Vol.3 No.2 2019

Universidad de Oviedo

CMR
INSTITUTE OF TECHNOLOGY
EXPLORE TO INVENT

ISSN: 2581-3242

International Journal of Machine Learning and Networked Collaborative Engineering

Vol.3 No.2 2019

International Journal of
Machine Learning and Networked Collaborative Engineering

PREFACE

The International Journal of Machine Learning and Networked Collaborative Engineering (IJMLNCE) with ISSN: 2581-3242 is now indexed in popular databases such asBASE (Bielefeld Academic Search Engine), CNKI Scholar, CrossRef, CiteFactor, Dimensions, DRJI, Google Scholar, Index Copernicus, JournalTOCs, J-Gate, Microsoft Academic, PKP-Index, Portico, ROAD, Scilit, Semantic Scholar, Socolar or WorldCat-OCLC. We are now proud to present the eighth volume of the journal, Volume No-03 Issue No-02, with some high-quality papers written by international authors and covering different aspects related to machine learning and collaborative engineering.

Puri et al. published a work entitled “Cloudbin: Internet of Things based waste monitoring system”. In this paper, authors present an IoT-based waste management system called Cloudbin to monitor and control waste garbage in urban areas. To that end, authors use different elements like an ultrasonic sensor, a GPS module or a methane detection mechanism. The problem of waste management is one of the key elements in which governments must take an active part.

Rimal published a work entitled “Machine Learning Prediction of Wikipedia Time Series Data using: R Programming”. In this work, author explains how prediction of automatic learning of Wikipedia time series work using the R environment. To that end, author focused on real data from Cristina Ronaldo, a famous football player, presenting, according to the author, the simplest way to predict times series data and its strengths for data analysis.

Sen et al. published a work entitled “Study of Energy Efficient Algorithms for Cloud Computing based on Virtual Machine Migration Techniques”. This study describes how energy efficiency in cloud computing is one of the most important features to be considered to measure the efficiency of such services, balancing power and quality of the service. Thus, authors discuss how virtual machine migration techniques can help to achieve energy efficiency.

Choudhary published a work entitled “Information Processing in GLIF Neuron Model with Noisy Conductance”. Authors investigate the generalized leaky integrate-and-fire neuron model with stochastic synaptic conductance and investigate the effect of varying concentration of electro-chemicals at synapse in a single neuron model. To that, they developed a simulation-based study with the temporal encoding technique to analyze the encoding mechanism.

Finally, Kothandan and Sujatha published a work entitled “Deep Neural Network with Stacked Denoise Auto Encoder for Phishing Detection”. In this paper, authors present and discuss a deep neural network to detect phishing uniform resource locators. They use a feature vector with a stacked denoise auto encoder. In addition, the noisy data is trained to reconstruct a clean input feature vector. Experiments are based on the Ham, Phishing Corpus and Phishload datasets to prove its effectiveness.

REFERENCES

[1] Puri, Vikram, Puri, Bhuvan, Jagdev, Sandeep Singh and Minh, Trí Tran Huu (2019). Cloudbin: Internet of Things based Waste Monitoring System. *International Journal of Machine Learning and Networked Collaborative Engineering*, 3(02) pp 76-82.

https://doi.org/10.30991/IJMLNCE.2019v03i02.001

[2] Rimal, Yagyanath , (2019). Machine Learning Prediction of Wikipedia Time Series Data using: R Programming. *International Journal of Machine Learning and Networked Collaborative Engineering*, 3(02) pp 83-92.

https://doi.org/10.30991/IJMLNCE.2019v03i02.002

[3] Sen,Santanu Kumar, Dey, Sharmistha and Bag, Rajib (2019). Study of Energy Efficient Algorithms for Cloud Computing based on Virtual Machine Migration Techniques. *International Journal of Machine Learning and Networked Collaborative Engineering*, 3(02) pp 93-101.

https://doi.org/10.30991/IJMLNCE.2019v03i02.003

[4] Baghela1, Vishwadeepak Singh, Bharti, Sunil Kumar and Choudhary , Saket Kumar (2019). Information Processing in GLIF Neuron Model with Noisy Conductance. *International Journal of Machine Learning and Networked Collaborative Engineering*, 3(02) pp 102-113.

https://doi.org/10.30991/IJMLNCE.2019v03i02.004

[5] Kothandan Sumathi and V. Sujatha (2019). Deep Neural Network with Stacked Denoise Auto Encoder for Phishing Detection. *International Journal of Machine Learning and Networked Collaborative Engineering*, 3(02) pp 114-124.

https://doi.org/10.30991/IJMLNCE.2019v03i02.005

Editor-in-Chief's

International Journal of Machine Learning and Networked Collaborative

Vijender Kumar Solanki, CMR Institute of Technology, Hdyerabad, TS, Insia

https://orcid.org/0000-0001-5784-1052

Vicente García-Díaz, University of Oviedo, Spain

https://orcid.org/0000-0003-2037-8548

DOI : https://doi.org/10.30991/IJMLNCE.2019v03i02

EDITORIAL BOARD MEMBER

HONORARY EDITOR

EDITOR-IN-CHIEF

ASSOCIATE EDITOR

MANAGING EDITOR

EDITORIAL BOARD MEMBER

Jordán Pascual Espada

- PhD
- Associate Professor
- Department of Computing,
- University of Oviedo, Spain.
- pascualjordan@uniovi.es

Marcin Paprzycki

- Ph.D, D.Sc
- Professor,
- Systems Research Institutes,
- Polish Academy of Sciences, Warsaw, Poland
- marcin.paprzycki@ibspan.waw.pl

Mihir Narayan Mohanty

- Ph.D, Professor,
- Department of ECE
- Siksha 'O' Anusandhan, (Deemed to be University), Odisha, India
- mihir.n.mohanty@gmail.com

Nguyen Cuong

- Ph.D, Lecturer,
- Department of Information Technology
- Quang Nam University, Vietnam
- nguyenhahuycuong@gmail.com

Rashmi Agarwal

- Ph.D, Professor, Department of Computer Applications
- Manav Rachna International Institute of Research And Studies, Haryana, India
- rashmi.fca@mriu.edu.in

Ruben Gonzalez Crepo

- Ph.D
- Director & Professor,
- Universidad Internacional de La Rioja (UNIR), Spain
- ruben.gonzalez@unir.net

Son Doan Trung

- Lecturer, Network & Security Department,
- University of People's Security, Hanoi, Vietnam
- son.doantrung@gmail.com

Tran Duc Tan

- Ph.D, Assciate Professor,
- Faculty of Electronics & Communications, VNU
- University of Engg. & Tech., Vietnam National University, Hanoi, tantd@vnu.edu.vn

Vishwanath Bijalwan

- Ph.D, Department of Electronics & Communication
- Institute of Technology, Gopeshwar, Uttarakhand, India
- vishwanath.bijalwan@itgopeshwar.ac.in

INDEX

EDITORIAL OFFICE

Vicente Garcia Diaz, Ph.D.
Associate Professor
Department of Computer Science
University of Oviedo, Spain
garciavicente@uniovi.es

Vijender Kumar Solanki, Ph.D
Associate Professor
Department of Computer Science & Engineering
CMR Institute of Technology (Autonomous), Hyderabad, TS, India
vijendersolanki@cmritonline.ac.in,spesinfo@yahoo.com

CALL FOR PAPER

International Journal of Machine Learning and Networked Collaborative Engineering (IJMLNCE) with ISSN **2581-3242**, is a quarterly published, open access, peer-reviewed, international journal, focuses on publishing authentic and unpublished quality research papers in the emerging field of Machine Learning and Collaborative Engineering.

This is a scientific journal aims to promote highly qualitative and innovative research works, not limited to but focuses on the area of machine learning, collaborative engineering, and allied areas. We publish only original research articles, review articles, and technical notes.

The journal is registered with CrossRef **10.30991/IJMLNCE**. Each accepted manuscript shall be assigned a unique digital object identifier for identification. The abbreviated key title for our journal allocated by ISSN is **Int. j. mach. learn. Networkedcollab.eng**.

The journal appears in popular indexes like BASE (Bielefeld Academic Search Engine), CNKI Scholar, CrossRef, CiteFactor, Dimensions, DRJI, Google Scholar, Index Copernicus, J-Gate, PKP-Index, Portico , ROAD, Scilit, Semantic Scholar, Socolar, WorldCat. Our journals manuscripts can also be found in the directory of ReserachGate & Kudos.

In this present technological era, the areas like machine intelligence, machine learning, and its associated domains are one of the most popular and demanding choices for the researchers as well as the industry personnel.

In last few years, numerous uses of machine learning and its related domain, has drawn ample attention of the people, that has generated a large number of applications in this field, making machine learning and collaborative engineering highly admired one.

Machine intelligence or machine learning is not a new concept. In terms of Artificial Intelligence, we were familiar with several aspects of the field, but nowadays with the introduction of machine learning, the use of this has been highly evolving, especially for improving the lifestyle of the human being.

There are numerous application areas of machine learning or machine intelligence, irrespective of any famous sector, from Healthcare, Space, Automation, Aviation industries etc. to entertainment industry and even in academia.

With this insight, IJMLNCE invites quality and authentic research works in form of manuscripts, for our forthcoming version. The Journal solicits latest research work carried out by the researchers, academicians and industry practitioners, in the domain of machine learning and collaborative engineering.

Plagiarism is considered as a serious ethical and professional malpractice in case of IJMLNCE. So similarity check should be taken care at its highest level. The originality of the work will be highly appreciated. Each paper will be peer-reviewed by the expert reviewers and in a multilayer approach. The editorial board of the journal believes in high moral and ethical practices.

The authors are invited to submit their authentic and unpublished manuscripts online only. They will get final decision about the acceptance or rejection of the manuscript within 06-12 weeks from the date of submission of manuscripts. Once the manuscript gets accepted, author will be asked to complete the copyright agreement. An e-mail communication will also be sent to the author’s registered mail ID, after verification of incorporation of all reviewers’ comments, in term of their final camera-ready-paper; it will be published to the upcoming journal’s volumes.

Please note that there is **no article processing cost** for publishing articles in this journal and the journal is open access.

IJMLNCE JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2019v03i02.001

Cloudbin: Internet of Things based Waste Monitoring System

[a]Vikram Puri[*],[b]Bhuvan Puri,[c]Sandeep Singh Jagdev, [d]Trí Tran Huu Minh

[a,d]*Duy Tan University, Da Nang, Vietnam*

[b]*D.A.V institute of Engineering and Technology, Punjab, India*

[c]*Ellen Technology(P) Ltd, Punjab, India*

[a]*purivikram@duytan.edu.vn(https://orcid.org/0000-0002-4023-9935)*

[b] *bhuvanpur239@gmail.com(https://orcid.org/0000-0002-3098-7892)*

[c] *sandeepsinghjagdev6@gmail.com(https://orcid.org/0000-0002-7834-823X)*

[d] *tritranhuuminh@gmail.com(https://orcid.org/0000-0002-8038-5048)*

Abstract

Nowadays, waste management has become a critical issue for the environment. Government and private agencies need to take certain action for proper management and cleanliness. The absence of systematic waste management system creates many issues for the environment and living creatures. Research on the Internet of Things (IoT) applications widely increased in many sectors. The waste management system is also one of the sectors. Therefore, in this study, IoT based waste monitoring system called Cloudbin is proposed to reduce the waste garbage from urban areas. In this system, Ultrasonic sensor is fixed on the top of the waste bin to monitor the level of garbage inside the bin and connected to the Blynk server. In addition, a GPS module is also employed to check the location of Waste Bin. Methane detection from garbage is an important feature in the system. Results show that the proposed system is suitable to monitor and control waste in cities.

Keywords

Internet of Things

Cloud Computing

IoT

Garbage Monitoring System

Artificial Intelligence

Blynk

1. Introduction

Since the last few decades, there is a ton of buildings and industries which have been constructed in urban areas [9]. The main cause behind this construction is to migrate a lot of people from rural areas to the urban cities for finding a job. In order to fulfill the need for shelter, the government constructed buildings for their accommodation. Many private and government sector based industries extended their branches in the urban areas to suffice the employment needs [10]. One of the major

[*]Corresponding author

Vikram Puri

Email: purivikram@duytan.edu.vn

problems originated from this development is “Waste” which is depleting the environment quality day by day.

Waste can be divided into three categories: 1) Solid waste 2) liquid waste and 3) gas waste, all can be hazardous. Solid and liquid can be recycled, reused or some of it can be converted into organic matter. The main source of liquid waste is from dirty water from homes or hazardous waste from industrial processes. Whereas solid waste comes from homes or industrial garbage or solid waste left after industrial processes. The government takes appropriate actions to reduce this garbage, recycle the solid waste which helps to keep the environment clean.

