ISSN 2319-2720
Volume 7, No.2, April – June 2018
International Journal of Computing, Communications and Networking
Available Online at http://www.warse.org/ijccn/static/pdf/file/ijccn18722018.pdf
https://doi.org/10.30534/ijccn/2018/18722018

Botnet Identification with Malware File Detection & Blocking

Pawanraj S P [1], Ravi B [2], Ranjan C M [3], Dr. Arun Biradar [4]
[1] Final Year B.E in CSE, EWIT, India, pawanrajcse@gmail.com
[2] Final Year B.E in CSE, EWIT, India, ravibcse1996@gmail.com
[3] Final Year B.E in CSE, EWIT, India, ranjangowda1996@gmail.com
[4] Professor & Head, Department of CSE, EWIT, India, hodcsea@gmail.com

ABSTRACT

Botnets are the foremost common vehicle of cyber-criminal activity. They're used for spamming, phishing, denial-of-service attacks, brute-force cracking, stealing non-public data, and cyber warfare.
In this work, we offer basically these contributions:
We use traffic monitoring technique to gather traffic flow information. We implement an inference algorithm for botnet detection. We develop a program to detect & block the malware file. We detect the original IPv4 address of the bot in the botnet.

Key words **–** Botnets, Server, Client, Cyber Security, Network Security, Bot Master, Servent Bot, Traffic Monitoring, Course Grained, C&C (Command & Control), DoS (Denial of Service), DDoS (Distributed Denial of Service), GUI (Graphical User Interface), HTTP (Hyper Text Transfer Protocol) IPv4 (Internet Protocol version 4), P2P (Peer to Peer), TCP (Transmission Control Protocol), UDP (User Datagram Protocol).

1. INTRODUCTION

Network Security consists of the policies and practices adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources.
Cyber Security refers to the technologies and processes designed to protect computers, networks and data from unauthorized access, vulnerabilities and attacks delivered via the Internet by cyber criminals.

One of the most popular threats is the Denial-of-Service (DoS) attack, which can be broadly categorized as a *volumetric* attack, where the target destination is overwhelmed by a huge number of requests, eventually leading to the impossibility of serving any of the users.
In particular, with a Distributed DoS (DDoS) attack, such a huge number of requests is produced in parallel by a net of robots. According to one of the classical DDoS representations, a relatively large ensemble of machines acts cooperatively under the supervision of one or more coordinators.

Botnets are networks of compromised, remotely controlled computer systems. So far, their main purposes include the distribution of spam e-mails, coordination of distributed denial-of-service attacks, and automated identity theft, e.g. credit card information and general banking data for financial fraud. Their presence is supported by the increasing global availability of broadband access to the Internet for network-enabled devices, which at the same time increases the value of the assets they threaten.
A botnet is a collection of Internet-connected user computers (bots) infected by malicious software (malware) that allows the computers to be controlled remotely by an operator (BOT MASTER) through a Command-and-Control (C&C) server to perform automated tasks, such as to steal personal data and passwords, attack public and private networks, exploit users' computing power and Internet access, and carry out Distributed Denial of Service (DDoS) attacks.

Botnets are a complex and continuously evolving challenge to user confidence and security on the Internet. Combating botnets requires cross-border and multidisciplinary collaboration, innovative technical approaches, and the widespread deployment of mitigation measures that respect the fundamental principles of the Internet.

A number of issues must be considered when addressing the problem of botnets. These include:

1.1 Geographic dispersion

Botnets can be widely spread across distance and geography, with infected computers and botnet herders operating in different countries and locations. Same applies to the C&C servers. As such, botnets are transnational and require a collaborative approach to detection, mitigation, and law enforcement.

1.2 Impacts on user rights

It is important to consider the impact on fundamental user rights and expectations when approaching strategies to combat botnets. Overly broad botnet-mitigation strategies, such as blocking all traffic from an infected network, could unintentionally keep innocent users from accessing the Internet and exercising rights, such as freedom of expression and opinion. In addition, some methods to detect and trace botnets, such as the indiscriminate collection of network traffic data, could violate the privacy of legitimate Internet users.

1.3 Impacts on technology use and innovation

Some technical and legal mitigation strategies, such as restricting access to suspected infected networks, may have negative consequences on the openness,

innovation potential, and global reach of the Internet. Further, technology-specific strategies are less likely to address the overall problem of botnets, as their creators may change tactics to avoid new obstacles.

