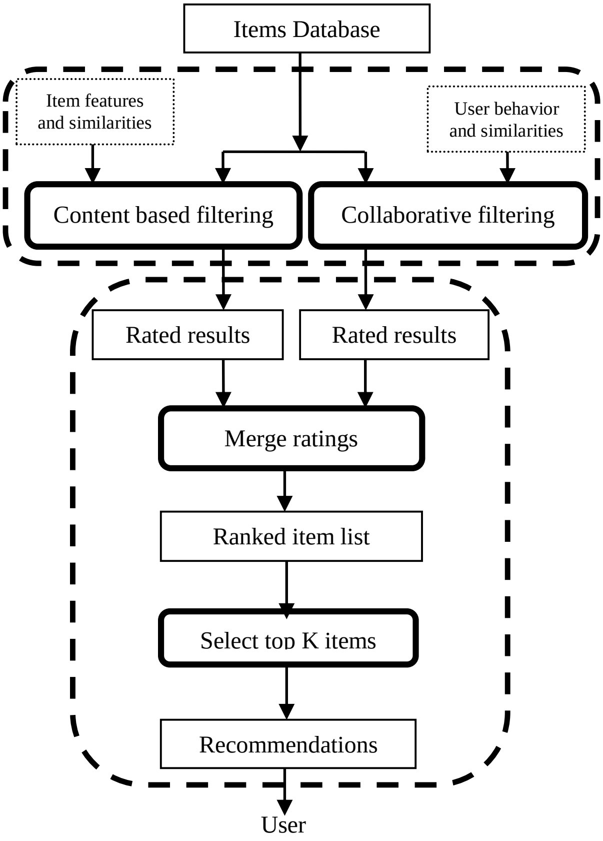


Figure 1. The hybrid filtering model

available on different domains, the model implements both CBF and CF methods separately. The two methods used (CBF and CF) complement each other and contribute to each other's effectiveness [17]. This hybrid approach uses the VSM on CBF and CF methods, tfidf and cosine similarity measure to compute relationships among items and users. CF and CBF methods are used to obtain separate ratings or score for every item. The more than one rating for every item are merged into a single value. The items are then ranked to form a single list of ranked items based on their scores, a set of items (e.g. top K, where K equals 5, 10….) topping the list (with highest scores or ratings) are finally presented to the user as recommendations.

3.1.The Vector Space Model

The vector space model [18] (VSM) is a standard algebraic model commonly used in information retrieval (IR). It treats a textual document as a bag of words, disregarding grammar and even word order. It represents both documents and queries by term sets and compares global similarities between documents and queries. The VSM typically uses tfidf (or a variant weighting scheme) to weight the terms. Then each document is represented as a vector of tfidf weights. Queries are also considered as documents. Cosine similarity is used to compute similarity between document vectors and the query vector. The term frequency $TF_{t,d}$ of term t in document d is defined as the number of times that a term t occurs in a document d. Note that;

$$TF_{t,d} = 1 \text{ if t exists in d} \qquad (1)$$

$$TF_{t,d} = 0 \text{ if t does not exist in d} \qquad (2)$$

It positively contributes to the relevance of d to t. The inverse document frequency IDF_t of term t measures the rarity of t in a given corpus. If t is rare, then the documents containing tare more relevant to t. IDF_t is obtained by dividing N by DF_t and then taking the logarithm of that quotient, where N is the total number of documents and DF_t is the document frequency of t or the number of documents containing t. Formally;

$$IDF_t = \frac{\log_{10} N}{DF_t} \qquad (3)$$

The TFIDF value of a term is commonly defined as the product of its TF and IDF values.

$$TF\text{-}IDF_{\ t,d} = TF_{t,d} \times IDF_t \qquad (4)$$

The TF-IDF weight W, for each term in a document d is given by;

$$W_{t,d} = (1 + \log_{10} TF_{t,d}) \times \frac{\log_{10} N}{DF_t} \qquad (5)$$

Generally;

$$W_{t,d} = \frac{1+\log_{10} N}{DF_t} \text{if } TF_{t,d} > 0 \qquad (6)$$

$$W_{t,d} = 0, \text{ otherwise} \qquad (7)$$

3.1.1 The Vector Space Model in Content-based filtering

Suppose a user profile is denoted by U and item profiles by I. $TF_{i,j}$ is the number of times the term t_i occurs in item $I_j \in I$, and the inverse document frequency of a term $t_i \in I_j \in I$ is calculated as;

$$IDF_i = \log_{10} I / DF_i \qquad (8)$$

Where DF_i is equal to the number of items containing t_i and I is equal to the total number of items being considered. Therefore;

$$TFIDF = TF_{i,j} \times IDF_i \qquad (9)$$

The TFIDF of each term is then calculated, and the vector of each user profile and item profiles are constructed based on their included terms. These vectors have the same length, so the similarity of these profiles can be calculated as;

$$Sim(U,I) = \frac{U\ .\ I}{|U| \times |I|} = \frac{\sum_1^t tfidf_U \times tfidf_I}{\sqrt{\sum_1^t tfidf_U^2 + \sum_1^t tfidf_I^2}} \qquad (10)$$

The resulting similarity should range between from 0 to 1. If Sim(U,I)=0,then the two profiles are independent and if Sim(U,I) > 0, the profiles have some similarity. Information about a set of items with similar rating patterns compared to the item under consideration is the basis for predicting the rating a U_i would give the item. The prediction formula is;

$$\text{Pred}(U_{i,}\ I_a) = \frac{\sum \text{similarity}(U_i\ ,I_b)\times r_{Ui,Ia}}{\sum \text{similarity}(U_i\ ,I_b)} \qquad (11)$$

Normally, the predicted rating of a user *u* for an item *i* in CBF is the average rating of the user on items viewed, therefore equation 11 can also be written as;

$$r'_{u,i}|\text{CBF} = \frac{\sum \text{similarity}(U_i\ ,I_b)\times r_{Ui,Ia}}{\sum \text{similarity}(U_i\ ,I_b)} \qquad (12)$$

$$r'_{u,i}|\text{CBF} = r_{Ui,Ia} \qquad (13)$$

Where $r_{Ui,Ia}$, isthe average rating of Ui on items already is viewed, and $r'_{u,i}|\text{CBF}$is the predicted rating of a user on an item in CBF.