In the Internet of Things(IoT) concept, things or objects are connected around a network [1]. Wireless technologies such as Bluetooth, Wi-Fi, Xbee and RFID make a communication bridge to overcome many challenges for the successful implementation of IoT system. Kevin Astron introduced the term ‘IoT’ in 1999 at MIT Lab [2]. In the initial findings, RFID was used for communicating, tracking and storing the data. However, RFID has a lot of barriers and limited use to fulfill many security challenges namely eave-dropping, jamming, replay attacks. Nowadays, IoT (see figure 1) performs data fetching, data gathering or storing and processing with artificial intelligence techniques to make the device smarter [3]. IoT extended its limited in various applications namely atmospheric monitoring [4], tracking system [5], traffic management system [6], healthcare industry [7] and smart buildings [8] to improve the quality of life.

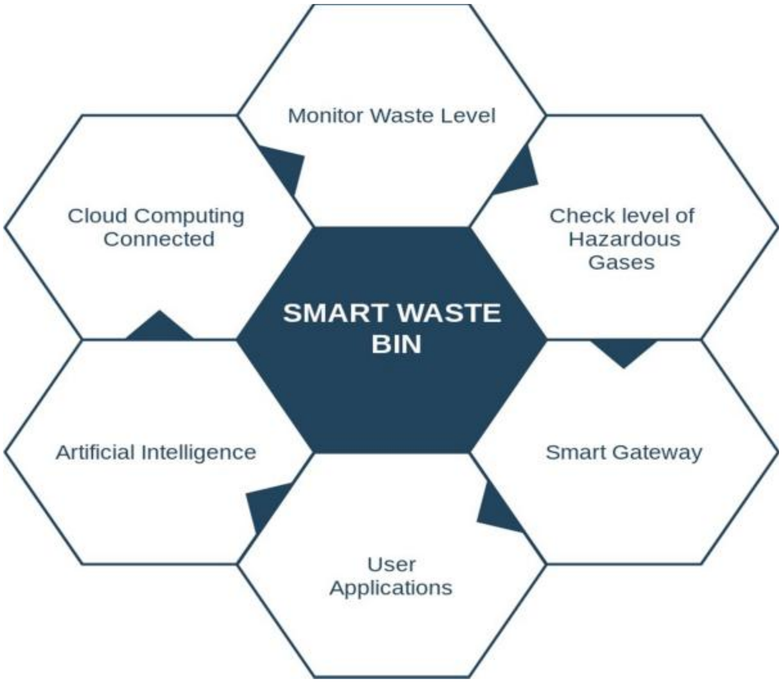


Fig. 1: Smart Waste Management System

IoT plays a vital role in controlling and monitoring the waste management system. Mustafa [11] designed and deployed a garbage monitoring based on the IoT. For monitoring purpose, Ultrasonic sensor and ARM microcontroller are used for the level detection and processing respectively. After monitoring and processing, the data is directly sent to the Thingspeak cloud server. Navghane [12] proposed a waste monitoring system based on the IR sensor. The sensor is used for detection of levels of Garbage and send to the cloud server for data storage and processing. Kumar [13] developed an IoT based monitoring system for the waste management system. GSM module is used to create connectivity between the sensors and cloud servers. User terminal namely Android application is used to display the processed data. The system is deployed in different locations and connected to one main server. Joshi [14] proposed a solution for waste management system called SMARTBIN which integrates wireless sensor network with cloud computing and machine learning techniques such as decision forest regression to improve the efficiency of garbage monitoring. Bharadwaj [15] proposed an IoT based smart monitoring system to monitor and manage solid waste. Data processing and Data sending is through the ATmega328 and LoRa technology. MQTT protocol is used to share data between the electronic circuit and cloud server. Begur [16] discussed a mobile-based real-time innovative solution for illegal dumping, monitoring, and management of waste.

In our proposed work, we propose an IoT based Waste management system to monitor waste bin level and also monitor the methane gas generated from the solid waste. In addition, our proposed system also sends GPS location to the User terminal for checking the exact location of the waste bin. The proposed paper is discussed as follows: Section 2 explains Methodology, Section 3 discusses results and discussion, and Section 4 discusses the conclusion.

2. Methodology

2.1. System Architecture

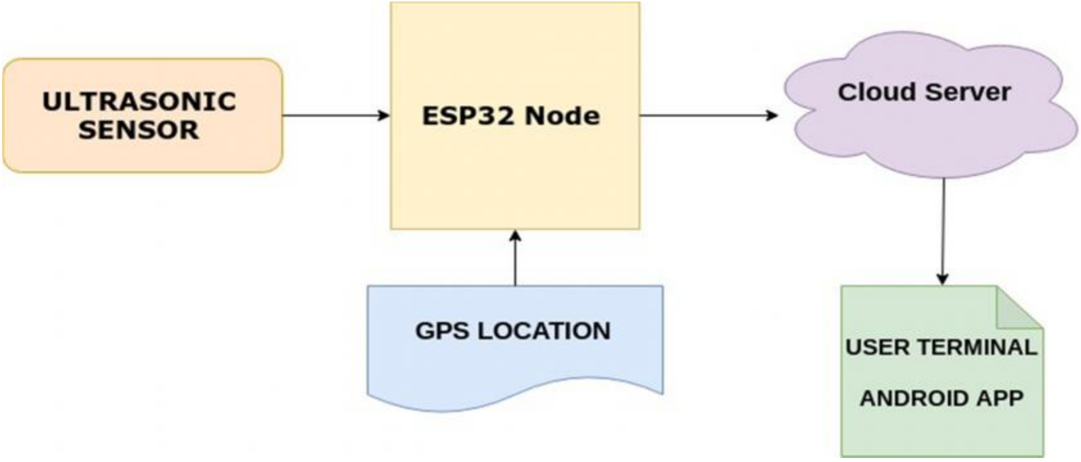


Fig. 2: System Architecture

Our proposed system is categorized into three different layers: 1) Sensing Layer 2) Processing Layer 3) Cloud Layer. Sensing layer consists of the sensor namely ultrasonic sensor which lends a hand to monitor the changes in the ultrasonic waves. These waves' frequency is too high for the human hears. The main working principle behind the sensor is to calculate the distance of the reflected wave. The formula for distance calculation of reflected wave is:

$$\textbf{Distance} = \tfrac{1}{2}\ \textbf{Time} * \textbf{Speed of sound} \qquad \textbf{(1)}$$

Speed of sound varies with the humidity and temperature.

The second layer is named as the processing layer which helps to fetch data from the sensor and applied algorithm for processing data. In our proposed system, NodeMCU is used for processing the data and is equipped with a Wi-Fi module. Moreover, GPS modules also interfaced with NodeMCU for waste bin location. Cloud layer in our proposed study is worked as the third layer. Blynk server employed to process different data and visualize with graphical effects on the user terminal.

2.2. Circuit Diagram

In the Proposed study, NodeMCU plays the major role. NodeMCU is an open source IoT platform integrated with the Wi-Fi module. It has 10 General Purpose Input/Output (GPIO) pins for connecting different modules or sensor. In our study, Ultrasonic sensor (HC-SR04) is connected to the NodeMCU with two different pins as follows: 1) one pin is used for input as ECHO 2) another pin used for output as TRIGGER. The ultrasonic sensor emits waves at a frequency of 40,000 Hz. If there is an obstacle or object between the waves, it reflects back to the sensor for measuring. The Object detection range varies from 2cm to 400cm. GPS module is also connected to the NodeMCU. NEO-6M GPS module is based on the serial communication pins which have in-built EEPROM and external antenna for better sensitivity. 9600 bps is by default baud rate of this module and operates between 3v to 5v. Table 1 represents the technical specification of NodeMCU.

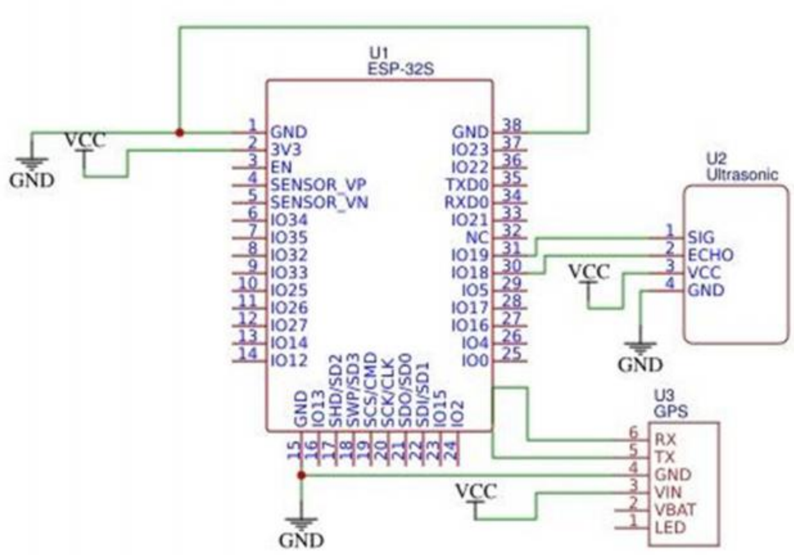


Fig. 3: Circuit Diagram

Table 1: Represents the technical specification of NodeMCU.

S.No	**Parameter**	**Values**
1	Firmware	Lua Scripting
2	Software	Arduino
3	Interface	USB_TTL
4	GPIO	10 Pins
5	In-built protocols	ADC, 1-wire, SPI, I2C
6	Power Supply	5-Volt
7	Antenna	In-Built
8	Network Interfacing	API

2.3. Cloud Server

Blynk is designed for the IoT application and also have the ability to control application remotely. It can store, process, visualize data, and graphical user interface for the Users. Blynk is categorized into three parts as follows:

1. **Application:** Android and iOS user-friendly application for smartphones and tablet having visual data with amazing graphics.
2. **Server:** It is responsible for all communication between the sensors and application. It is based on the Blynk cloud or local server. It can handle hundreds and thousands of devices at one time.
3. **Libraries:** It is already compatible with all types of hardware platforms and enabled communication with Blynk cloud

3. Result and Discussion

To evaluate the performance of our proposed study, location detection, level detection and methane in the garbage is employed in this study. Figure 4 represents the data shown on the Blynk server application

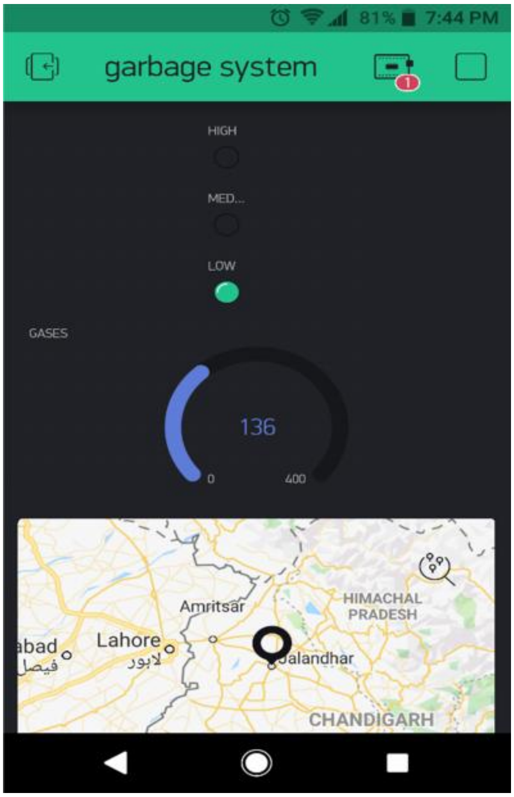


Fig. 4: Result of Proposed Work

For the level detection, three different levels are processed as low level, medium, and full level. When the level of the waste bin is full level, it sends information to the municipal authority to clear the waste bin. Methane detection is also an important parameter to check how much hazardous waste material is present inside the waste bin. GPS location is also stamped when it sent to the Cloud server (Blynk).

4. Conclusion

In this study, we have designed and developed an IoT based waste monitoring system to replace the existing waste monitoring systems. Due to the increase and migration of people from rural to urban areas, waste production has increased in the form of solid, liquid and gas. The main purpose behind the system is to maintain the cleanliness of urban areas. The basic architecture of our system is a centralized structure in which every waste bin is connected to the Blynk server for monitoring, tracking and processing the data value.

Our proposed system works on three different layers. One is to check the level of garbage inside the waste bin and the checking is based on the three different levels. Second is to check the hazardous gas, methane, inside the garbage through the use of a gas sensor. Last, we stamp GPS coordinates with the data to check the location of the bin. Nevertheless, the proposed system's performance is better as compared to the existing waste bin with respect to data transferring and accuracy. The system also decreases the usage of manual workers. Moreover, methane, being a hazardous gas for the workers, can be detected inside the bin before opening. In the future, we look forward to implementing Artificial Intelligence techniques for the bin to make it smarter.

References

[1]. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). “Internet of Things (IoT): A vision, architectural elements, and future directions”. *Future Generation Computer Systems*, 29(7), 1645-1660.

[2]. Ashton, K. (2009). That ‘internet of things’ thing. *RFID journal*, 22(7), 97-114.

[3]. Hong, I., Park, S., Lee, B., Lee, J., Jeong, D., & Park, S. (2014). “IoT-based smart garbage system for efficient food waste management”. *The Scientific World Journal*, 2014.

[4]. Lazarescu, M. T. (2013). “Design of a WSN platform for long-term environmental monitoring for IoT applications”. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(1), 45-54.