A number of factors contribute to the ongoing challenge of combating botnets, including:

Botnet strategies, technologies, and techniques are constantly evolving and adapting in response to mitigation measures. Botnets have become popular tools for cybercriminals because they are cheap to deploy and operate, hard to uncover, and are available for purchase or rent through criminal networks. Botnet creators and herders are geographically dispersed from the offending bots and are skillful at hiding their locations and identities.

There are vulnerable computers connected to the Internet (e.g., those that are not sufficiently secured or whose users are susceptible to being lured into introducing botnet malware into their computers). Botnet operators actively search for vulnerable systems to infect. Botnets are designed to take advantage of the Internet's fundamental properties (the Internet Invariants3) and its architectural design, where the intelligence is in the end devices (e.g., botnet command and control servers and infected computers) rather than the network itself.

The most important part of a botnet is the so-called command-and-control infrastructure (C&C). This infrastructure consists of the bots and a control entity that can be either centralized or distributed. One or more communication protocols are used by the botmasters to command the victim computers and to coordinate their actions. The sets of instructions and functionality of botnets vary widely with the motivation behind their use.

2. Types of Botnets:-

2.1 Centralized C&C

In a centralized C&C infrastructure, all bots establish their communication channel with one, or a few, single connection points. These are usually command-and-control servers, under the control of the botmaster. Because all bots connect to these servers, botmasters are able to communicate with the bots simultaneously and can issue commands to all the bots that are both online and connected to the botnet.

2.2 Decentralized C&C

In decentralized command-and-control architectures, loosely coupled links between the bots enable communication within the botnet and provide the basis for its organization. A common term for this class of botnets is peer-to-peer botnets, as this is the name of the corresponding network model. The knowledge about participating peers is distributed throughout the botnet itself. Consequently, information about the whole botnet cannot be obtained directly, and commands have to be injected into one peer of the botnet.

3. RELATED WORK

There are three metrics to classify the collected network traffic based on HRU, TGD and dissimilarity of group activities. The first two metrics (i.e. HRU and TGD) examine the collected data for each client in the network where the third metric (i.e. GAD) correlates the network traffic of a single client with its previous network activities along with correlation with other devices activities in a similar network. This is due to the fact that the Botnets are designed for a coordinated form of attack in which the Bots that belong to the same group pose similar activities [1].

BotGM is a new approach for tracking botnet activities with passive network monitoring. It relies on a behavioral graph modeling of NetFlow records. However, rather than building a single massive graph being difficult to analyze, BotGM creates a collection of smaller graphs for comparison with each other and thus indirectly comparing the behavior of end-hosts [2].

PeerHunter is a network traces sampling and mixing method to make the experiments as unbiased and challenging as possible. Experiments and analysis have been conducted to show the effectiveness and scalability of our system [3].

BotShark is a new approach for detection and prevention of Bots in the network. According to experimental evaluation and proposed model it is clear that system fully prevents the Bots as direct communication of end users and web server does not occur. Another thing is as users get response through virtual IP if there is a attacker who is unaware of BotShark will be convinced that the IP is of web server and hence can perform various attacks on such IP which further cannot be executed as such IP or web server will not be exists in the network [7].

The analyzers that monitor a network for synchronized C&C communication, and investigated the potential of using synchronized behavior as an indicator of infection as well as of upcoming attacks. Using traffic from 107 botnet infections representing 13 different botnet families, when compared the detection accuracy of our approach with a commercial blacklist and showed that it achieved at least comparable accuracy [11].

4. DETECTING & TRACKING BOTNET

There are mainly two approaches of botnet detection and tracking methods. One is honeynet based method and the other is based on passive traffic monitoring.

4.1 Traffic Monitoring

It gathers traffic flow information from many vantage points within the network. The core idea is based on the

attack and control chain of the botnet. The major steps are listed as follows:

Identify bots based on their attack activities, such as scanning, emailing of spam and viruses, or DDoS traffic generation. The activities are reported by other security system. Analyze the flows of these bots to find candidate controller connections (CCC). Analyze the CCC to locate the botmaster.