3.1.2 The Vector Space Model in Collaborative filtering

The user profiles are represented as both documents and queries in an n-dimensional matrix. The weight for each term t in a user profile p is given by:

$W_{i,\ j} = TF_{i,j} \times IDF_i$ which can also be written as;

$$W_{i,\ j} = TF_{i,j} \times \log_{10} P\ /\ p_i \qquad (14)$$

$$IDF_i = \log_{10} P\ /\ p_i \qquad (15)$$

Where, $TF_{i,j}$ is the frequency of a term t in a profile p, P is the total number of profile, p_i is the total number of profiles containing term t and $W_{i,\ j}$ is the weight of the i^{th} term in a profile j. The similarity between user U_i and user U_j is calculated using cosine similarity measure. The equation for calculating the similarity is as follows;

$$\text{Sim}(U_i,U_j)= \frac{U_i\ .\ U_j}{|U_i|\times|U_j|} = \frac{\sum_{k=1}^{n} \text{tfidf}_{k,i}\times \text{tfidf}_{k,j}}{\sqrt{\sum_{k=1}^{n} \text{tfidf}_{k,i}^2\ \ +\sum_{k=1}^{n} \text{tfidf}_{k,j}^2}} (16)$$

Again the resulting similarity should range between from 0 to 1. If $\text{Sim}(U_i,U_j)=0$,then the two users are independent and if $\text{Sim}(U_i,U_j)=1$,the users are similar. The information about a set of users with a similar rating behavior compared to the current user is the basis for predicting the rating a user U_i would give an item he or she has not rated. Based on the nearest neighbor of user U_i it is easy to determine the prediction of user U_i.

$$\text{Pred}(U_{i,}\ I)= r'_i + \frac{\sum \text{similarity}(U_i,U_j)\times(r_{j,\text{item}}-r'_j)}{\sum \text{similarity}(U_i,U_j)} \ (17)$$

Where, U_j is U_i nearest neighbor, r'_i is the average rating of U_i, $r_{j,\text{item}}$ is the rating of U_j on the given item andr'_j is the average rating of U_j. Also, given that the predicted rating of a user u on an item I in CF is given as$r'_{u,i}|\text{CF}$, equation 17 can therefore be written as:

$$r'_{u,i}|\text{CF} = r'_i + \frac{\sum \text{similarity}(U_i,U_j)\times(r_{j,\text{item}}-r'_j)}{\sum \text{similarity}(U_i,U_j)} \qquad (18)$$

3.2 Hybridization Process

Table1. Extended user-item, user-user matrix

		Item					User profile-Attribute tf-idf					User-User cosine similarity				
		i_1	i_2	i_3	…	i_m	a_1	a_2	a_3	…	a_l	u_1	u_2	u_3	…	u_n
User	u_1	-	-	4	…	3	0.04	0	0	…	0	1	0.1	0	…	0.2
	u_2	4	2	-	…	5	0	0.01	0.02	…	0.02	0.1	1	0.1	…	0
	u_3	-	-	3	…	-	0.04	0	0	…	0.02	0	0.1	1	…	0
	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
	u_n	-	3	-	…	-	0.04	0.01	0.02	…	0.02	0.2	0	0	…	1
Item-Attribute tfiidf	a_1	0.0	0.0	0.0	…	0.0										
	a_2	0.0	0.01	0.0	…	0.01										
	a_3	0.02	0.02	0.0	…	0.0										
	…	…	…	…	…	…										
	a_l	0.02	0.0	0.0	…	0.0										
User-Item cosine similarity	u_1	0.3	0.2	0.1	…	0.1										
	u_2	0.3	0.0	0.0	…	0.0										
	u_3	0.1	0.5	0.3	…	0.4										
	…	…	…	…	…	…										
	u_n	0.0	0.2	0.4	…	0.2										

Rated item

Unrated item

As stated earlier the HF model combines CBF and CF which uses user-item matrix and user-user matrix respectively. Table 1 shows the model matrix with sample tfidf and cosine similarity scores among users and items. This model is based on the idea of deriving recommendation items by combining predictions computed by each individual recommenders CBF (Eq. 13) and CF (Eq. 18), here the separate scores of an individual recommender on an item $i \epsilon I$ recommended to a user $u \epsilon U$ are merged into a single unit.

To take into account the difference in the contribution of each predictor in the final rating prediction, each predictor is assigned a parameter. Such that the resulting rating prediction$r'_{u,i}|$HF of a user *u* on an item *i* from HF is computed as follows;

$$r'_{u,i}\,|\text{HF} = \mu r'_{u,i}\,|\text{CBF} \; + \alpha r'_{u,i}\,|\text{CF} \qquad (19)$$

Where $\mu r'_{u,i}\,|\text{CBF}$ and $\alpha r'_{u,i}\,|\text{CF}$ are the predicted rating of an item $i \epsilon I$ for user $u \epsilon U$ in CBF and CF respectively.

To compute the value for each parameter, a function *S(n)* that gives the weight of a user's rating *n* ($n = |IR_u|$) is used. The sigmoid function satisfies these constraints for *S(n).*

The parameters μ and α can be computed using the sigmoid function as follows;

$$\mu = \frac{1}{1+e^{-n}} \qquad (20)$$

$$\alpha = 1 - \frac{1}{1+\, e^{-n}} \qquad (21)$$

These parameters μ and α, represent the weight confidence levels given to CBF and CF respectively. The resulting rating predictions of items from the hybrid approach are ranked based on their prediction scores, from the ranked items list the top scoring set of items (top k items) are selected and provided to the user as recommendations.

4. HYBRID DESIGN

4.1 System Physical Architecture

Figure 2 below shows the physical architecture of the proposed hybrid recommender system; it shows a set of simpler systems each with its own local context that is independent but not inconsistent with the context of the larger system as a whole. Both servers could still be physically implemented in a single network node.

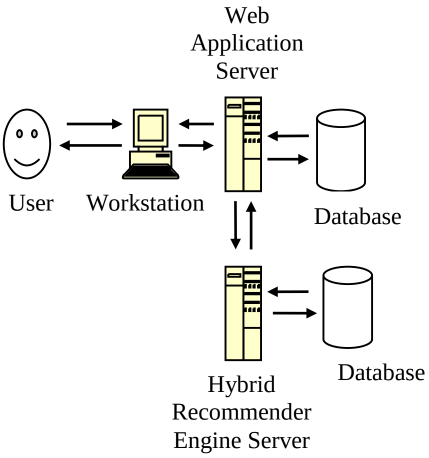


Figure 2. The Physical architecture

4.2 System Component Diagram

Figure 3 shows a simple component viewpoint of the Hybrid Recommender system. The Hybrid Recommender module, while calculating the accurate recommendation, uses the data stored in the Database module via a RESTful API (Representational Sate Transfer Application Program Interface). The Hybrid Recommender module executes the methods on the background. It is connected to the User Interface module via the RecommendationRetrieval interface that enables the resulting recommendations to be shown to the user.

Figure 3. The Component Diagram

4.3 System Activity Diagram

The following activity diagram shows the flow of events within the proposed hybrid approach. It shows how the user interacts with the system.