[5]. Gama, K., Touseau, L., & Donsez, D. (2012). “Combining heterogeneous service technologies for building an Internet of Things middleware”. *Computer Communications*, 35(4), 405-417.

[6]. Foschini, L., Taleb, T., Corradi, A., & Bottazzi, D. (2011). “M2M-based metropolitan platform for IMS-enabled road traffic management in IoT”. *IEEE Communications Magazine*, 49(11), 50-57.

[7]. A. J. Jara, M. A. Zamora, and A. F. G. Skarmeta, “An internet of things-based personal device for diabetes therapy management in ambient assisted living (AAL)”. *Personal and Ubiquitous Computing*, vol. 15, no. 4, pp. 431–440, 2011.

[8]. Zhang, D., Shah, N., & Papageorgiou, L. G. (2013). “Efficient energy consumption and operation management in a smart building with microgrid”. *Energy Conversion and Management*, 74, 209-222.

[9]. Yusof, N. M., Jidin, A. Z., & Rahim, M. I. (2017). “Smart garbage monitoring system for waste management”. In MATEC Web of Conferences (Vol. 97, p. 01098). *EDP Sciences*.

[10]. Tikone, N., Zagade, P., Singh, G., & Cherian, M. (2018). “Smart Garbage Management”. Available at *SSRN* 3361515.

[11]. Mustafa, M. R., & Azir, K. K. (2017). “Smart Bin: Internet-of-Things Garbage Monitoring System”. In MATEC Web of Conferences (Vol. 140, p. 01030). *EDP Sciences*.

[12]. Navghane, S. S., Killedar, M. S., & Rohokale, V. M. (2016). “IoT based smart garbage and waste collection bin”. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, 5(5), 1576-1578.

[13]. Kumar, S. V., Kumaran, T. S., Kumar, A. K., & Mathapati, M. (2017, August). “Smart Garbage Monitoring and Clearance System using Internet of Things”. In 2017 *IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)* (pp. 184-189). IEEE.

[14]. Joshi, J., Reddy, J., Reddy, P., Agarwal, A., Agarwal, R., Bagga, A., & Bhargava, A. (2016, August). “Cloud computing based smart garbage monitoring system”. In 2016 *3rd International Conference on Electronic Design (ICED)* (pp. 70-75). IEEE.

[15]. Bharadwaj, A. S., Rego, R., & Chowdhury, A. (2016, December). “IoT based solid waste management system: A conceptual approach with an architectural solution as a smart city application”. In 2016 *IEEE Annual India Conference (INDICON)* (pp. 1-6). IEEE.

[16]. Begur, H., Dhawade, M., Gaur, N., Dureja, P., Gao, J., Mahmoud, M., & Ding, X. (2017, August). “An edge-based smart mobile service system for illegal dumping detection and monitoring in san jose”. In 2017 *IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation* (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI) (pp. 1-6). IEEE.

Author’s Biography

Vikram Puri is a Researcher at the Center of Simulation and Visualization, DuyTan University, Da Nang, Vietnam and also Visiting Research Scholar at University of Nevada, LasVegas, USA. His research of interest is eHealth Technologies, Internet connected medical equipment, wearable health technologies. He has Bachelor of Technology in Electronics and Communication at Punjab Technical University, Punjab, India. He is currently pursuing his master and PhD in Computer Science at DuyTan University, Vietnam. He has total 3 years of Industrial experience. He has delivered many workshops and seminars regarding the new research development in collaboration with Intel. He was the Embedded Developer in Enjoin Technology (P) Ltd., 2014–2016 and Senior Embedded Developer in Ellen Infotech Pvt. Ltd., 2016–2017. He is also working as consultant for the corporate. He has written research papers on SCI Q1, Q2 and Scopus indexed Journals. He is also acts as reviewers for the SCI journals and conferences.

Bhuvan Puri is a student of D.A.V institute of Engineering and Technology affiliated from the Punjab Technical University, Punjab, India. He is pursuing B.Tech in Mechanical Engineering. Currently he is working in field of IoT and its application as well as AutoCAD designing. He has experience of working under learned and experienced mentors.

Sandeep Singh Jagdev is a Senior Embedded Engineer in Ellen Technology (P) LTD., Punjab, India. He has Bachelor of Technology in Electronics and Communication at Punjab Technical University, Punjab, India. He has total 2 years of Industrial experience. He has delivered many workshops and seminars regarding the new research development. He has written research papers on Scopus indexed Journals. His research area is IoT and its applications, digital circuit designing and computer vision.

TríTran Huu Minhis an IoT Engineer in Duy Tan University, Da Nang, Vietnam. He has Master of Science in Control Engineering from Bach Khoa University, Vietnam. He has total 1 years of experience in Duy Tan University. He has developed products using Internet of Things. His research area is automation, IoT.

How to Cite

Puri, Vikram, Puri, Bhuvan, Jagdev, Sandeep Singh and Minh, Trí Tran Huu (2019). Cloudbin: Internet of Things based Waste Monitoring System. *International Journal of Machine Learning and Networked Collaborative Engineering*, 3(02) pp 76-82.

doi : https://doi.org/10.30991/IJMLNCE.2019v03i02.001.

IJMLNCE JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2019v03i02.002

Machine Learning Prediction of Wikipedia Time Series Data using: R Programming

Yagyanath Rimal

Faculty of Science and Technology

Pokhara University, Nepal

Email: rimal.yagya@gmail.com

https://orcid.org/0000-0003-1045-7728

Abstract

This review article explains the prediction of automatic learning of Wikipedia time series data using r programming. Although many time series forecast researchers have been analyzed the time series could not cover the gap between chart interpretation and time series analysis of the Internet database directly. Its main objective is to explain the simplest way to time model series whose data structure was different using R programming, the result was sufficiently summarized with different forecast models. The simplest form of analysis with graphical interpretation to obtain conclusions from the time search is Cristiano_Ronaldo of Wikipedia, a best player in euro football team. Whose trend and prediction is analyzed for next 2020 from the past records trend. Therefore, this document presents the simplest way to predict time series data and its strengths for data analysis using R programming.

Keywords

Data Analytics Machine Learning, Autocorrelation Function, Particle Automatic Correlation Function

1. Introduction

A historical series is a set of observations at different measurement periods of an employee who collected at regular time intervals, such as monthly, weekly or yearly, such as the state budget, etc. According to (Gahirwal, 2018), the precondition is the one whose interval must be the same. Time series forecasts are widely used in econometrics, mathematical and financial forecasts of statesmen and different weather forecasts and earthquakes. Magazine without independent variable (VANNESCHI, 2017). With the time-based model, the researcher can interpolate the graphics of the model and then foresee a future project. The temporal variable usually uses yt = yt-1 + E as a univariate model. The data in cross section are such data collection procedures that are similar to the time series, but the time series data only have one variable depending on the interval and the association of a single value, but in the cross-data collection can collect many variables Elements in a fixed time interval. The data model may vary based on the time series model. It is not expected, although there are many types such as seasonal, trend, cyclic and random. Models are increasing or decreasing models. The period of occurrence has been corrected within a year or less. The cyclical model is a good example of the government budget. Some data are also random. Purely random, whose average is zero and the variance is constant (CHEN, 2013). So the dispersion diagram will not indicate the correct model. The automatic regressive model is a model that corresponds to the sequences yt-1, yt-2 and t-3. Therefore, the model becomes yt = b0 + b1yt-1 + b2yt-2 + b3y3-3 ... ..Et. The AR (0) means

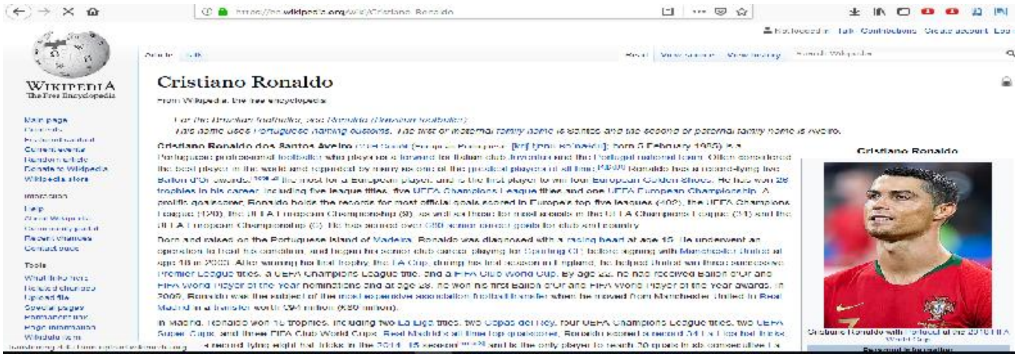
B0, so the AR model is better when it works. The moving average model always speaks of the error terms in each regression and t = B0 + Et + Q1E1 + Q2Et-2 + QET-q. Therefore, ARMA is a precise model that uses AR and MA models in temporal data (Michael Jachan, 2007). regressive automatic movement media regressive integrated mobile media ARIMA is commonly known as the Box Jenkins methodology (1976), a method used to predict the basis of information from its own variables, based on an analysis of trends univariate tendencies. After analyzing the periodic properties of the variables, machine learning. Regulators, policy makers and companies to make serious and economic forecasts; however, the choice is based on the two hypotheses (SALAM, 2013). The AR model is applied in series as stationary if there is an invariable time variation. This decreases the major MA duration model applied to the autoregressive process convergent final order that uses the autocorrelation function (ACF) is the covariance of the following terms and functions automatic creation partial yt (PACF) and yt p. The series is not firm, can be fixed after differentiation.

Machine learning is the system that takes input and output as input parameters, so that the computer system automatically produces the above parameters based on some models. This can be applied to the analysis of neural networks, deep learning and artificial intelligence networks. traditional programming always requires two inputs and programs and possibly produces an output, while the process of machine learning always requires an input and real requirement, since the formal entry system produces the show and expected output depending on the needs of the system (Sunday 2018). A fixed series after an integrated differentiation in the order 1. Analysis Box - Jenkins concerns a systematic method of identification, regulation, monitoring and the use of integrated models of the integrated moving time series (ARIMA initial). The method is appropriate for medium to long time series. (Buncher 2018) The best correspondence between the ARIMA model and the time series data is the best correspondence of ARIMA (0,0,0) means that p = q = i = 0.

The data science process includes two processes; The first process begins with cleaning the raw data and data collection analysis using different algorithms to produce the display data in this process. At each stage, IT skills, mathematics, statistics and interpretation are required. However, the data available in today's world in any structured, unstructured and semi-structured format are raw data. The primary phase includes the integration of raw data and, therefore, the selection of the required data, at some point requires a processing required before the analysis. data cleaning requires 50% to 80% of the work of a set of scientific data (Ruiz, 2017). Time series analysis includes methods for analyzing time series data to extract significant statistics and other data characteristics. The prediction of time series is the use to predict future values based on the values observed above. Historical headline series and retail sales in this publication are widely used for non-stationary data such as the economic climate. We will demonstrate different approaches to predict the time series to detail. There are two types of machine learning: supervised learning and unsupervised learning. However, R programming, python and weka are the best tools for data analysis; The data scientist can use many other data analysis processes. R has extensive facilities for analyzing time series data. This section describes the creation of a historical series, seasonal decomposition, exponential models and modeling and prediction with the ARIMA package the prognosis (Change, 2018). Modeling time series, as the name suggests, means working with (time days, hours, minutes) time-based data for hidden information and making informed decisions. Time series models are very useful models when data is related in series. Most business houses that work with time series data to analyze the number of sales next year, website traffic, place of competition and much more. However, it is also one of the areas that many analysts do not understand. There are three basic criteria for a series to be classified as stationary series. The series average should not be a function of time, but must be a constant. The variance of the series should not be a function of time. This property is called homoscedasticity with a variable data distribution. The term covariance i-th term and (i + m) th should not be a function of time, therefore, covariance is not constant over time for uniform (Chohlan 2018). The reason I took this first section is that, unless the time series is stopped, it is not possible to create a time series model. In cases where the fixed criterion is violated, the first requirement becomes stationary time series and therefore to try out stochastic models to foresee this historical series. There are many ways to bring this stationarity. This is the most basic concept of time series. (Srivastavo, 2015).

2. Using R Programing

Here I use data set of the database Cristiano_Ronaldo of Wikipedia, is the top scorer in international football careers. He has won 26 trophies in his career, including five league titles, five UEFA Champions League titles and the UEFA European Championship. Ronaldo, a highest scorer, holds the record for the Master challenge.