5. HTTP/P2P Botnet

IRC-based botnet has been studied extensively in recent years. The research on the other two kinds of botnet has just begun. The existing works are anatomy of some samples. Their network behaviors have been rarely studied, not to mention the number of botnet and the number of infected hosts. It's likely that more HTTP and P2P botnets will appear in the near future, so we need to pay more attention to them.

6. PROBLEM STATEMENT

The existing system does not possess the following:-
Detecting & blocking the arriving malicious file with the custom made extensions (E.g.:- filename.bot) &
Identification of the ORIGINAL IPv4 address of the bot computer of the botnet.

7. EXISTING CONCEPT

We need flow clustering-based analysis approach to identify hosts that are mostly likely running P2P applications. Approach does not rely on any transport layer used by which can be easily violated by P2P applications.
It is mainly due to the fact that the traffic profile of a bot-compromised host might be completely distorted by the legitimate P2P application running on it simultaneously. For instance, in our experiments, when a host is running a Waledac and a Bitorrent application simultaneously. Existing Technique - BOT Graph.

7.1 Technique definition
A bot-compromised host might be completely distorted by the legitimate P2P application running on it simultaneously Servers are that they represent a single point of failure.

7.2 Drawbacks
There is a fundamental disadvantage of centralized C&C architecture. Servers are that they represent a single point of failure.

8. PROPOSED SYSTEM

In this project a detailed dissection of the botnet's scanning behavior, including general methods to correlate, visualize, and extrapolate botnet behavior across the global Internet.

We use the technique of coarse grained BOTNET detection.
P2P botnets before they are detected in contrast to our approach can detect and profile various P2P applications.

8.1 Merits
We also identify the performance bottleneck of our system and optimize its scalability. We presented a novel botnet detection system that is able to identify stealthy botnets, whose malicious activities may not be observable.

9. SYSTEM ARCHITECTURE & DESIGN

The architecture design of the proposed defense system is shown in Fig.1. The proposed System consists of four phases. They are Traffic Monitoring, Malware File Detection & Blocking, Botnet Detection & Detecting Original IPv4 Address. The proposed System finally produces Bug report, blacklist of IP address.

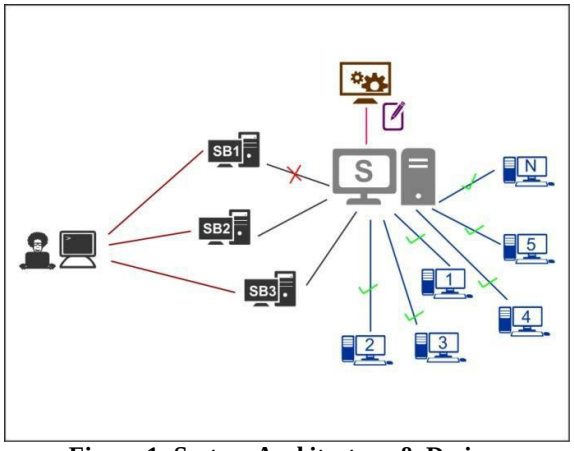

Figure 1: System Architecture & Design

8.1 Main Server
Main Server is a system to which all the client computers are attached. All the related information like client profile database is stored on this server. The server also stores the list of IPv4 addresses of the systems that are declared as bots in the past.

8.2 Client
Client (user) is an all-purpose end user who wishes to use the server for transferring files to other clients. Amongst the clients there is possibility of being an attacker who can penetrate Bots or can perform various attacks.

8.3 Bot Master
A botmaster is a person/computer & client who operates the command and control of botnets for remote process execution.
The botmaster will often hide his/her identify via proxies, TOR and/or shells to disguise their IP

address from detection of investigators and law enforcement.

8.4 Servant Bots

A servant bot(s) is a client computer(s) which performs the activities & executes remote processes as per the command issued by the botmaster. These are the client computers that are controlled by the botmaster.

9. WORKING OF THE PROPOSED SYSTEM

Initially, all the client computers have to register itself to the database of the main server only then the clients can send and receive files between each other. After registration, the clients can login to its respective account to perform send & receive operations. When the users are online, each of the users is notified about the other user who is online. The server initially maintains a Block list of certain IPv4 addresses of the clients who are declared as bots in the past. Those users are blocked and are not allowed to transfer files to the other online users.