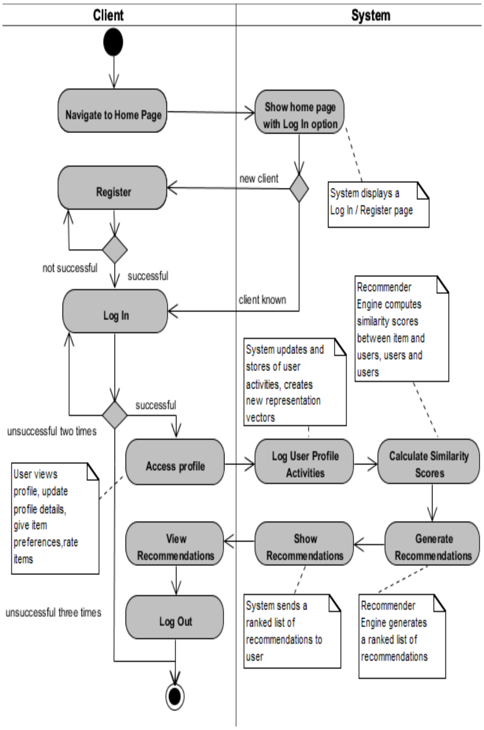

Client
System
Navigate to Home Page
Show home page with Log In option
Register
new client
System displays a Log In / Register page
not successful
successful
Log In
client known
Recommender Engine computes similarity scores between item and users, users and users
System updates and stores of user activities, creates new representation vectors
successful
unsuccessful two times
Access profile
Log User Profile Activities
Calculate Similarity Scores
User views profile, update profile details, give item preferences,rate items
View Recommendations
Show Recommendations
Generate Recommendations
unsuccessful three times
Log Out
System sends a ranked list of recommendations to user
Recommender Engine generates a ranked list of recommendations

Figure 4. The Activity Diagram

5. EXPERIMENTS AND RESULTS

5.1 Dataset

The MovieLens (http://www.grouplense.org) 100k dataset was used. This data was collected by the GroupLens Research Project at the University of Minnesota during a seven-month period between 19th September 1997 and 22nd April 1998. The MovieLens is used mainly because it is publicly available and has been used in many hybrid recommender systems and therefore considered a good benchmark for this purpose. This dataset contains 943 users, 1682 movie items and 100000 ratings. Each user rates a minimum of 20 movies using integer values 1 to 5 and not all movies are rated by all users. There are 19 movie genres. A movie can belong to more than one genre. A binary value of 1 and 0 is used to indicate whether a movie belongs to a specific genre or not. The dataset is split into 5 subsets, each having (80%) training and (20%) test sets.

5.2 Evaluation metrics

The evaluation was done using prediction accuracy metric: Mean Absolute Error (MAE), which is used to represent how accurately a RS estimates a user's preference for an item. MAE is calculated by averaging the absolute deviation of a user's predicted score and actual score. The smaller the MAE the more precise the RS.

$$\text{MAE} = \frac{\sum_i^n |s_i - p_i|}{n} \qquad (22)$$

Where, n is the total number of items, i is the current item, s_i is the actual score a user expressed for item i, and p_i is the RS's predicted score a user has for i.

In this experiment 5-fold cross validation was performed on sub datasets 1 to 5 provided by MovieLens 100k dataset, 80% training data and 20% test data on each sub dataset. This experiment compares the results of the hybrid approach to CBF and CF methods implemented separately.

5.3 Results

Even though the 5 sub data sets used have almost the same number of users and items, they have different rating patterns therefore a standard number of users and items were used for experiment across all the datasets. Results presented here are the average of MAE across all the sub data sets given the specified number of users and items.

.

Table 2. Average MAE given 100 items

No of Users	Filtering Methods		
	CF	CBF	HF
100	0.3686	0.3828	0.3433
350	0.3374	0.3632	0.3162
500	0.3398	0.3659	0.3161
800	0.3258	0.3555	0.3081

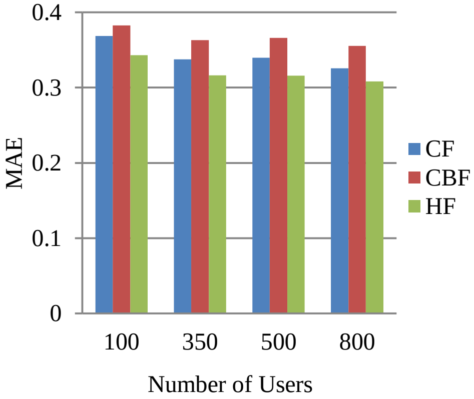


Figure 5. MAE given 100 items

Table 3. MAE given 500 items

No of Users	Filtering Methods		
	CF	CBF	HF
100	0.3396	0.3588	0.3043
350	0.3110	0.3560	0.2998
500	0.3016	0.3544	0.2954
800	0.2971	0.3519	0.2953

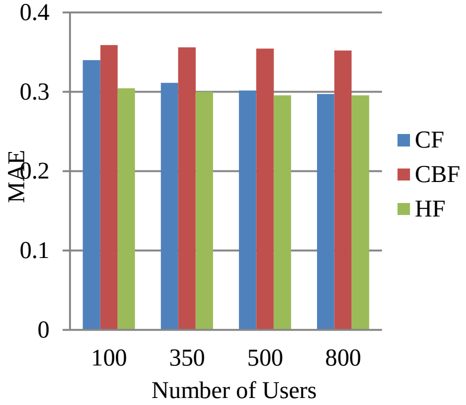


Figure 6. MAE given 500 items

Table 4. MAE given 700 items

No of Users	Filtering Methods		
	CF	CBF	HF
100	0.3326	0.3554	0.3029
350	0.3324	0.3690	0.3167
500	0.3203	0.3676	0.3122
800	0.3020	0.3564	0.2986

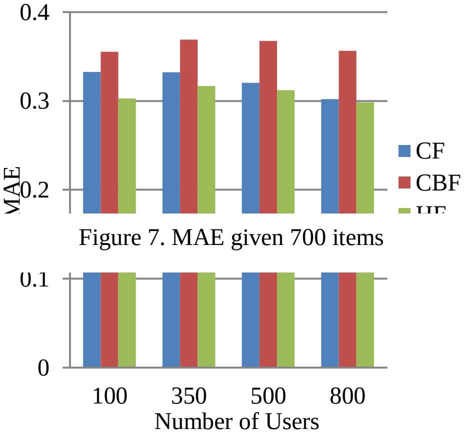


Figure 7. MAE given 700 items

Table 5. MAE given 1200 items

No of Users	**Filtering Methods**		
	CF	**CBF**	**HF**
100	0.3374	0.3539	0.305
350	0.3386	0.3859	0.3245
500	0.3294	0.3745	0.3145
800	0.2997	0.3442	0.2883

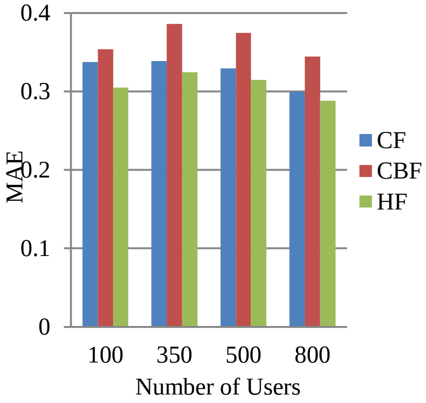


Figure 8. MAE given 1200 items

Collaborative filtering contributes greatly in the results of this approach more so where there are large numbers of items; its performance becomes better with increasing number or items and users respectively, but does not perform as well with large number of users and small number of items. On the other hand, content-based filtering does not make much contribution to this approach, its performance worsens as the number of items increases and its prediction is worse in cases where there are small number of users and items respectively.