WIKIPEDIA
The Free Encyclopedia

Cristiano Ronaldo

From Wikipedia, the free encyclopedia

Cristiano Ronaldo

Fig. 1: Cristiano Ronaldo(Source: Wikipedia, 2018)

```
> install.packages("wikipediatrend")
Error in install.packages : Updating loaded packages
> library(wikipediatrend)
> devtools::install_github("petermeissner/wikipediatrend")
> library(wikipediatrend)
> wc=wp_trend( page="Cristiano_Ronaldo",from="2015-07-01",to="2018-10-31")
The package wp_trend() function that allows us to get page view of data time series data.
> wc
    granularity date      views
1124 daily     2018-0 ..36744        715  daily    2017-0 ..30132
612  daily     2017-0 ..19273        232  daily    2016-0 ..60506
927  daily     2018-0 ..23233        379  daily    2016-0 ..74485
1003 daily     2018-0 ..22562        454  daily    2016-0 ..18957
789  daily     2017-0 ..31208        994  daily    2018-0 ..38602   ... 1209 rows of data not shown
> library(ggplot2)          > View(wc)
> qplot(date,views,data=wc)
```

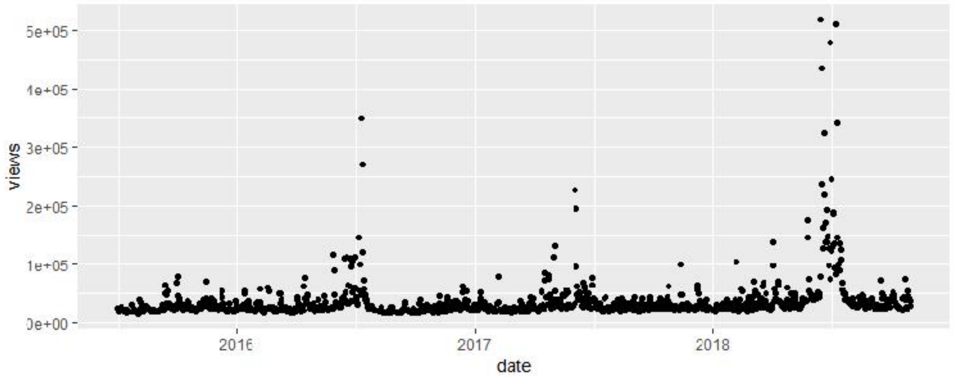


Fig. 2 :Daily Views Search for Cristiano Ronaldo keyword

From the above figure there were more seasonality in data sets but good in rising pattern in some duration.

```
> summary(wc)
project   language     article
Length:1219 Length:1219 Length:1219 Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character
access  agent      granularity
Length:1219 Length:1219 Length:1219
Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  : character        date                  views
Min.:2015-07-0100:00:Min.: 15478  1st Qu.:2016-04-30:01stQu.:23040
Median :2017-03-01: Median:27818
Mean:2017-03-01 00: Mean:37300
3rd Qu.:2017-12-30:3rd Qu.:37117
Max.:2018-10-31 00:Max.:519908
> wc$views[wc$views==0]=NA
> ds=wc$date
> y=log(wc$views)
> df=data.frame(ds,y)
> qplot(ds,y )
```

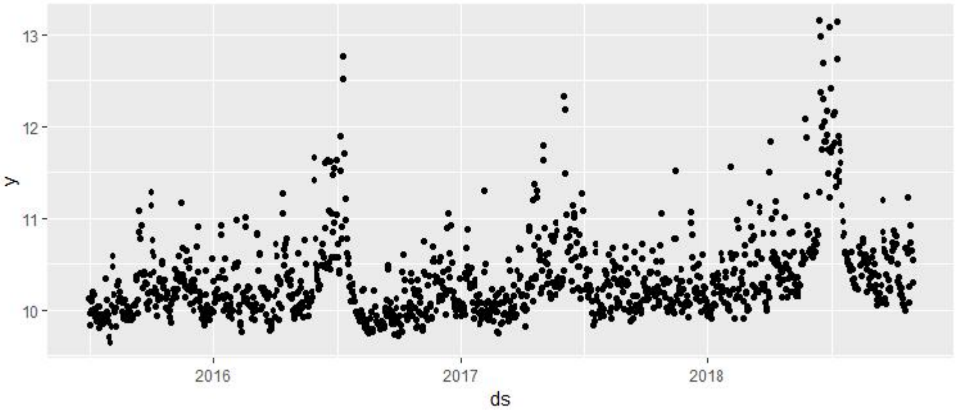


Fig. 3: Daily Log of goal Score search for the Keyword

This figure plots the log goal score and data variable in x axis shows pattern.

```
> installed.packages("prophet")
> library(prophet)
```

Here we are using forecast function columns ds (date type) and y, the time series. If growth is logistic, then df must also have a column cap that specifies the capacity at each ds. If not provided, then the model object will be instantiated but not fit; use fit.prophet(m, df) to fit the model.

```
> m=prophet(df)
> m
$`growth`
[1] "linear"
$changepoints
 [1] "2015-08-09 GMT" "2015-09-17 GMT" "2015-10-26 GMT"
 [4] "2015-12-04 GMT" "2016-01-12 GMT" "2016-02-20 GMT"
 ………………………………
[22] "2017-11-04 GMT" "2017-12-13 GMT" "2018-01-21 GMT"
[25] "2018-03-01 GMT"
$n.changepoints           [1] 25
$changepoint.range        [1] 0.8
$yearly.seasonality        [1] "auto"
$weekly.seasonality       [1] "auto"
……………………….
$uncertainty.samples     [1]1000
$specified.changepoints  [1] FALSE
$start       [1]"2015-07-01 GMT"
$y.scale     [1] 13.16141
$logistic.floor[1] FALSE
```

```
$t.scale       [1] 105235200
$changepoints.t
[1] 0.03201970 0.06403941 0.09605911 0.12807882 0.160
……………………..
[6] 0.19211823 0.22413793 0.25615764 0.28817734 0.320
[11] 0.35221675 0.38423645 0.41543514 0.44745484 0.47
[16] 0.51149425 0.54351396 0.57553366 0.60755337 0.63
[21] 0.67159278 0.70361248 0.73563218 0.76765189 0.79
$seasonalities
$seasonalities$`yearly`
$seasonalities$`yearly`[1]365.25
$seasonalities$`$fourier.[1] 10
$seasonalities$`yea$r.sca[1] 10
$seasonalities$`$mode[1]"addite"
$seasonalities$weekly$mode [1]
$extra_regressors   list()
$`stan.fit` NULL
$params
$params$`k` [1] 0.6589777
$params$m [1] 0.7410152
$params$delta
[,1]     [,2]       [,3]    [,4]
[1,] -6.962949e-05 -0.4095019 -0.3294997 -5.841874e-08
[,5]      [,6]      [,7]    [,8]
[1,] -4.191081e-08 -3.546061e-09 2.497178e-07 0.0244269
………………
[,22]   [,23]     [,24]   [,25]
[1,] -8.958495e-08 -3.950414e-08 -3.443782e-08 -0.0429
$params$sigma_obs [1] 0.02643767
$params$beta
[,1]   [,2]      [,3]       [,4]
[1,] 0.005891249 -0.01776793 -0.008432092 0.01224181
[,5]   [,6]      [,7]       [,8]
…………….
[,25]       [,26]
```

```
[1,] 0.0001815386 0.001226727
$history
ds      y floor      t  y_scaled
1  2015-07-01 10.129507   0 0.0000000000 0.7696371
2  2015-07-02  9.968854     0 0.0008210181 0.7574307
3  2015-07-03  9.834834     0 0.0016420361 0.7472479
………
198 2016-01-14 10.156656     0 01617405583 0.7716998
……………………………………………………………
199  2016-01-15  9.999298     0
0.1625615764 0.7597438
200  2016-01-16 10.128190     0 0.1633825944 0.7695370
[ reached getOption("max.print") -- omitted 1019 rows ]
$history.dates
[1] "2015-07-01 GMT" "2015-07-02 GMT" "2015-07-03 GMT"      [4] "2015-07-04 GMT" "2015-07-05 GMT" "2015-07-06 GMT"
[7] "2015-07-07 GMT" "2015-07-08 GMT" "2015-07-09 GMT"…………………………………………………………….
 [997] "2018-03-23 GMT" "2018-03-24 GMT" "2018-03-25 GMT"
[1000] "2018-03-26 GMT"
[ reached getOption("max.print") -- omitted 219 entries ]
[ reached getOption("max.print") -- omitted 219 entries ]
$train.component.cols
Additive weekly yearly multiplicati
1     1      0      1       0
2     1      0      1       0
………………………….
25    1      1      0       0
26    1      1      0       0
$component.modes
$component.modes$`additive`
[1] "yearly"            "weekly"
[3] "additive_terms"  "extra_regressors_additive"
[5] "holidays"
$component.modes$multiplicative
[1]"multiplicative_terms"
```

```
[2]"extra_regressorsmultiplicative"
attr(,"class")
[1] "prophet" "list"
> future=make_future_dataframe(m,periods=365) #For next year
> tail(future)
        ds
1579 2019-10-26          1580 2019-10-27
1581 2019-10-28          1582 2019-10-29
1583 2019-10-30          1584 2019-10-31
> forecast=predict(m,future)
> tail(forecast)
 ds trend additive additive_lowr
1579 2019-10-26 11.08094    -0.17044324 -0.17044324
1580 2019-10-27 11.08184    -0.08389419 -0.08389419
…………………..
additive_terms_weekly weeklylowr
1579-0.17044324  0.010662724  0.010662724
1580-0.08389419  0.093079248  0.093079248
……………………..
1584-0.18950107 -0.031694505 -0.031694505
weekly_upperyearly_lower upper
1579  0.010662724 -0.1811060   -0.1811060   -0.1811060
1580  0.093079248 -0.1769734   -0.1769734   -0.1769734
…………………….
1584 -0.031694505 -0.1578066   -0.1578066   -0.1578066
multiplicative_terms multiplicative
1579      0          0
………………
1584      0          0
multiplicative_terms yhat yhat
1579 0  10.11238 11.64221
………………..
1584 0  10.11592 11.63837
trend_lower trend_upper yhat
1579 10.50417  11.70324 10.91049
```

```
………………..
1584 10.49862  11.72232 10.89598
> tail(forecast[c('ds','yhat','yhat_lower','yhat_upper')])
ds    yhat yhat_lower yhat_upper
1579 2019-26 10.91 10.11 11.642
…………………
1584 2019-31 10.89 10.11 11.638
> exp(10.89598)
[1] 53959.01
> plot(m,forecast)
```

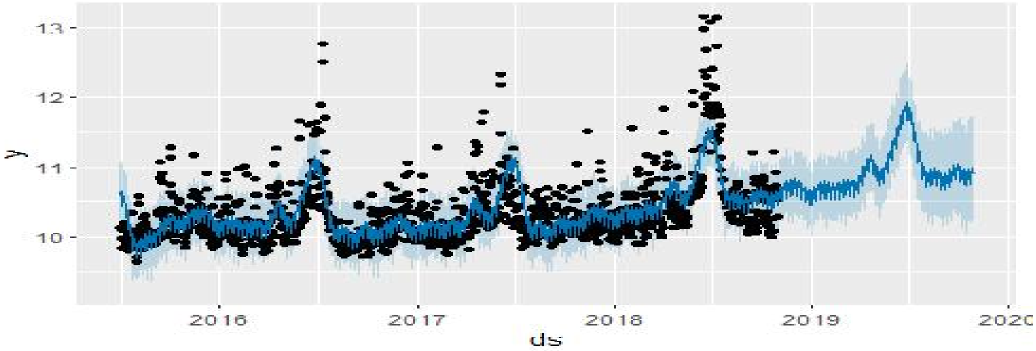


Fig. 4 : Daily Trend Forecasting for ‘Cristino Ronaldo’

From the above plot Cristino Ronaldo had prosperous in upcoming years.

```
> prophet_plot_components(m,forecast)
```

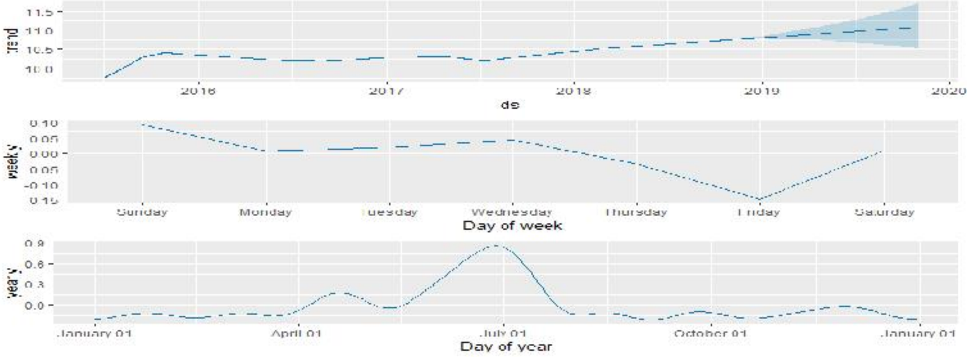


Fig. 5 : Daily Trend, Week Day Trend , and Date wise Trend for keyword ‘Cristino Ronaldo’

From the above figure *Cristiano Ronaldo* had good future in real Madrid champions ship however Friday matches does not suite him and July is the best suitable moment.

Conclusion

A common goal of time series analysis is the extrapolation of past behavior in the future. Forecasting procedures include random walks, moving averages, trend models, simple, linear, quadratic, and seasonal exponential time series models. Business forecasts can be based on historical data models used to predict future market behavior. The time series forecasting method is a data analysis tool that measures historical data points, using line charts to predict future conditions and events. It is essential to analyze trends before building any type of time series model. The details that interest us refer to any kind of trend, seasonality or random behavior in the series. Once we know that patterns, trends, cycles and seasonality, the function of the attacking forces and the defense of the competition could be analyzed, corrective measures will be taken, so Cristiano Ronaldo will have a good future in the next years after entering in the Real Champions League of Madrid has will good future coming day.