When the client is sending the file to another client, the server checks in its block list and if the IP address of the sender is present then the server does not allow the file to reach the receiver. Else the file is allowed to reach the receiver without any barrier. Now, suppose if the IP address of the sender is not there in the block list of server, the file is transferred to the client and if the file is a malware then the receiver computer is infected.

In order to solve this problem, an Antivirus program (software) is developed and installed in the receiver's computer. The security program performs the following operations:-
The program scans the receiving file completely for presence of malware. If malware detected, the file is blocked by the software. The IPv4 address of the sender of the file is fetched and this IP address is sent to the server in order to add the address to its block list to prevent further file transfer to any of the clients in the network. Thus, the proposed system is successful in securing the clients in the network.

10. IMPLEMENTING ALGORITHM

The Botbuster Algorithm

Let us examine how the algorithm works. First, note that a botnet made of one user, besides making little sense in practice, is by definition non-identifiable, since we assumed that the characteristics of the messages at a single-user level do not reveal any special information.

Now, at the beginning of the algorithm, user 1 is initially declared as a bot, namely, $\hat{B} = \{1\}$. Then, it is checked whether users 1 and 2 form a botnet. If so, $\hat{B} = \{1, 2\}$ is taken as the current botnet estimate. If not, $\hat{B} = \{1\}$ is retained.

Algorithm: $\hat{B}_{new}$=BotBuster

$N = \{1, 2, \ldots, N\}$; $\hat{B}_{new} = \Phi$;
for $b0 \in N$ do

$\hat{B} = \{b_0\}$;
for $j \in N \setminus \{b0\}$ do
if $\exists j' \in \hat{B} \cup \{j\}$: $\neg\gamma(\hat{B} \cup \{j\}) \wedge \gamma(\hat{B} \cup \{j\})$ then
$\hat{B} = \hat{B} \cup \{j\}$
end
if $|\hat{B}| > \max(1, |\hat{B}_{new}|)$ **then**
$\hat{B}_{new} = \hat{B}$;
end
end

Table 1: Implementing Algorithm

Then, it is checked whether the currently estimated botnet $\hat{B}$ forms a bot with user 3, and so on. At the end of the inner loop, the algorithm ends up with an estimate $\hat{B}$. If the cardinality of the estimated set is greater than one, it is taken as a current estimate.

The procedure is then restarted by choosing user 2 as initial pivot, and sequentially checking the remaining users as explained before. At the end of the inner loop, the algorithm ends up with another estimate $\hat{B}$. If the cardinality of the estimated set is greater than one and greater than the cardinality of the previously estimated set8, then it is taken as a current estimate. Otherwise, the previous estimate is retained. The procedure ends when all users have been scanned as pivots.

11. IMPLEMENTATION TECHNIQUES

11.1 User Interface Design

In this module we design the windows for the project. These windows are used to send a message from one peer to another. We use the Swing package available in Java to design the User Interface. Swing is a widget toolkit for Java. It is part of Sun Microsystems' Java Foundation Classes (JFC) — an API for providing a GUI for Java programs.

11.2 Coarse Grained P2P Detection

This component is responsible for detecting P2P clients by analyzing the remaining network flows after the Traffic Filter component. For each host *h* within the monitored network we identify two flow sets, denoted as *Stcp(h)* and *Sudp(h)*, which contain the flows related to successful outgoing TCP and UDP connection, respectively. We consider as successful those TCP connections with a completed SYN, SYN/ACK, ACK handshake, and those UDP (virtual) connections for which there was at least one "request" packet and a consequent response packet.

11.3 File Uploading & Sending

This module is used to upload required file from storage device to user account and send the file into destination account. There are many different types of files:

Data files, Program files, Directory files & so on. Different types of files store different types of information.

11.4 BOT Detection

Since bots are malicious programs used to perform profitable malicious activities, they represent valuable assets for the bot master, who will intuitively try to maximize utilization of bots. This is particularly true for P2P bots because in order to have a functional overlay network (the botnet), a sufficient number of peers needs to be always online. In other words, the active time of a bot should be comparable with the active time of the underlying compromised system.