However, across all of the evaluations, results show that the hybrid filtering model achieves better prediction accuracy than each of the traditional filtering methods implemented separately. Collaborative filtering and content-based filtering performing on average 6% and 17% worse than the hybrid approach respectively; the hybrid approach achieves an average MAE of 0.3084 whereas collaborative and content-based filtering achieve 0.3258 and 0.3622 respectively.

6. CONCLUSIONS AND FURTHER WORK

In this paper a hybrid approach that combines content-based and collaborative filtering methods has been used to improve recommendation accuracy. Both methods use the effective information retrieval model the VSM, a very simple efficient ranking algorithm TFIDF and cosine similarity measure to find the relationships among users, items and attributes. The evaluation of the proposed hybrid model using real data has proven it achieves better prediction accuracy compared to a single content-based and single collaborative based recommender system. Because of this good performance, this hybrid recommendation approach and the information retrieval methods can therefore be adapted in different domains for recommendation purposes.

The possible future work related to this study is first to test the efficiency of this approach to other larger datasets and secondly, to explore the possibilities of experimenting with other variants of tfidf, similarity measures and the vector space model to see how well they perform in this kind of hybrid recommender environment.

7. REFERENCES

[1] Gavgani V.Z. “Health Information Need and Seeking Behavior of Patients in Developing Countries' Context; an Iranian Experience,” Proceedings of the 1st ACM International Health Informatics Symposium, 2010, paper 11–12, pp. 575–579.

[2] Kazienko P., Kołodziejski P. 2005. ‘WindOwls – Adaptive Systems for the Integration of Recommendation Methods in E – commerce’, Springer Verlag, 218 – 224.

[3] Adomavicius G., Tuzhilin A. 2005. Toward the next generation of recommender systems:

A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. (2005); 17:734-749.

[4] Tuzhilin A., Adomavicius G., 2005. Towards the next generation of recommender systems. A survey of the state of the art and possible extensions. IEEE Trans Knowl Data Eng, 17:734 – 749.

[5] Bell R., Koren Y., and Volinsky Ch. Chasing $1,000,000: How we won the Netflix Progress Prize. ASA Statistical and Computing Graphics Newsletter, 18(2):4–12, 2007.

[6] Burke R. Hybrid recommender systems: survey and experiments, User Modeling and User Adapted Interaction 12 (4) (2002) 331 – 370.

[7] Melville P., Mooney R., Nagarajan R. Content-boosted collaborative filtering for improved recommendations, in:18th National Conference on Artificial Intelligence (AAAI-02), 2002, PP. 187-192.

[8] Park S. T., Pennock D., Madani O., Good N., DeCoste D. Naïve filterbots for robust cold start recommendations, in: KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Minning, 2006, pp. 669-705.

[9] Spiegel S., Kunegis J., Li F. ”Hydra: a hybris recommender system [cross-linked rating and content information] in CIKM-CNIKM, 2009, pp. 75-80.

[10] Pazzani M. J. “A framework for collaborative, content based and demographic filtering,” Artfi.Intell. Rev., vol. 13, no. 5-6, 1999, pp. 393-408.

[11] Melville P., Mooney R. J. Nagarajan R. “Content boosted collaborative filtering for improved recommendation,” in proceedings of AAAI/IAAI, 2002, pp.187 – 193.

[12] Basu C., Hirsh H., Cohen C. Recommendation as classification: Using social and content-based information in recommendation. In Proceedingsof the Fifteenth National Conference on Artificial Intelligence (AAAI-98), 1998, pp 714–720.

[13] Pazzani A., Michael J. A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 1999, 13(5-6):393–408.

[14] Marko B., Yoav S. Fab: Content-based, collaborative recommendation. Communications of the Association for Computing Machinery, 1997,40(3):66–72.

[15] Good N., Schafer J. B., Konstan J.A., Borchers A., Sarwar B., Herlocker J., Riedl J. Combining collaborative filtering with personal agents for better recommendations. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), 1999, pp 439–446.

[16] Jahrer M., Toscher A., Legenstein R. “Combining predictions for accurate recommender systems,” in Proceedings of the SIGKDD conference. New York, NY, USA: ACM 2010, pp. 693-702.

[17] Burke R. D. “Hybrid recommender systems” survey and experiments,” User Model, User-Adapt. Interact, vol 12, no. 4, 2002, pp. 331-370.

[18] Montaner, M., Lopez, B. and De la Rosa J.L.2003. ‘A Taxonomy of Recommender Agents on the Internet’, Artificial Intelligence Review, Kluwer Academic Publisher, 19, 285 – 330.

International Journal of Computer Applications Technology and Research
Volume 5–Issue 12, 775-781, 2016, ISSN:-2319–8656

Review of SIP based DoS attacks

Abdirisaq M. Jama
Department of Telecommunication, Faculty of Engineering
Open University Malaysia(OUM)
Kuala lumpur, Malaysia

Othman O. Khalifa
Department of Electrical and Computer Engineering
International Islamic University,
Malaysia

Abstract: The Voice over Internet Protocol (VoIP). The VoIP is relatively new and is gaining more and more popularity as it offers a wide range of features and is much more cost effective as compared to the traditional PSTN. But the VoIP brings with it certain security threats which need to be resolved in order to make it a more reliable source of communication. Session Initiation Protocol (SIP) today is considered the standard protocol for multimedia signaling, and the result is a very generic protocol. SIP is specified by the IETF in RFC 3261. From a structural and functional perspective, SIP is application layer signaling text-based protocol used for creating, modifying, and terminating multimedia communications sessions among Internet endpoints. Unfortunately, SIP-based application services can suffer from various security threats as Denial of Service (DoS). attacks on a SIP based VoIP infrastructure that can severely compromise its reliability. In contrast, little work is done to analyze the robustness and reliability of SIP severs under DoS attacks. In this survey, we are discussing the DoS flooding attack on SIP server. Firstly, we present a brief overview about the SIP protocol. Then, security attacks related to SIP protocol. After that, detection techniques of SIP flooding attack and various exploited resources due to attack were discussed and finally the paper reviews previous work done on SIP based DoS attacks.

Keywords: Voice over IP; Session Initiation Protocol; attack; security; Denial of Service

1. INTRODUCTION

VoIP is a technology which allows users to use telephone services using Internet connection in IP based network.
These telephone services are provided by the Public Switched Telephone Network(PSTN)

The fundamental process of VoIP includes conversion of voice into digital signals with the segmentation of voice signals into a stream of packets and then sending those voice packets across the network using Real Time Transport Protocol (RTP) [1].

All we need to make a VoIP call is a microphone, speakers and an internet connection. The main advantages of using Internet to make calls is that (i) it is very cheap as compared to the traditional PSTN system of making calls (ii) offers a rich feature set and (iii) is highly flexible.