References

[1]. Buncher, D. (2018). “The Box-Jenkins methods”. *NCSS Statistical Software*.

[2]. Changing, S. L. (2018). “The world, one article at a time”. *Toronto Canada. Opinion=my own. ttps://www.linkedin.com/in/*
susanli/, Sr. Data Scientist,.

[3]. CHEN, M.-Y. (2013). “ Time Series Analysis (I)”. *Department of Finance*.

[4]. Chohlan, A. (2018). “A little book of R for time series” . *DataCamp's manipulating time series in R course by Jeffrey Ryan. .*

[5]. Domingos, P. (2018). “A Few Useful Things to Know about Machine Learning”. *Department of Computer Science and Engineering*.

[6]. Gahirwal, M. (2018). “Inter Time Series Sales Forecasting”. *Information Technology, Vivekanand Education Information Technology, Vivekanand Education*.

[7]. Michael Jachan. (2007). “Time-Frequency ARMA Models and Parameter Estimators for Underspread Nonstationary”. *IEEE Transactions on Signal Processing*, Vol. 55, No. 9, September 2007.

[8]. Ruiz, A. (2017). “The Cognitive Coder InfoWorld”.

[9]. SALAM, M. A. (2013). “Modeling and Forecasting Pakistan's Inflation by Using”. *Statistical Officer, Statistics Department, State Bank of Pakistan, Karachi, Pakistan.*

[10]. Srivastavo, T. (2015). “A Complete Tutorial on Time Series Modeling in R”.

[11]. Vanneschi, l. (2017). “Retail forecasting under the influence of promotional discounts”. *Instituto Superior de Estatística e Gestão de Informação.*

Author’s Biography

Yagyanath Rimal is lecturer of Computer Science Information Technology at Pokhara University, Nepal. He received his Master of Science in Information Technology from SMU in 2006, India with Information Technology and programming specialization, he had wide area of interest like teaching, writing books, doing research and publishing international research articles, web development and presentation at international conferences.

How to Cite

Rimal, Yagyanath , (2019). Machine Learning Prediction of Wikipedia Time Series Data using: R Programming. *International Journal of Machine Learning and Networked Collaborative Engineering*, 3(02) pp 83-92.

doi : https://doi.org/10.30991/IJMLNCE.2019v03i02.002.

IJMLNCE JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2019v03i02.003

Study of Energy Efficient Algorithms for Cloud Computing based on Virtual Machine Migration Techniques

[a]Santanu Kumar Sen, [b]Sharmistha Dey[*] , [c]Rajib Bag

[a]Gurunanak Institute of Technology, West Bengal,India

[b] Maulana Abul Kalam Azad University of Technology,West Bengal,India

[c]Professor Rajib Bag, Supreme Knowledge Education ,India

[a]papri.dey@gmail.com,(https://orcid.org/0000-0002-8019-0937)

Abstract

Green cloud is a catchphrase in today's IT industry. Due to the wide acceptibility of cloud,the industry is being inclined towards cloud application and the high demand of performance requirement in cloud leaves a question mark on energy consumption by cloud data centers. Hence energy efficiency in cloud computing is one of the most significant parameters to follow nowadays to evaluate the efficiency of the cloud service. It is a driving force for adaptability of a cloud computing service in recent era. For a highly commercial service like cloud, maintaining the QoS parameters and keeping the service availability and service quality highly optimized to get the competitive advantage, cloud data centers are almost available on a 24x7 basis ; which in turn is a reson for high power consumption. So it is very much necessary to maintain a balance between power and quality of the service. One feasible solution for achieving energy efficiency is Virtual Machine migration technique in real time or when they are in turned off condition. This paper discusses about several VM Migration techniques and analyses their perspectives.

Keywords

Cloud,

Data Center,

DVFS,

Load Balancing,

QoS,

VM Migration,

Virtualization

1. Introduction

Green cloud is an emerging practicing technology by todays developers and service providers, which reduces the power consumption as well as environment friendly[1].By adopting the new approach the cloud providers are not only being cautious towards the environment, but also they are saving their cost using energy efficient algorithm by reducing power consumption and also making a balance between energy saving and keeping quality of service as per the expectation of their client as well.

[*] Corresponding author

Sharmistha Dey

Email: papri.dey@gmail.com

At present large-scale data center comprises thousands of computer nodes. These data center devours huge amount of electric power and a big cause for CO2 emission to environment[2,4]. Cloud applications are installed in remote data centers, where large capacity servers and storage systems are being maintained. A fast growth of demand for cloud based services fallouts into establishment of massive data centers consuming very high amount of power. To protect this, energy efficient models are required, especially for complete infrastructure to decrease functional costs while preserving vital Quality of Service (QoS)[4]

1.1. Virtual Machine Migration Techniques: What, Why , When

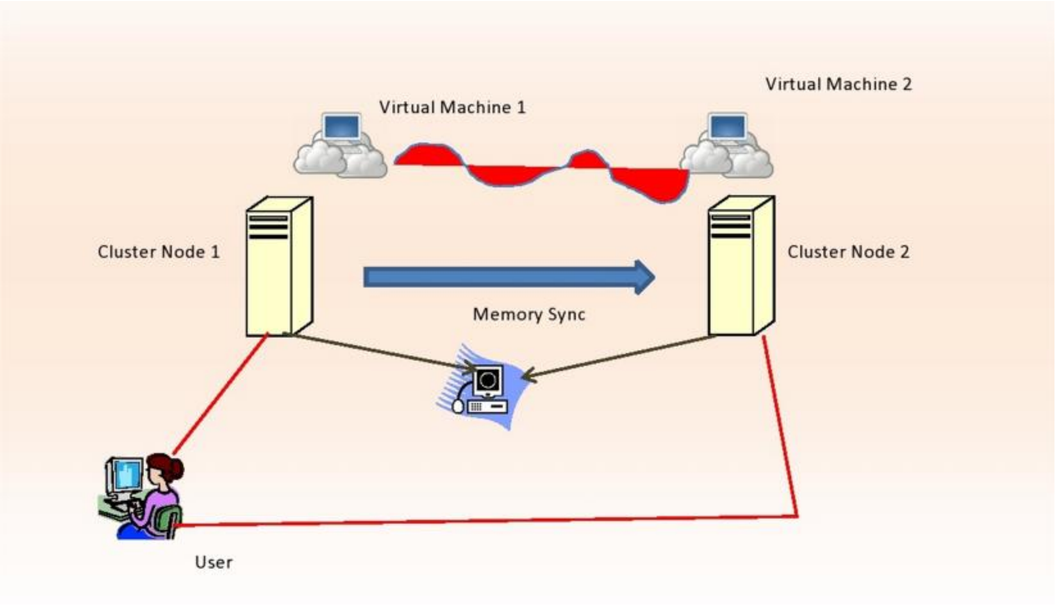


Fig. 1: Live VM Migration

1.2. Types of Virtual Machine Migration

Hot VM Migration - you can transfer an active or running virtual machine to a different host/ server , or you can interchange its disks or files to another data store without any interruption in the obtainability of the virtual machine. This is also known as "live migration"[12]. This technique also follows two different mechanisms- pre-copy memory migration and post copy memory migration.

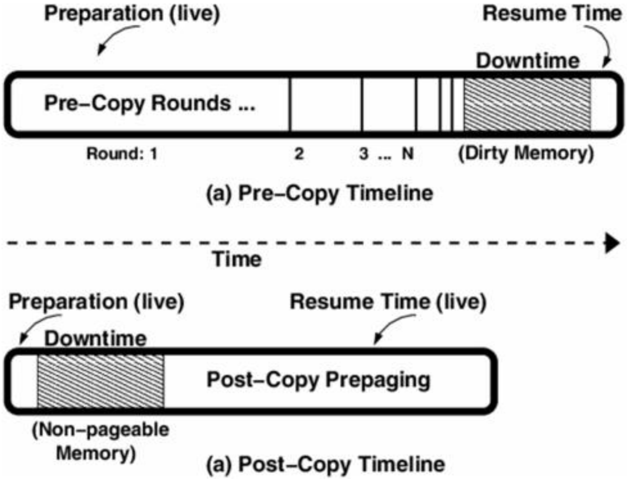


Fig. 2 :Pre- Copy Vs. Post Copy VM Live Migration(https://www.researchgate.net/figure/The-timeline-of-a-Pre-copy-vs-b-Post-copy-migration_fig1_221137806)

In case of pre-copy migration the Hypervisor copies the memory pages from source to the destination folder in the turned on condition , i.e. when the VM is in the still running . If some of the memory pages change during this process, they will be re-copied until they cover the bad copying rate[15].

Cold Migration – In this technique, a powered off condition or deferred virtual machine is being moved to a new host. Additionally, you can reposition the disk files for turned off or suspended virtual machines and move them to new storage locations. In case of cold migration process, virtual machines can be repositioned from one datacenter to another easily . To perform a cold migration, virtual machines can be moved manually or a scheduled task is set up.[12,16]

In VM data center you can have the following relocation options [12]

Table 1: Different types of relocation option for VM Migration

Change Host	Only relocate a virtual machine, but do not relocate it's storage to the destination host. You can move the virtual machine by using either cold or hot migration. One can also use vMotion technique to move a turned-on virtual machine to another host.
Change Data store	Moving a virtual machine along with it's storage, including the virtual disks and configuration files or a combination of these, to a new data center may be another process of VM migration. This also can be done by applying hot or cold migration techniques . One can use Storage Migration to move a running or turned-on virtual machine and its storage to a new data store.
Change Host and Data store	Moving a virtual machine to a destination host and moving its disk or virtual machine folder to another data store. One can change the host and data store using cold or hot migration. With vMotion technique also , the VM can be migrated to a new host and data store also simultaneously in environments without any shared storage

1.3 Type of VM Migration algorithm

VM Migration algorithms may be broadly categorized into three different segments :

Heuristic Algorithm – In such algorithms the set of constraints are problem dependent and deliver solutions to a problem in a restricted time. Most of the algorithms under such category are greedy algorithms.Several researchers re working in heuristic algorithm development for live VM migration. Abdullah, M., Lu, K., Wieder, P(2017) works on dynamic VM consolidation using a heuristic approach. The researchers have proposed a best fit algorithm for intelligent VM allocation and they have also proposed the dynamic utilization rate[19]. Where as some other group of researchers used heuristic approach to solve virtual machine scheduling prolem[20]. There are works on location selection policy to solve the overloading problem of virtual machines using this heuristic approach.

Meta Heuristic algorithms – They are mainly general purpose algorithms and nature inspired. VM Migration algorithms based on Ant colony optimization,particle swarm optimization or bee colony optimization, queen bee optimization falls under such category. A survey has been made upon several meta heuristic algorithms for virtual machine scheduling[21]. Genetic evolutionary approach, load balance oriented meta heuristic algorithm, genetic approach combined with knapsack problem there are seversal dimensions of meta heuristic approaches for VM migration process.

Hybrid Algorithms - A combination of heuristic and meta heuristic algorithms

2. Motivation of the Study

Cloud , being the fastest growing technology in this present technological era, faces the problem of energy efficiency.Cloud data center is a shared pool of computing as well as communication resources[17]. Due to highly competitive environment in this sector each and every service provider wants to satisfy their customers with a highly optimized service having 24 x 7 service availability, high storage, optimized level of multi tenancy and virtualization etc. But while meeting the QoS factors without any compromise, most of the cloud providers have to invest a lot in terms of power consumption in data centers also [2-4]. An idle server consumes 70% of the peak power[11] and hence leads to major energy efficiency . So today researchers have started work on this not only to protect the environment but also make their service more economic also[4].Green cloud revolution has started very recently.Green cloud computing not only limited to energy consumption but also concerned about reducing the E waste[17]. A survey has been made by the authors to enlighten the scopes and challenges in this area. (Singh N & Dhir V.,2017). Some researchers (Banerjee A. et. Al ,2013) have proposed energy efficient models[3] . Some problems have been solved by linear regression method or other statistical methods , others have preferred working on hardware level . While performing hardware level operations for making it more energy efficient, DVFS is one such hardware based algorithm along with live VM migration(Patel V & Bheda H., 2014). The other relevant works can be found in some other papers (Forsman et al., 2015; Sun et al., 2016; López-Pireset al., 2018; Maleklooet al., 2018

From the study of various migration techniques , it indicates a direction towards the migration of Virtual machine to be one of the most popular load balancing techniques or popular energy efficient algorithm for reducing power consumption while maintaining the QoS parameters . VM Migration algorithm includes constraints and equalities in the algorithm as valid conditions[9]

3. Study of Several VM Migration Techniques

In present era several researchers doing researches on virtual machine migration techniques. Banerjee A. et al(2013), in their paper have discussed about several energy efficient tools that are already available in the industry and they also discussed about various dimension of energy consumption while maintaining the QoS of the cloud service, which is actually a big challenge[4] . Among those techniques energy aware resource scheduling, cluster based energy conservation protocols. The paper is based on generic approach and the authors also proposed an energy efficient scheduling method based on hyper graph and its matrix representation. But the implementation section is not highlighted in this paper[4].