11.5 Clustering and Eliminating

The distance between two flows is subsequently defined as the Euclidean distance of their two corresponding vectors. We then apply a clustering algorithm to partition the set of flows into a number of clusters. Each of the obtained clusters of flows, *Cj (h)*, represents a group of flows with similar size. For each *Cj (h)*, we consider the set of destination IP addresses related to the flows in the clusters, and for each of these IPs we consider its BGP prefix (using BGP prefix announcements).

11.6 Detection of Attacker's IPv4 address

In this module used to determine the geographical location of website visitors based on the IP addresses for applications such as fraud detection. We can find the IP address of the attacker.

CONCLUSION

While botnets are widespread, the botnet research is still in its infancy. This paper surveys state-of-art botnet research that can be categorized into three areas, i.e. understanding botnet, detecting & tracking botnets, and countering against botnets. In understanding botnet research, it is proposed to learn botnet behaviors and characteristics through source code analysis, binary analysis or wide area measurement. Some formal models are also proposed to predict botnet advancement. In detecting & tracking botnet researches, honeynet and traffic monitoring approaches are proposed to detect botnets based on some of their unique behaviors. Finally, the research on defending against botnet proposes to simply shut down botmaster after they are identified. Those current botnet studies are still in a preliminary stage. Previous analysis shows that majority of botnet traditionally used IRC for their command and control. But we believe the botnets will advance to new communication architectures, for example, P2P-based botnet. And currently the defense against botnet is not very efficient, so much more work needs to be done in this field. Finally future botnet prediction may give us an advanced view of the botnet development. Good model can help people know the properties of botnet and thus control it.

REFERENCES

1. Maryam Var Naseri, Wardah Zainal Abidin, Meisam Eslahi. **Correlation-Based HTTP Botnet Detection Using Network Communication Histogram Analysis**, 2017 IEEE Conference on Application, Information and Network Security (AINS), pp. 7-12.
2. Sofiane Lagraa, J´erˆome Franc¸ois, Abdelkader Lahmadi, Marine Miner, Christian Hammerschmidt and Radu State. **BotGM: Unsupervised Graph Mining to Detect Botnets in Traffic Flows**, pp. 1-8.
3. Di Zhuang, J. Morris Chang. **PeerHunter: Detecting Peer-to-Peer Botnets through Community Behavior Analysis**, 2017 IEEE, pp. 493-500.
4. Anton O. Prokofiev, Yulia S. Smirnova, Vasiliy A. Surov. **A Method to Detect Internet of Things Botnets**, 2018 IEEE, pp. 105-108. https://doi.org/10.1109/EIConRus.2018.8317041
5. Victor G. T. da Costa, Sylvio Barbon Junior, Rodrigo S. Miani, Joel J. P. C. Rodrigues, Bruno B. Zarpel˜ao. **Detecting Mobile Botnets Through Machine Learning and System Calls Analysis**, IEEE ICC 2017 Communications Software, Services, and Multimedia Applications Symposium,.
6. Bhan Sengar, Professor. B.Padmavathi. **P2P bot detection system based on Map Reduce**, Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC), pp. 628-634.
7. Ms. Pooja M. Pondkule, Mrs. B. Padmavathi. **BotShark - Detection and Prevention of Peer-to-Peer Botnets by Tracking Conversation using CART**, International Conference on Electronics, Communication and Aerospace Technology ICECA 2017, pp. 291-295.
8. Ms.Amruta Kapre, Mrs. B. Padmavathi. **Adaptive behaviour pattern based botnet detection using traffic analysis and flow interavals**, International Conference on Electronics, Communication and Aerospace Technology ICECA 2017, pp. 410-414.
9. Ahmed A.Awad, Samir G. Sayed, Sameh A. Salem. **A Network-based Framework for RAT-Bots Detection**, pp. 128-133.
10. Massimiliano Albanese and Sushil Jajodia, Sridhar Venkatesan. **Defending from Stealthy Botnets Using Moving Target Defenses,** 2018 IEEE, pp. 93-97.
11. Zainab Abaid, Mohamed Ali Kaafar and Sanjay Jha. **Early Detection of In-the-Wild Botnet Attacks by Exploiting Network Communication Uniformity: An Empirical Study**, 2017 IFIP, pp. 7-12. https://doi.org/10.23919/IFIPNetworking.2017.8264866
12. Reham A. Al-Dayil, Mostafa H. Dahshan. **Detecting Social Media Mobile Botnets Using User Activity Correlation and Artificial Immune System**, 2016 7th International Conference on