Session Initiation Protocol (SIP) is the IETF standard for IP telephony and it is defined in RFC 3261 as an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants [2].

SIP is structured as a layered protocol, which means that its behavior is described in terms of a set of quite independent processing stages with only a loose coupling between each stage.

The lowest layer of SIP is its syntax and encoding and the second layer is the transport layer. It defines how a client sends requests and receives responses, and also, how a server receives requests and sends responses over the network. The third layer is the transaction layer. Transactions are the fundamental component of SIP. A transaction is a request sent by the client transaction layer to the server transaction layer, along with all responses to that request which are sent from the server transaction layer back to the client transaction layer. The transaction layer handles application-layer re-transmissions, matching of responses to requests, and application-layer timeouts. The layer above the transaction layer is called the transaction user (TU). When a TU wants to send a request, it creates a client transaction instance and passes the request along with the destination IP address, its port, and its transport layer information as shown in figure 1.

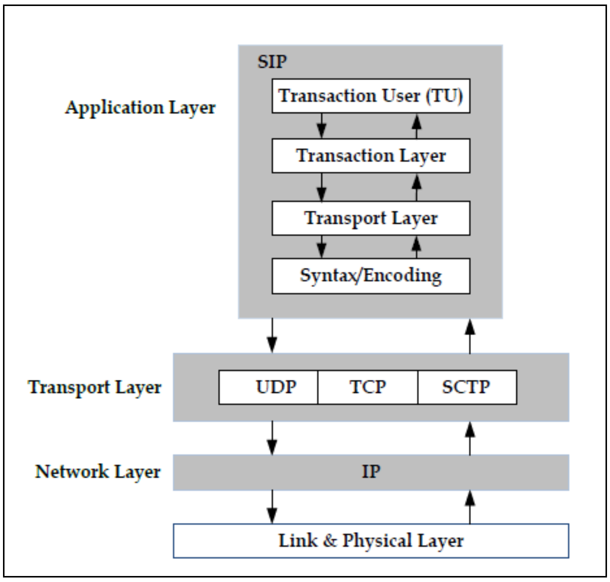


Figure. 1 SIP Protocol layers

SIP is a lightweight application layer protocol designed to manage and establish multimedia sessions such as video conferencing, voice calls, and data sharing through requests and responses. It is increasingly gaining favor over H.323 in the VoIP environment. Three advantages of SIP are as follows [2] :-

- It uses Uniform Resource Locators (URL): addressing scheme, which is physical location independent. Addressing can be a phone number, an IP address, or an e-mail address.
- It allows multiple media sessions during one call. This means that users can share a game, Instant Message (IM), and talk at the same time.
- It is a "light" protocol and is easily scalable.

VoIP has gained tremendous acceptance and is widely deployed today mainly due to the reduced costs, demand for multimedia communications, and demand for convergence of voice and data networks.

Securing VoIP is not an easy task, as it needs efforts in several stages. One of the essential issues in VoIP security is protecting the signalling messages being exchanged between VoIP infrastructures. As it is built on standard IP networks, it is vulnerable to the wide range of network attacks associated with the Internet, such as Denial of Service (DoS).

Much attention is paid to enhance the features and interoperability of SIP protocol with less focus on security. A SIP based VOIP network is potentially vulnerable to general IP and VOIP attacks as well as attacks which are unique to SIP. To secure a SIP based VOIP system, it is necessary to understand the nature of different kinds of attacks and how they can affect to degrade the performance of a SIP system. Many solutions and strategies have been proposed to solve SIP based VoIP security issues.

This paper attempts to explore the SIP based DoS security issues. The following section presents the general components of the SIP architecture. Section 3 addresses the security requirements and the possible threats and attacks in SIP based VoIP, while it briefly describes SIP's security mechanisms. Section 4 emphasizes DoS attacks and section 5 illustrates related work on SIP DoS attackes and section 6 concludes the paper providing some pointers to future research work.

2. SIP OPERATION

The Session Initiation Protocol is a text-based signaling communications protocol, which is used to creation, management and terminations of each session. It is responsible for smooth transmission of data packets over the network. It considers the request made by the user to make a call and then establishes connection between two or multiple users. When the call is complete, it destroys the session.

SIP can be used for two party (unicast) or multi party (multicast) sessions. It works in along with other application layer protocols that identify and carry the session media.

2.1 SIP Components

SIP is a text based client-server protocol similar to Hyper Text Transfer Protocol (HTTP). A SIP-based VoIP system is composed of the following types of entities [2] as shown in figure 2 each having specific functions to perform:-

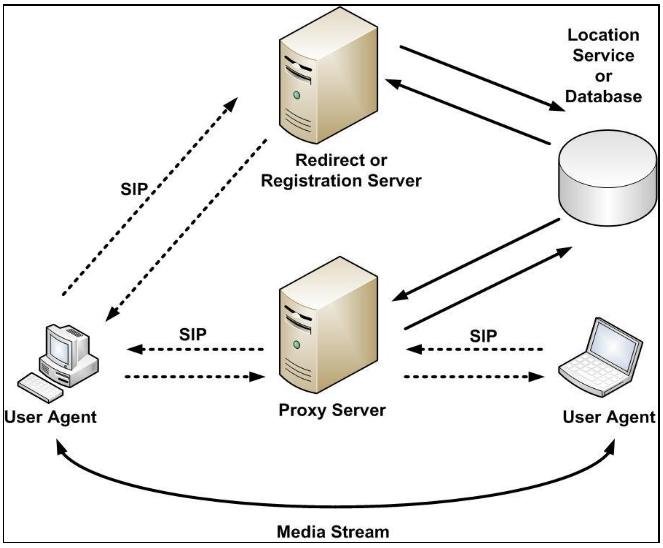


Figure. 2 Basic logical components of a SIP based system

User Agent (UA): is the component interacting with the end user to complete a SIP request. A SIP client can act as both a SIP User Agent Client (UAC) and a SIP User Agent Server (UAS), where the UAC generates outgoing SIP requests which mean that it is the entity that initiates the call, and UAS handles incoming SIP call requests.

SIP proxy server: the SIP proxy server receives SIP requests from various user agents and forwards them to the appropriate hosts. It may also contain an authentication function. It is used as mediators in VoIP and forwards requests to UAS, UAC or other proxies. A proxy server is often responsible for a domain that a client is registered to. A Proxy may enforce a policy and for example, verify that a user is allowed to initiate a call. A proxy can also interpret and if necessary, rewrites specific parts of a SIP request before forwarding it.

Location Server: A location server is used to store the locations of registered users. It is used by a proxy to find the destination client's possible location.

Redirect server: Redirect server accepts SIP request from a client, maps the SIP address of the called party and returns the address to the client. Redirect Server doesn't forward request to other servers.