Singh N & Dhir V.(2017) have conducted a survey on several VM Migration techniques which can be helpful to get a clear view about the present trend and future direction of energy efficient cloud algorithms [3].This has created a pathway for the cloud researchers in this arena.

One feasible solution for VM Migration technique has been proposed by Patel V & Bheda H.(May, 2014) by using DVFS technique for making energy efficiency for real time data in cloud data centers. The authors have introduced the hardware based Dynamic Voltage Frequency Scaling (DVFS) concept for live VM migration. This papers proposes a technique by adjusting the system voltage based on CPU utilization[13]. When the workload is more, real time migration can be provided for effective usage of resources.

By applying this technology, without the requirement of restarting the power supply, system voltage and frequency can be adjusted in accord with the specification of the actual CPU design into a dissimilar working voltage. While CPU works in lower voltage, the energy consumption can successfully be saved. Here the solution can be achieved by CPU utilization monitoring but for an overloaded CPU when the numbers of VMs are nearer to maximum , then this solution faces some criticality to achieve the expected result.

Huang, C. J. & et. Al, (2013) in their work have discussed about adaptive resource scheduling for cloud computing to make it more energy efficient [2]. A genetic algorithm based resource dispatched technique has been proposed in this paper. This paper discusses about energy efficiency as well as QoS guaranty.

Deshai R. M & Patel B. H (April,2015) have discusses about energy efficient algorithm using virtual machine migration[15]. The authors have proposed a characteristic based compression technique for VM migration. Their proposal follows a good path to solve overloaded resources but it takes a long time for migration which creates problem in maintaining QoS of cloud service as most of the cloud service providers promise in their Service Level Agreement(SLA)[15].Deshpandey U. and Keahey K.(2017) has proposed a live VM Migration technique which is traffic sensitive[8]. In their study the authors have adapted an approach which reduces the contention between the migration technique and the traffic load.

Raval N and Thakkar R(2016) has performed a survey among existing VM migration techniques. The authors have performed there study regarding several compression techniques and their optimizability and working procedure. The authors have proposed a characteristics based algorithms using Run Length Encoding.

Table 2: Comparative study of papers on energy efficient algorithms for VM Migration (selected)

Paper Title	**Authors' Name**	**Journal/ Conference Name(With volume and Issue, if available**	**Main approach**	**Remarks**
Efficient VM Migration technique for energy reduction in cloud computing	Singh N & Dhir V	International Journal of Advanced Research in Computer Science, volume 8, issue 9	Survey among several VM migration techniques for energy efficiency in cloud computing	Give insights about several techniques used
Energy Efficiency Model for cloud Computing	Banerjee A. et al	International Journal of Energy, Information and CommunicationsVol.4, Issue 6	Energy aware resource schedule and cluster based energy conservation protocols	Various dimensions of energy consumptions along with QoS parameters performance intact.But lack in the implementation area
An adaptive resource management scheme in cloud computing	Huang, C. J. & et. al	Engineering Applications of Artificial Intelligence, 26(1)	Adaptive Resource scheduling algorithm of cloud computing	Provided AI included solution
Dynamic virtual machine consolidation for improving energy	Deng, D., He, K., & Chen, Y	.Cloud Computing and Intelligence Systems	Reducing energy consumption and SLA violation	QoS parameter is not well maintained

efficiency in cloud data centers				
Efficient Virtual Machine Migration in Cloud Computing	Desai R. M & Patel B. H	Fifth International Conference on Communication Systems and Network Technologies	Characteristics based compression algo for reducing total migration time along with total downtime	Too much dependent on defined characteristics
Reducing Energy Consumption with Dvfs for Real-Time Services in Cloud Computing	Hitesh Bera	IOSR Journal of Computer Engineering	Using DVFS based live migration technique to to increase efficiency of energy management in real time	Hardware dependent and complex architecture to manage
Reducing VM Migration Time By Compression Algorithm And Setting Threshold Of Dirty Page Rate	Raval M and Thakkar R.	International Journal of Innovative Research in Technology	The paper discusses about different VM Migration categories and a characteristics based algorithm along with memory compression technique has been proposed	

The approach of VM Migration based on Dynamic Voltage scaling(DVFS) may be applied using CPU Monitoring and it is primarily dependent on hardware resources . The main approach of such techniques is to decline the CPU voltage(V) and frequency(F) to reduce the power consumption(P) according to the following formulae :

$$\mathbf{P} = \mathbf{V}^2 * \mathbf{F} * \mathbf{C} \qquad \textit{\textbf{Equation 1}}$$

In the above formula Power consumption(P) is dependent upon Voltage level(V), Frequency(F) and Capacity of the system(C).

$$V_{CPU} = \frac{MIPS(VM)}{MIPS(HOST)} \qquad \textit{\textbf{Equation 2}}$$

$$Utilization\ of\ CPU = \sum_{i=1}^{n} V_{CPU} / MIPS(HOST) \qquad \textit{\textbf{Equation 3}}$$

But this process may generate some voltage drop due to CPU frequency reduction as well as generating high overhead. So this may not be very effective techniques for VM Migration in real time.

4. Conclusions and Future Direction

This study discusses about energy efficient cloud computing algorithms, energy consumption requirement by cloud computing and among other several solutions , live Virtual Machine migration technique as a very effective solution. Through this study the authors has tried to focus on heuristic, meta heuristic as well as hybrid approach of live VM migration algorithms.

In conventional cloud data centers power consumption was in its highest level, in such cases,specially when the system has an overload, virtual machine migration technique may be effectively applied. One of the significant challenges of the VM migration technique is the time required to migrate the VM from one data center to other. Storage reallocation is also another challenge for VM Migration. Though there are plenty of research works in this area , still it leaves ample scope for the researchers to generate feasible and optimized solutions. The new paradigm shift is being directed from cloud towards fog computing and also the use of predictive modelling for VM migration leaes a huge scope for exploring this area.

References

[1]. What is Green Cloud? "https://www.cloudoye.com/wiki/w/what-is-green-cloud", accessed on May 07,2019:12.30 am , New Delhi, India

[2]. Huang, C. J. & et. al., (2013). "An adaptive resource management scheme in cloud computing". *Engineering Applications of Artificial Intelligence*, 26(1), 382-389.

[3]. Singh N & Dhir V.(2017). "A literature review: Efficient VM Migration technique for energy reduction in cloud computing", *International Journal of Advanced Research in Computer Science*, 8(9), Nov–Dec, 2017,98-114

[4]. Banerjee A. et al(2013), "Energy Efficiency Model for Cloud Computing". *International Journal of Energy, Information and Communications*, Vol.4, Issue 6(2013), pp.29-42

[5]. Borgetto, D., &Stolf, P. (2014). "An energy efficient approach to virtual machines management in cloud computing.In Cloud Networking (CloudNet)", IEEE 3rd International Conference on (pp. 229-235).

[6]. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., &Buyya, R. (2011). "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms". *Software: Practice and experience*, 41(1), 23-50

[7]. Deng, D., He, K., & Chen, Y. (2016). "Dynamic virtual machine consolidation for improving energy efficiency in cloud data centers". *4th International Conference on In Cloud Computing and Intelligence Systems (CCIS),* 2016 (pp. 366-370).

[8]. Deshpande, U., & Keahey, K. (2017). "Traffic-sensitive live migration of virtual machines". *Future Generation Computer Systems*, 72, pp.118-128.

[9]. Chaima Ghribi, Makhlof Hadiji and Djamal Zeghlache, "Energy efficient VM scheduling for cloud data centers: exact allocation and migration algorithms", in *13thIEEE/ACM International Symposium on Cluster, Cloud, and GridComputing(CCGrid)* ,pp.671-678, May 2013

[10]. Prasher A. & Bhatia R(2014). "A Review On Energy Efficient Cloud Computing Algorithms",*International Journal of Application or Innovation in Engineering & Management*, volume 3,no 4, April 2014,pp. 364-368

[11]. E. Naone, "Conjuring clouds," *Technology Review*, vol. 112, no. 4, pp.54–56, 2009

[12]. https://pubs.vmware.com/ , accessed on May 10,2019 : 02.30 pm , New Delhi, India

[13]. Patel V & Bheda H.(May, 2014) "Reducing Energy Consumption with DVFS for Real-Time Services in Cloud Computing" *IOSR Journal of Computer Engineering (IOSR-JCE)* Issue 3, Ver. II (May-Jun. 2014), pp. 53-57. DOI: 10.9790/0661-16325357

[14]. López-Pires, F., Barán, B., Benítez, L., Zalimben, S., & Amarilla, A. (2018). "Virtual machine placement for elastic infrastructures in overbooked cloud computing datacenters under uncertainty". *Future Generation Computer Systems*, 79, 830-848.

[15]. Desai R. M & Patel B. H (April,2015), "Efficient Virtual Machine Migration in Cloud Computing", *Fifth International Conference on Communication Systems and Network Technologies*, Gwalior,India

[16]. https://en.m.wikipedia.org/wiki/Live_migration, accessed on May 11,2019:12.30 pm, New Delhi, India.

[17]. Radu L.D.(November, 2017), "Green Cloud Computing: A Literature Survey",MDPI, Symmetry 2017, 9, 29

[18]. Raval N & Thakkar R. (2016),"Reducing VM Migration Time By Compression Algorithm And Setting Threshold Of Dirty Page Rate", *International Journal of Innovative Research in Technology*,2(12),pp.45-55

[19]. Abdullah, M., Lu, K., Wieder, P., "A Heuristic-Based Approach for Dynamic VMs Consolidation in Cloud Data Centers", *Arab J Sci Eng* (2017) 42: 3535. https://doi.org/10.1007/s13369-017-2580-5.

[20]. Moalla F, Balma A. and Mrad M (December,2017), A Rapid Heuristic For The Virtual Machines Migration Scheduling Problem, *International Conference on Control and Signal Processing*,Tuinisia

[21]. Khanchi M nd Tyagi S,"VM Scheduling in Cloud Computing using Meta-heuristic Approaches", *International Journal of Scientific & Engineering Research*, Volume 7, Issue 12, December-2016

6. Authors' Biography

Dr. Santanu Kumar Sen, received BE(CSE), M.Tech (CSE), MBA (IS) and PhD(Engg.) from REC Silchar and Jadavpur University respectively. He is a Fellow of IET(UK), IE(I), IETE(I) and Sr. Member of IEEE (USA), CSI(I) and life members of ISTE. Presently he is working as Professor and Principal in Gurunanak Institute of Technology. He has around 25 years of experience in the field of Computer Science and Engineering in which 8 years in Industry and 17 years in Engineering Academia including Abroad.

He has got Rashtriya Shiksha Gourav Puroskar from Centre for Education Growth and Research (CEGR) in 2016

,Academic Excellence Award – Special Leadership Award from JIS Group in 2017,Indira Gandhi Sadbhavna Award" from Global Achievers Foundation in 2014, Bharat Bibhushan Samman Puraskar in 2013.He has more than 70 research papers and 4 patent has been filed under his supervision.

His research interests are Computer Network, Network Security, Routing algorithms ,Cloud Computing, IoT Security, Big Data,

Machine Learning, Deep Learning.

Sharmistha Dey, received B. Sc(CU), MCA(WBUT), M Tech(CU) She is working as a research scholar in West Bengal University of Technology, West Bengal. Her research area is Cloud Security.

She has published several papers in International journals and conferences like IEEE,Springer,MGH.

Her research interest is Cloud Computing, Wireless Network, Cyber Security, Machine Learning, Data Analytics, Artificial Intelligence.

Dr. Rajib Bag was born in 1969, received his B.Sc (Physics Hons.) from Calcutta University, M.Sc. (Physics) from Vinoba Bhave University and M.Tech. & Ph.D (Engg.) from Jadavpur University, India in the year of 1991, 1996, 2007 & 2012 respectively. His doctoral work was in the field of control systems. Presently, he is working as a Professor & Head in the department of Computer Science & Engineering at Supreme Knowledge Foundation Group of Institutions under Maulana Abul Kalam Azad University of Technology, West Bengal, India. He has more than 40 publications in reputed refereed journals and conference proceedings to his credit. Presently five research scholars are doing their research work in different areas under his supervision. His research interest includes image and signal processing, education technology , machine learning, deep learning and IOT security besides control systems.