Information and Communication Systems (ICICS), pp. 109-114. https://doi.org/10.1109/IACS.2016.7476095
13. Francisco Villegas Alejandre, Nareli Cruz Cort´es, and Eleazar Aguirre Anaya. **Feature selection to detect botnets using machine learning lgorithms**.
14. Elisa Bertino, Nayeem Islam. **Botnets and Internet of Things Security**, 2017 IEEE, pp. 76-79.
15. Koki Hongyo, Tomotaka Kimura, Takanori Kudo, Yoshiaki Inoue, and Kouji Hirata. **Modeling of countermeasure against self-evolving botnets**, 2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), pp. 227-228. https://doi.org/10.1109/ICCE-China.2017.7991078
16. Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, Angelos Stavrou, **DDoS in the IoT: Mirai and Other Botnets**, 2017 IEEE, pp. 80-84.
17. Takanori Kudo, Tomotaka Kimura, Yoshiaki Inoue, Hirohisa Aman, and Kouji Hirata. **Behavior Analysis of Self-Evolving Botnets**, 2016 IEEE. https://doi.org/10.1109/CITS.2016.7546428
18. Zhuo Lu, Wenye Wang, and Cliff Wang, **On the Evolution and Impact of Mobile Botnets in Wireless Networks**, 2015 IEEE.
19. Milan Oulehla, Zuzana Komínková Oplatková, David Malanik. **Detection of Mobile Botnets using Neural Networks**, FTC 2016 - Future Technologies Conference 2016, 6-7 December 2016, 2016 IEEE, pp. 1324-1326.
20. Manuel Gil Pérez, Alberto Huertas Celdrán,and Fabrizio Ippoliti, Pietro G. Giardina and Giacomo Bernini, Ricardo Marco Alaez and Enrique Chirivella-Perez, Félix J. García Clemente and Gregorio Martínez Pérez, Elian Kraja and Gino Carrozzo, Jose M. Alcaraz Calero and Qi Wang. **Dynamic Reconfiguration in 5G Mobile Networks to Proactively Detect and Mitigate Botnets**, 2017 IEEE, pp. 28-36.
21. Aditya K Sood, Sherali Zeadally, Richard J Enbody. **An Empirical Study of HTTP-based Financial Botnets** 2013 IEEE, pp. 1-16.
22. Huan Yang, Liang Cheng and Mooi Choo Chuah. **Detecting Peer-to-Peer Botnets in SCADA Systems**, 2016 IEEE, pp. 7-12. https://doi.org/10.1109/GLOCOMW.2016.7848877
23. Jinxue Zhang, Rui Zhang, Yanchao Zhang, Guanhua Yan. **The Rise of Social Botnets: Attacks and Countermeasures**, 2016 IEEE, pp. 1-14.
24. Daniel Plohmann, Elmar Gerhards-Padilla, Felix Leder. **Botnets: Detection, Measurement, Disinfection & Defence**, pp. 1-12.
25. Zhaosheng Zhu, Guohan Lu, Yan Chen. ***Botnet Research Survey***, Annual IEEE International Computer Software and Applications Conference, pp. 967-972. https://doi.org/10.1109/COMPSAC.2008.205

Pawanraj S P: pursuing B.E in CSE, EWIT (VTU), Bengaluru. His areas of interest are Computer Security, Databases, Computer Networks, Storage Area Networks, Programming the Web, Software Engineering, Cloud Computing, Computer Graphics, etc.

Ravi B: pursuing B.E in CSE, EWIT (VTU), Bengaluru. His areas of interest are Computer Security, Databases, Computer Networks, Software Engineering, Java & Python Programming etc.

Ranjan C M: pursuing B.E in CSE, EWIT (VTU), Bengaluru. His areas of interest are Computer Security, Databases, Computer Networks, Computer Graphics, Software Engineering, Internet of Things etc.

Dr. Arun Biradar: Professor & Head, Department of Computer Science & Engineering, East West Institute of Technology (VTU), Bengaluru. Qualification: B.E, M.Tech, Ph.D, Board of Examiner (Member), VTU, Belagavi. His areas of research are Wireless Ad-hoc Networks, Computer Networks, Software Engineering, Genetic Algorithms, Machine Learning, IoT and Cloud Computing.