Registrar server: It processes REGISTER messages, and it maps the users URI to their current location. It is a server that accepts register requests from a UA and stores the information into a location service in the domain it handles. When an UAC wants to initiate a session with a UAS, UAC must discover the current host (IP address) where the UAS is reachable. This discovery process is often done by SIP proxy servers and redirect servers which are responsible for receiving a request, determining where to send it based on knowledge of the location of the user, and then sending it there. To do this, SIP network elements asks the location service, that responds with a UA address within a particular domain.. In some systems, the registrar server is located on the SIP proxy server.

2.2 SIP Messages

In [2] defines the various types of messages that the SIP can support. The SIP messages fall widely under two categories, Requests and Responses. Some of the SIP supported Requests and Responses are listed in Table 1 and 2 respectively.

Table 1. SIP requests

SIP Request	Purpose
INVITE	To initiate a session
BYE	To terminate an existing session
OPTIONS	To determine the SIP messages and codecs that the UA or server understands
REGISTER	To register a location from a SIP user
ACK	To acknowledge a response from an INVITE request
CANCEL	To cancel a pending INVITE request (it is important to note that this operation does not affect a completed request
SUBSCRIBE	To indicate the desire for future NOTIFY requests
NOTIFY	To provide information about a state change that is not related to a specific Session
REFER	To transfer calls and contact external resources

Table 2. SIP responses

SIP Response	Purpose
100 Trying	To indicate a proxy has received an INVITE request, and is processing it.
180 Ringing	The INVITE has been forwarded to the destination
200 OK	A session has been set up
401	A response to a REGISTER request, if the user did Unauthorized not provide correct authentication information
407	Proxy Authentication Required A response to an INVITE request, if authentication is enabled on the proxy, and the user did not provide correct authentication information
408	Request timeout To indicate there is no response to a request within a certain time
503	Service unavailable. To indicate the current request cannot be processed

2.3 SIP Process

The SIP operation is introduced as a specific example. Communication between Alice and Bob is used to explain SIP operation. Besides, their end to end controls. An initial request starts from SIP server. It may be used as a user agent server. Otherwise, it will act as proxy server. The SIP proxy server was considered as the example here, for SIP signaling it should pass through SIP proxy server. When Alice log on to her SIP soft phone or hard-phone first step will be to register to the server sending invite messages, the server will response to Alice by informational trying, then proxy server will forward a second trying which will be received by Bob's telephony device. Bob will ring his phone. The assumption is that Bob will pick up his incoming call, the message will be send for both SIP proxy server and Alice. When the SIP messages request succeeds, Final response to the INVITE "ACK" will be sent from Alice to Bob as illustrated in Figure 3.

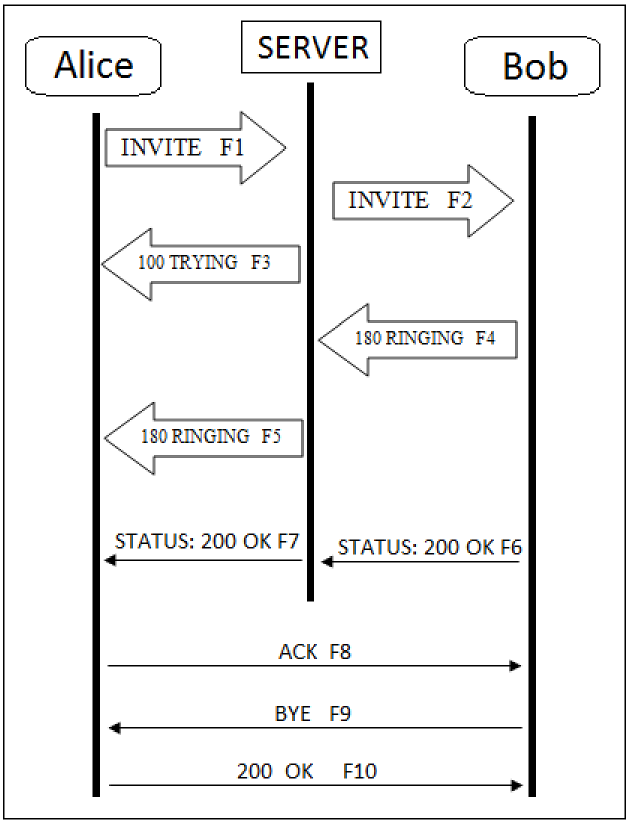


Figure. 3 SIP Process

So, the media will start unless Bob ends the call. Thus, message from Bob will inform the server BYE and server will forward his request to Alice. After that, final agree message will be exchanged. RTP works point to point even if there are SIP proxy servers. SIP normally uses Real Time Protocol (RTP). The purpose is to establish a media session.

3. SIP BASED VOIP SECURITY

Security and privacy requirements in a VoIP environment are expected to be equivalent to those in PSTN, even though the provision of secure Internet services is much more complicated. SIP messages may contain information that a user or a server wishes to keep private.

The flexibility and rich feature-set of SIP based IP telephony compared to traditional PSTN based phone comes with the additional security risks. SIP based IP telephony system is vulnerable to general Internet attacks, as well as attacks which are specific to SIP. As most of SIP development so far has focused on features and interoperability, there exists ample opportunity to work on SIP security. Various security attacks and threats applicable to SIP are discussed in the following subsections.

3.1 SIP Vulnerability issues

As with any other network protocol, SIP is exposed to a wide range of security attacks. When deployed in a private network where network equipment and users are trustworthy and physical security is agreeably sufficient, SIP security may not be needed. However, since SIP can be deployed in an unreliable and untrustworthy environment like Internet, it is susceptible to various security attacks that include the common TCP/IP attacks.

The following Table 3 illustrates what people need to secure their network and how it may be vulnerable.

Table 3. Vulnerability issues

What You Must Do	What Hackers Can Do
Protect every point of entry Attack the weakest point of entry	Attack the weakest point of entry
Be constantly vigilant 24 / 7 / 365	Attack at a time of their choosing
Close every vulnerability	Exploit all vulnerabilities
Close every known vulnerability	Search for new vulnerabilities

For instance, VoIP suffers from all known attacks associated with any Internet application or subsystem. Table 4 illustrates some of the identified threats - attacks, their impact on the overall SIP security.

Table 4. Network and application security issues

Issues	Impact
Eavesdropping: Unauthorized interception decoding of signaling messages	Loss of privacy and confidentiality
Viruses and Software bugs	DoS / Unauthorized access
Replay: Retransmission of genuine messages for reprocessing	DoS
Spoofing: Impersonation of a legitimate user	Unauthorized access
Message tampering/Integrity: The message received is the message that was send	Loss of integrity, DoS
Prevention of access to network services e.g. by flooding SIP proxy servers / registrars	DoS
SIP-enabled IP phones: Trivial File Transfer Protocol (TFTP) Eavesdropping, Dynamic Host Configuration Protocol (DHCP) Spoofing, Telnet	Loss of confidentiality, Unauthorized access, DoS

4. SIP DOS ATTACKS

SIP system is deployed in the Internet that can be Considered hostile environment, in which SIP messages may be exposed to a range of security threats and attacks. Following are some classified attacks on the SIP protocol.