How to Cite

Sen,Santanu Kumar, Dey, Sharmistha and Bag, Rajib (2019). Study of Energy Efficient Algorithms for Cloud Computing based on Virtual Machine Migration Techniques. *International Journal of Machine Learning and Networked Collaborative Engineering*, 3(02) pp 93-101.

doi : https://doi.org/10.30991/IJMLNCE.2019v03i02.003**.**

Vol. 03, No. 2,(2019),102–113, **ISSN:2581-3242**

IJMLNCE JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2019v03i02.004

Information Processing in GLIF Neuron Model with Noisy Conductance

[a]Vishwadeepak Singh Baghela[*], [b]Sunil Kumar Bharti, [c]Saket Kumar Choudhary,

[a]SET, SVU, Gajraula, UP, India

[b]CSE, GEC, G B Nagar, UP, India

[c]FCA, MRIIRS, Faridabad, Haryana, India

[a]vdsbaghela@gmail.com (https://orcid.org/0000-0001-6201-4254)

[b]sirbharti@gmail.com (https://orcid.org/0000-0002-0245-6496)

[c]saket_au@rediffmail.com, skchoudhary.fca@mriu.edu.in (https://orcid.org/0000-0002-0448-3076)

Abstract

In this article, we investigate the generalized leaky integrate-and-fire (GLIF) neuron model with stochastic synaptic conductance. A neuron remains connected with other neuron via dendrites and axons at synapse, which can be treated as an electrical capacitor. Dendrites carry electro-chemical signals from input neuron to synapse whereas axons are responsible for their transmission form synapse to other neurons. Concentration of these electro-chemicals in synapse varies during entire time period. We investigate the effect of varying concentration of electro-chemicals at synapse in a single neuron model. Concentration variation of electro-chemicals at synapse is incorporated as noise in GLIF model. Excitatory and inhibitory synaptic conductance of neuron in GLIF is assumed as stochastic entities driven by Gaussian White noise. Stationary state membrane potential distribution for the proposed model is computed with reflecting boundary conditions, which is noticed as geometrically distributed. In order to investigate spiking activity and information encoding mechanism, an extensive simulation based study has been carried out. Temporal encoding technique is used to analyze the encoding mechanism. It is noticed that ISI distribution has higher variance with respect to excitatory input than inhibitory input. ISI distribution also exhibits the power-law behavior for electro-chemical balance situation.

Keywords

Colour Noise,

Conductance,

GLIF Model,

GIF Model,

ISI Distribution,

Stochastic,

Decay Constant

1. Introduction

Neuron processes information in form of action potential (spike) and transmits in sequence of spikes [11, 16]. These spike sequence exhibit the highest scale of variability in their patterns. Variability in spike

[*] Corresponding author

Vishwadeepak Singh Baghela

Email: vdsbaghela@gmail.com

sequence contains encoded information which transmits to other neurons or part of body [11, 16, 19]. A number of measurements have suggested the reliability of encoded information into variable spike sequences. Neurotransmitters, which are responsible for membrane potential fluctuation, can be categorized in two classes: namely, excitatory neuro-transmitters and inhibitory neuro-transmitters [4, 6, 22]. Excitatory neurotransmitter increases the membrane potential whereas inhibitory neurotransmitter decreases the membrane potential value. Random arrival of these neurotransmitters causes fluctuation in net potential value; which also affect the spiking activity and generates variability in spiking pattern [4, 6]. Potential value contributed by these neurotransmitters can be classified as excitatory potential and inhibitory potential and their arrival pattern has been modeled via Poisson process i.e. arrivals are independent of each other [10, 19, 23].

Lapicque has introduced the integrate-and-fire (IF model) neuron model which is also the first mathematical model of neuron [1, 25]. In IF model, it is assumed that a neuron receives potential in form of neurotransmitters from other neurons and external world in form of input, which increases its membrane potential and at a fixed potential value (threshold), it emits collected neurotransmitters and generate an action potential. Leaky Integrate-and-Fire (LIF) neuron model is an extension of IF model, where membrane decay constant ( ) has been incorporated [1, 14, 25]. Mathematical representation for rate of change of membrane potential in LIF model can be given as below.

$$\frac{dV}{dt} = - \quad V(t) + I(t) \tag{1}$$

Here $I(t)$ is the time dependent input stimulus. Many researchers have modeled $I(t)$ in different ways viz. constant value, periodic value, stochastic value driven by Gaussian white noise, *etc*. A number of researcher have assumed $I(t)$ as a sum of current due to excitatory neurotransmitter and inhibitory neurotransmitter [2, 3, 22] i.e. $I(t)$ can be replaced via

$$I(t) = G_e(V_e - V)S_e + G_i(V_i - V)S_i \tag{2}$$

Here, G_e and G_i are excitatory and inhibitory synaptic conductance, S_e and S_i are excitatory and inhibitory synaptic strengths. Substitution $I(t)$ of from Eq. (2) to IF model and LIF model transform them into Generalize Integrate-and-Fire (GIF) and Generalized Leaky-Integrate-and-Fire (GLIF) neuron models [2, 3, 28]. In this article, we investigate spiking activity, information processing mechanism and stationary state membrane potential for GLIF model with stochastic synaptic conductance.

The article is structured in 6 sections. After a brief introduction about IF model, spiking variability in section 1, Section 2 deals with the formulation of GLIF model. Here stochastic synaptic conductance has been used to model the assumption. Section 3 deals with mathematical computation of GLIF model and stationary state membrane potential for GLIF neuron model is computed. Information processing mechanism into GLIF model is investigated in next section 4. Here, a detailed simulation based study is performed to compute ISI distribution. Discussion related to model and findings has is elaborated in section 5. Finally, the last section contains conclusions and future scope for the study.

2. Generalized Leaky Integrate-and-Fire Neuorn Model

Driving After more than 50 years of Lapicque's IF model, Stein has proposed GIF model where input stimulus is modeled as a sum of pre-synaptic excitatory and inhibitory currents [2, 3, 28]. Furthermore, Stein has assumed the arrivals of pre-synaptic current as a Poisson process and investigated information processing in terms of neuronal firing rate [27]. Wilbur and Rinzel [29] have investigated Stein's Generalized IF model only for excitatory neurotransmitters. They have measured multiple parameters like firing rate, calculation time etc. which is important for physiological point of view. Wilbur and Rinzel [29] have also investigated the inter-spike-interval distribution (ISI distribution). Tuckwell [26] has studied the firing rate of GLIF model with excitatory current and inhibitory current, both, and noticed a fine agreement in firing rate with existing experimental data. Richardson and Gerstner [20] have studied GLIF model with stochastic synaptic conductance. They have modeled synaptic conductance via Ornstein-Uhlenbeck process and have computed voltage distribution, conductance distribution, ISI distribution etc. Lansky [17] has assumed evolution of membrane potential in GLIF model as a diffusive process and computed ISI distribution. He [17] has also computed the cumulative ISI distribution for exponentially distributed excitatory and inhibitory neurotransmitters. Hurby [13] has studied GLIF model with realistic synaptic potential and noticed bursting period, quiescence period, spiking rate and spike frequency. In order to avoid membrane potential fluctuations, Goris et. al. [12] modeled spike generation in GLIF model as a Poisson process and investigated response distribution with excitatory neurotransmitters. This model [12] is found comparatively more suitable for visual sensory neurons. Teeter et. al. [24] has applied unsupervised methods with GLIF neuron to classify cell types in mammalian neocortex. Choudhary et. al. [4], Choudhary and Solanki [6] has modeled membrane potential contribution due to excitatory neurotransmitters and inhibitory neurotransmitters via hypo-exponential distribute delay kernel in distributed delay framework (DDF) of threshold based neuron model. Their model is capable to generate different kinds of spiking patterns as reported in multiple literatures. Uni-modal, bi-modal, multi-modal etc. kind of ISI distribution patterns has been noticed in their investigation [4, 6]. DDF provides a way to capture the effect of past values of membrane potential over its present evolution so that variability in spiking pattern can be explained in better way [7]. Stationary state membrane potential distribution of LIF model in DDF exhibit no change under the change of delay kernel functions and statistically remains constant which is noticed as Gaussian distributed [6, 7].

3. Stationary State Probability Distribution in GLIF Model

Substitution of $I(t)$ from Eq. (2) to Eq. (1) yields [2, 3]

$$\frac{dV}{dt} = - \quad V(t) + G_e(V_e - V)S_e + G_i(V_i - V)S_i \tag{3}$$

Here V_e and V_i are initial value of excitatory potential and inhibitory potential. Further simplification of Eq. (3) results into GLIF as

$$2_m \frac{dV}{dt} = -(V - V_0) + G_e(V_e - V)S_e + G_i(V_i - V)S_i \tag{4}$$

2_m is membrane resistance. It is noticed in multiple literature that electrical properties of membrane changes after each spike so that synaptic conductance also changes [20]. Thus, excitatory and inhibitory synaptic conductance G_e and G_i can be assumed to be a time dependent entities so that Eq. (4) can be rewritten as

$$2_m \frac{dV}{dt} = -(V - V_0) + G_e(t)(V_e - V)S_e + G_i(t)(V_i - V)S_i \tag{5}$$

Following Richardosn and Gerstener [20], $G_e(t)$ and $G_i(t)$ are assumed to be stochastic entities. In forgoing study, we are modeling these two entities driven by Gaussian white noise i.e. $G_e(t) = \bar{G}_e + \P_e(t)$ and $G_i(t) = \bar{G}_i + \P_i(t)$ [5, 9, 23]. Here $\bar{G}_e$ and $\bar{G}_i$ are mean value of excitatory and inhibitory synaptic conductance. $\P_e(t)$ and $\P_i(t)$ are mutually exclusive Wiener processes driven by Gaussian white noise with intensity , $<\pm_e >= 0$, $<\pm_i >= 0$,respectively, i.e. $<\P_e(t)\P_i(t) >= 0$, $<\P_e(t) >= 0$, $<\P_i(t) >= 0$, $<\P_e(t_1)\P_e(t_2) >= \frac{\pm_e^2}{2}$ and $<\P_i(t_1)\P_i(t_2) >= \frac{\pm_i^2}{2}$. Thus Eq. (5) simplifies as

$$2_m \frac{dV}{dt} = -(V - V_0) + (\bar{G}_e + \pm_e dW_e(t))(V_e - V)S_e + (\bar{G}_i + \pm_i dW_i(t))(V_i - V)S_i \tag{6}$$

Further simplification of Eq. (6) results into

$$dV = -(AV - B)dt + \left(\frac{\pm_e S_e (V - V_e)}{2_m} dW_e \right) + \left(\frac{\pm_e S_e (V - V_i)}{2_m} dW_i \right) \tag{7}$$

Here, $A = \frac{1 + \bar{G}_e S_e + \bar{G}_i S_i}{2_m}$ and $B = \frac{V_0 + \bar{G}_e V_e S_e + \bar{G}_i V_i S_i}{2_m}$

Let $p(V,t)$ be the spatial probability distribution corresponding to the membrane potential $V(t)$ described by Eq (7), then its associated Fokker-Planck equation can be given as

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial V}\{-(AV - B)p\} + \frac{1}{2}\left(\frac{\pm_e^2 S_e^2 + \pm_i^2 S_i^2}{2_m^2} \right) \frac{\partial^2}{\partial V^2}(V^2 p) \tag{8}$$

The probability current-flux $J(V,t)$ associated with Eq. (8) takes the form

$$J(V,t) = (AV - B)p + \frac{1}{2}\left(\frac{\pm_e^2 S_e^2 + \pm_i^2 S_i^2}{2_m^2} \right) \frac{\partial}{\partial V}(V^2 p) \tag{9}$$

In order to obtain stationary state probability distribution of membrane potential with reflecting boundaries

$$J(V,t) = 0 \tag{10}$$

Simultaneous use of Eq. (9) and Eq. (10) results

$$(AV - B)p + \frac{1}{2}\left(\frac{\pm_e^2 S_e^2 + \pm_i^2 S_i^2}{2_m^2} \right) \frac{\partial}{\partial V}(V^2 p) = 0 \tag{11}$$

Further simplification of Eq. (11) results

$$V^2 \frac{\partial p}{\partial V} = -\left(\frac{2A}{T^2} + 2 \right) V + \frac{2B}{T^2} \tag{12}$$

Here $\left(\frac{\pm_e^2 S_e^2 + \pm_i^2 S_i^2}{2_m^2} \right) = ¸$ Integration of Eq. (12) with reflecting boundary conditions results

$$p = V^{-\mu} e^{- \ /V} \tag{13}$$

Here, $2\left(\frac{A}{¸^2} + 1 \right) = \mu$ and $\frac{2B}{¸^2} =$. When V is very large as compared to i.e. $\frac{}{V} \to 0$ then Eq. (13) takes the form becomes

$$p = KV^{-\mu} \tag{14}$$

Here K is a constant value.

4. Infomation Processing in GLIF Model

In order to investigate the information processing mechanism in the GLIF model with proposed stochastic conductance, an extensive simulation based study is performed. We apply Monte-Carlo numerical simulation technique to investigate associated stochastic differential equation in the proposed GLIF model.

There are a number of numerical simulation methods are proposed in literature, we apply Euler-Maruyama (EM) scheme in the simulation study [15]. Following EM scheme, the time duration T for membrane potential evolution is divided into n equally spaced sub-intervals $[0,t_1]$, $[t_1,t_2]$,..., $[t_{n-1},t_n]$, each sub-interval has size $h = T/n$ which is also known as step size. According to Eq. (7), let V_i be the membrane potential at time $t = t_i$ then the membrane potential at successive time $t = t_{i+1}$ becomes

$$V_{i+1} = Vi - \left(AV - B\right)h + \left(\frac{\pm_e S_e (V - V_e)}{2_m} dW_e\right)\sqrt{\P_{ei}} + \left(\frac{\pm_e S_e (V - V_i)}{2_m} dW_i\right)\sqrt{\P_{ii}} \tag{15}$$

for $i = 1, 2, ..., n$, with initial values $V_0 = 0$, $X_0 = 0$ and $Y_0 = 0$. $\P_{ei}$ and $\P_{ii}$ are independent identically distributed standard Gaussian variates.