Denial-of-Service (DoS) attacks are a class of network attacks performed to interrupt or terminate applications, servers, or even whole networks, with the aim of disrupting legitimate users' communication. Disruption targets are web browsing, listening to online radio, or even interrupting essential communication, e.g. power plant network control traffic. DoS attacks are commonly performed intentionally and in most cases difficult to counter.

Several components in a VoIP system, including media gateways, IP phones, IP PBX, VoIP firewalls and so on process signaling, causing DoS against the signaling interfaces to be a major issue [3].

DoS attacks can have different forms, and they can also be differently motivated. Generally, users might like the feeling of having power to force their will onto others by disrupting some sort of their communication.

The goal of a DoS attack is to render the service or system inoperable. Hence an attack can be directed toward different entities in the network, depending on the attacker's intent. If the aim is to render the service as a whole inoperable, the main target will be the core servers in the SIP infrastructure.

Three different types of SIP DoS attacks were classified. They are SIP Message Payload Tampering, SIP Message Flow Tampering and SIP Message Flooding. A classification, to be illustrated below and as depicted in Figure 4 below.

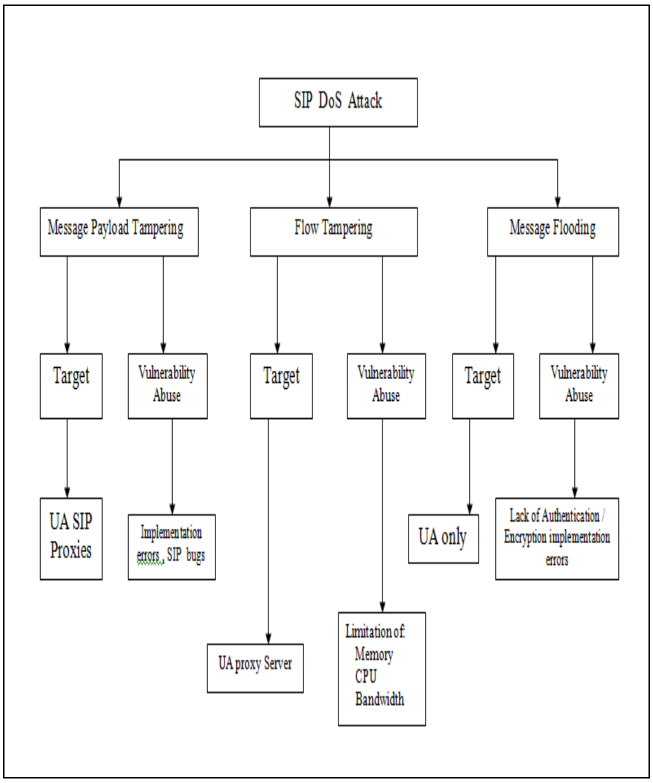


Figure. 4 Classification of SIP DoS attacks.

4.1 DoS by Message Payload Tampering

The first class of attacks is based on tampering with the actual SIP message or more specifically, the SIP payload. SIP is a text-based protocol and messages are transported usually with clear text.

On the other hand, it makes the implementations of the protocol vulnerable to malformed message attacks. SIP message parsers that process incoming messages have to be efficient to handle the degree of protocol flexibility and also be robust against malformed message attacks.

Attackers can try to inject harmful content into a message, e.g. by entering meaningless or wrong information with the goal of creating a buffer overflow at the target. Also, such messages can be used to probe for vulnerabilities in the target. Harmful code that will be executed in an unforeseen context can be introduced into the payload.

This kind of vulnerability exploitation, an attacker sends messages crafted in a specific way that takes advantage of that given vulnerability. By launching a Nuke attack on e.g. the TCP/IP stack, the whole system might crash eventually. Such vulnerabilities are easy exploited by an attacker, but also easily eliminated. As soon as the vulnerability is detected, it can be fixed by modifying the source code. Usually, vendors provide patches for their software soon after a new exploit has become known. The local system administrator then has to install the patch to prevent further attacks.

4.2 DoS by FlowTampering

One common way to achieve a DoS attack is to exploit vulnerabilities in a software component on the target machine. This includes vulnerabilities in application servers, network stacks, or general operating system vulnerabilities.

Flow based DoS attacks aim at causing a disruption to an ongoing call by impersonating one of the call participants. The SIP protocol defines a specific sequence of message exchanges for call setup and termination [2].

The attacker needs to know the session parameters in order for these attacks to function correctly. The parameters can be sniffed from the network. By sending a message out of its expected sequence, an attacker can disrupt the regular call flow. Such attacks are mostly targeted at SIP User Agents.

There are two common strategies to launch a DoS attack, by either exploiting software vulnerability or by depleting resources at the target host. This type of attacking a Resource is to overwhelm a resource at the target. The attack tries to overwhelm resources at the target by generating more requests than the target can handle.

There are three common resources an attacker can exploit:
1. Memory
2. CPU power
3. Link Bandwidth

4.3 DoS by Message Flooding

The most common incarnation of a DoS attack is where an attacker sends a huge amount of SIP related messages to a target with the goal to overwhelm the target's processing capabilities, and hence rendering the target inoperable.

Message flooding DoS attacks are the most common attacks on the SIP architecture. REGISTER floods are aimed at the SIP registrar, INVITE floods target the SIP proxy/redirect server and authentication DoS affects either or both. DoS attacks are easy to launch requiring an attacker to simply craft a SIP message and send it.

Such attacks are generally hard to detect, as the utilized attack messages are usually valid messages and thus not easily distinguishable from regular messages.

5. RELATED WORK

There are different studies which focus on security issues and countermeasures of SIP based DoS.

In [5] focuses on SIP DoS attacks, The study examine how SIP flooding attacks affect the performance of a SIP-based system, and propose an Improved Security-Enhanced SIP System (ISESS) to counter such attacks. Experimental results are provided to demonstrate the effectiveness of ISESS. The Experimental results show that with ISESS, during a flood-based denial of service attack.

In [6] describes DoS attack are realized by people for key security issues and also it is implemented to increases the security threat, protecting systems against DoS attack. The fast growing concern are improved by DoS attack which are noticed with more researchers where the attacker design a flow or system bug to report as a resource of a victim system, and also users can prevent from accessing the service or to degrade the quality of service which they get. For example, the operating systems with DoS were early work with type of resource exhaustion attack. The services are to be exhausted when supposed to be not available. The computer or network resource exists by DoS attack to avoid damage, e.g. a user account or network connection. The resource availability, and the affected will users are collate by attack. The DoS attack is not only necessary at the unique one but also materialized to resource exhaustion.

In [7] aims to provide scheme to detect low rate SIP flooding attacks using area under curve of monitored dynamic SIP traffic with classification of SIP flooding attacks and its influence on SIP server under low rate DoS attack. Compared to the other detection technique our technique is better, due to its advantages of accuracy, fast, light weight, and flexibility to deal with DoS attack detection. Experimental results show the effectiveness of the scheme. the only drawback of this study is that it detects of low rate attacks and prevention methods are missing.

In [8] proposes a new hybrid (anomaly and misuse) SIP flooding attack detection algorithm, which overcomes the existing problems in many of other detection algorithms & is better than existing algorithms. The proposed algorithm is tested using simulated traffic datasets, and compared with three well known anomaly algorithms and one misuse detection algorithm. The test results show that the new algorithm has high detection accuracy and high completeness.

Another study in [9] , The authors built and configured a real test-bed for SIP based services to generate normal and assumed attack traffics. the test-bed was validated and evaluated our intrusion detection system with the dump traffic of this real test-bed and we also used another specific available dataset to have a more comprehensive evaluation. The experimental results show that the approach was effective in classifying normal and anomaly traffic in different situations. The Receiver Operating Characteristic (ROC) analysis is applied on final extracted results to select the working point of our system. the only drawback of this study was that the authors only focused for detection which lacks prevention mechanisms.

In [10] authors proposed a VoIP-aware attack-detection scheme. The proposed scheme is able to detect VoIP network attacks including VoIP DoS and SPAM. It can detect VoIP DoS attacks with low false negatives using a statistics-based detection algorithm and can recognize SPAM with low false positives using a caller behavior-based detection algorithm. Authors have included experimental results to confirm the proposed scheme. this study focused detection mechanisms only.

In [11] proposed a two layer DoS prevention architecture that handles both SIP flooding and malformed packet attacks on a standard VoIP network hence real network topology simulation test is missing.

In [12], proposed a stateful SIP inspection mechanism, called SIP VoIP Anomaly Detection (SIPAD), that exploits a SIP-optimized data structure to detect malformed SIP messages and SIP flooding attacks. SIPAD pre-compiles a stateful rule tree that rearranges the SIP rule set by hierarchical correlation. On the basis of current state and the message type, SIPAD computes the corresponding branches from the stateful rule tree, and examines a SIP message's structure by comparing it to the branches. The SIPAD provides higher detection accuracy, wider detection coverage and faster detection than existing approaches. Conventional SIP detection schemes tend to have high overhead costs due to the complexity of their rule matching schemes. Experimental results of their SIP-optimized approach, by contrast, indicate that it dramatically decreases overhead and can even be deployed in resource-constrained environments such as smartphones. However, this study lacks prevention techniques.

In [13], has given more priority to DoS attack by flooding of different SIP-messages. A small work is done to analyze the performance of SIP server and quality of ongoing VoIP calls under DoS attacks. We show the utilization of CPU and memory during the multiple simultaneous calls. On the basis of measurements we show that a standard SIP server can be easily overloaded by simple call requests. It also shows that simple call request can degrade quality of ongoing calls.

The study proposed in [14] detects DoS attacks using an entropy-based IDS. In such a system, however, an attacker can sniff the network and obtain an entropy value.
In other words, entropy-based DoS solutions are vulnerable to spoofing attack because an attacker can keep the entropy value within an expected range and, therefore, provide realistic conditions to DoS attacks to occur. First, the attacker monitors the entropy before launching the attack and then calculates the mean, standard deviation and variance values. Subsequently, it spoofs the entropy during the attack. The authors show that this detection system can be deceived because the spoofed packets not only penetrate into the network but also help DoS attacks to occur.

6. CONCLUSION

IP is not an easy signaling protocol to secure. A discussion of some present solutions for SIP security malfunctions consisting of implementations and simulations is presented in this study. The SIP security solutions identified suggest that security mechanisms cannot provide 100% protection against SIP attacker, but threats can be mitigated significantly. A number of studies were reviewed and some common problems and their solutions were presented. Several SIP security solutions were found to be ultimately related to device security. The solutions presented here are not achieved by securing a single protocol but should involve the whole system.

7. REFERENCES

[1] Anchal Sehgal, Dervish Ghosh and Dr.Charu Gandhi. 2015. Literature Survey of VoIP Security International Journal of Emerging Technology and Advanced Engineering , Volume 5, Special issue 1, April 2015

[2] IETF Network Working Group. (2016) . SIP: Session Initiation Protocol. Retrieved September 21, 2016. http://www. ietf.org/rfc/rfc3261.txt.

[3] Liu, Z.H., J.C. Chen and T.C. Chen. 2009. Design and analysis of SIP-based mobile VPN for real-time applications. IEEE Trans. Wireless Commun., 8: 5650-5661. DOI: 10.1109/TWC.2009.090076

[4] Tarendra G. Rahangdale1, Pritish A. Tijare and Swapnil N.Sawalkar. 2014. An Overview on Security Analysis of Session Initiation Protocol in VoIP network, International Journal of Research in Advent Technology, Vol.2, No.4, April 2014 E-ISSN: 2321-9637

[5] Xianglin Deng, Malcolm Shore. 2009. Advanced Flooding Attack on a SIP Server. IEEE Computer Society,Page No(647- 652), 2009.

[6] Lin Fan, 2010. “A Group Tracing and Filtering Tree for REST DDoS in Cloud Computing”, International Journal of Digital Content Technology and its Applications, 4(9).

[7] Abhishek Kumar, Dr. P. Santhi Tilagam,” A Novel Approach for Evaluating and Detecting Low Rate SIP Flooding Attack” International Journal of Computer Applications,Volume 26– No.1,Page No (0975 – 8887),July 2011.

[8] Dahham Allawi, Alaa Aldin Rohiem, Ali El-moghazy and Ateff Ghalwash,”New Algorithm for SIP Flooding Attack Detection”,IJCST , Volume- 4, Issue- 3, Page No(10-19), March 2013.

[9] Zoha Asgharian; Hassan Asgharian; Ahmad Akbari and Bijan Raahemi Detecting Denial of Service Message Flooding Attacks in SIP based Services Electrical & Electronics Engineering Journal / Vol . 44 / No.1 / Spring 2012

[10] Jonghan Lee & Kyumin Cho & ChangYong Lee & Seungjoo Kim , VoIP-aware network attack detection based on statistics and behavior of SIP traffic, Peer-to-Peer Netw. Appl. (2015) 8:872–880

[11] S. Ehlert, G. Zhang, D. Geneiatakis, G. Kambourakis, T. Dagiuklas, J. Markl, and D. Sisalem. (2012). Two layer Denial of Service prevention on SIP VoIP infrastructures. Computer Communications. Page No(2443-2456).

[12] Dongwon Seo ,Heejo Lee , Ejovi Nuwere ”SIPAD: SIP–VoIP Anomaly Detection using a Stateful Rule Tree”,Elsevier,Computer communication,Page No(562-574),2013.

[13] Abhishek Bansal, Prashant Kulkarni, Alwyn R. Pais” Effectiveness of SIP Messages on SIP Server”, Proceedings of 2013 IEEE International Conference on Information and Communication Technologies,Page No(251-256),2013.

[14] Ozcelik I, Brooks RR. Deceiving entropy based DoS detection. Computers and Security 2015; 48: 234–245.