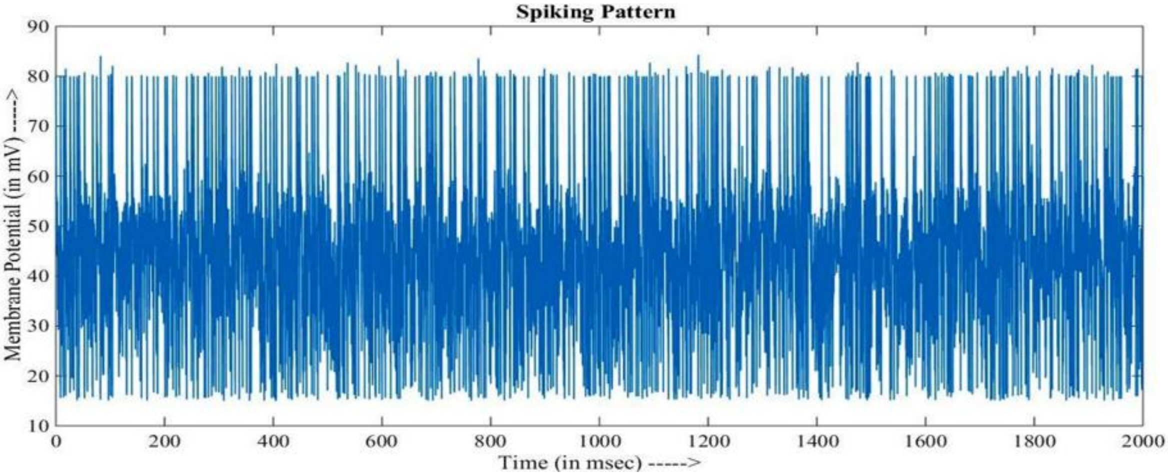


Fig. 1: Spiking Pattern in GLIF Model

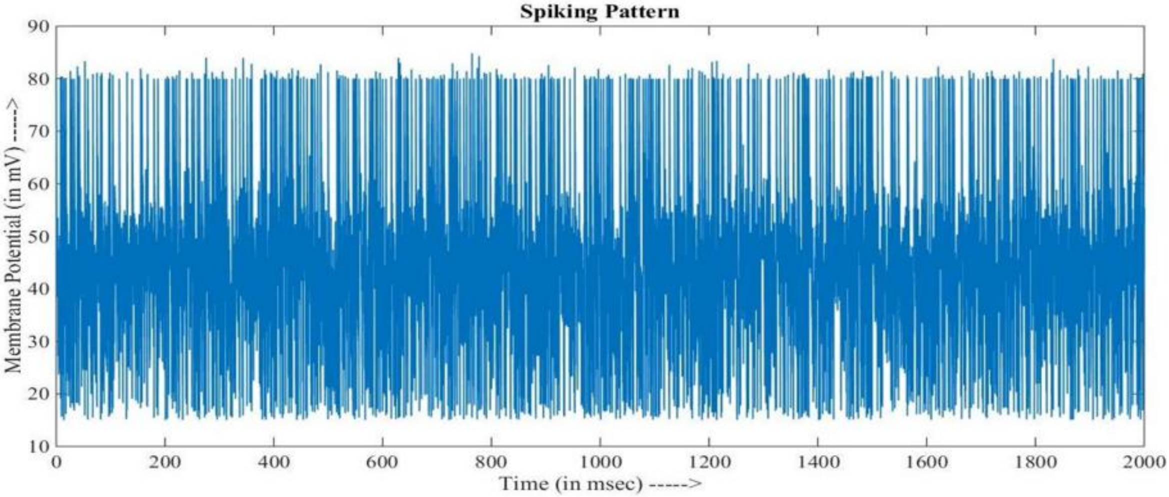


Fig. 2 : Spiking Pattern in GLIF Model

Neuron encodes information in two ways, namely, rate encoding scheme and temporal encoding scheme [19, 22, 23]. We apply temporal encoding scheme to investigate information processing mechanism in the GLIF model. Neuron uses time interval between two consecutive spikes to encode information in rate encoding scheme. The probability distribution of this time intervals is known as inter-spike-interval (ISI) distribution. In order investigation information processing mechanism in GLIF model, spiking pattern of the propose model, for different combination of parameters, is investigated. For similar combination of parameter values, ISI distribution of the model is also studied. Combination of parameter values is given in Table 1. Spiking pattern for the GLIF model is shown in Fig 1. to Fig 5 whereas Fig 6 to Fig 10 illustrates ISI distribution for the GLIF model with parameter combinations given in Table 1. We use^{2} , V_0 , $V_{threshold}$, V_{reset} , V_E^0 and V_I^0 as a constant throughout the simulation study. Their values are 1, -80, -15, -80, -15 and -75, respectively. We varied rest of six parameter values as given in the following Table1.

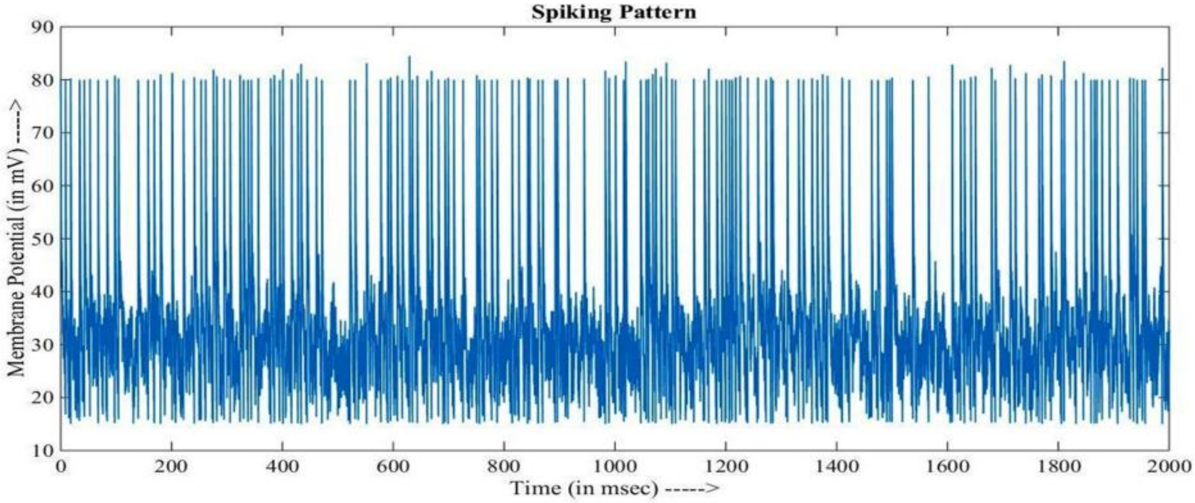


Fig. 3: Spiking Pattern in GLIF Model

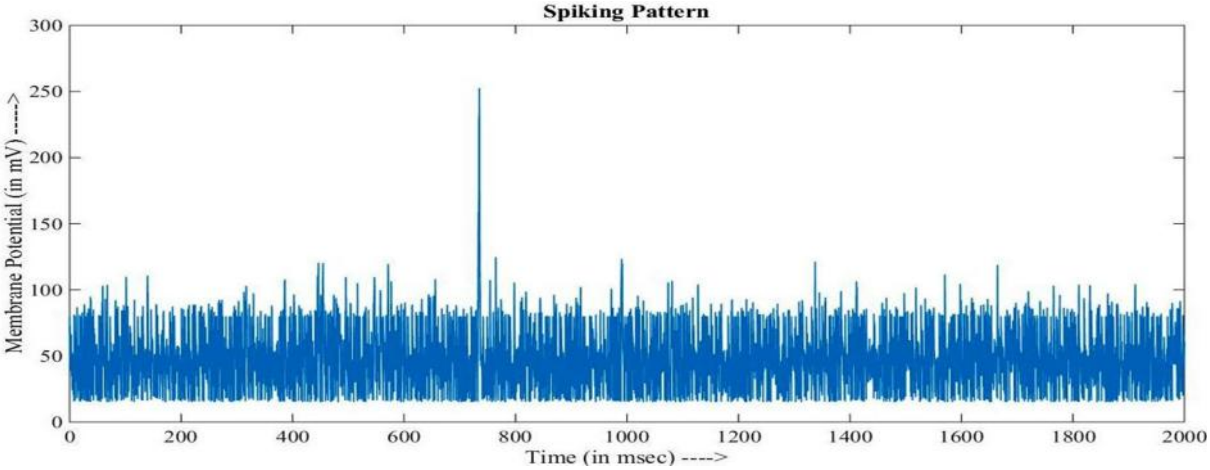


Fig. 4: Spiking Pattern in GLIF Model

Table 1: Combination of parameter values

Fig. No.	G_E	G_I	S_E	S_I	$\pm_E$	$\pm_I$
1, 6	0.5	0.7	1	-1.3	0.1	0.3
2, 7	0.5	0.7	1	-1.1	0.1	0.4
3, 8	0.5	0.7	1.1	-1.5	0.1	0.1
4, 9	0.5	0.7	1.1	-1.5	0.3	0.3
5, 10	0.3	0.5	1.1	-1.1	0.3	0.3

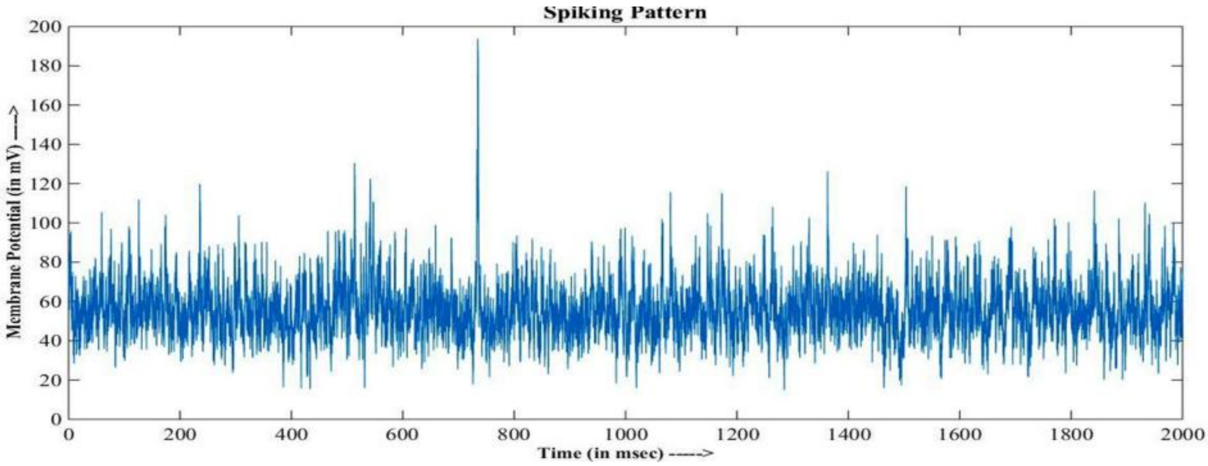


Fig. 5: Spiking Pattern in GLIF Model

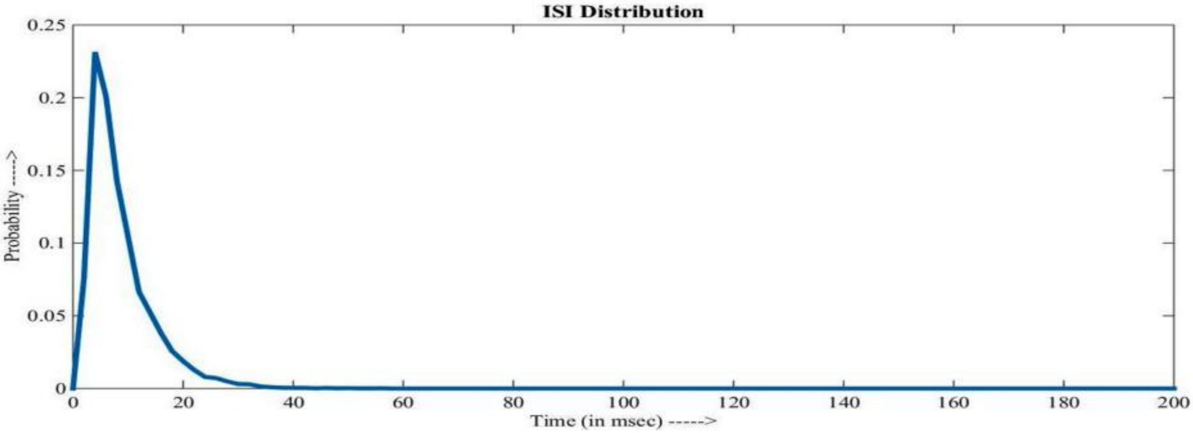


Fig. 6: Inter-Spike-Interval distribution for GLIF Model

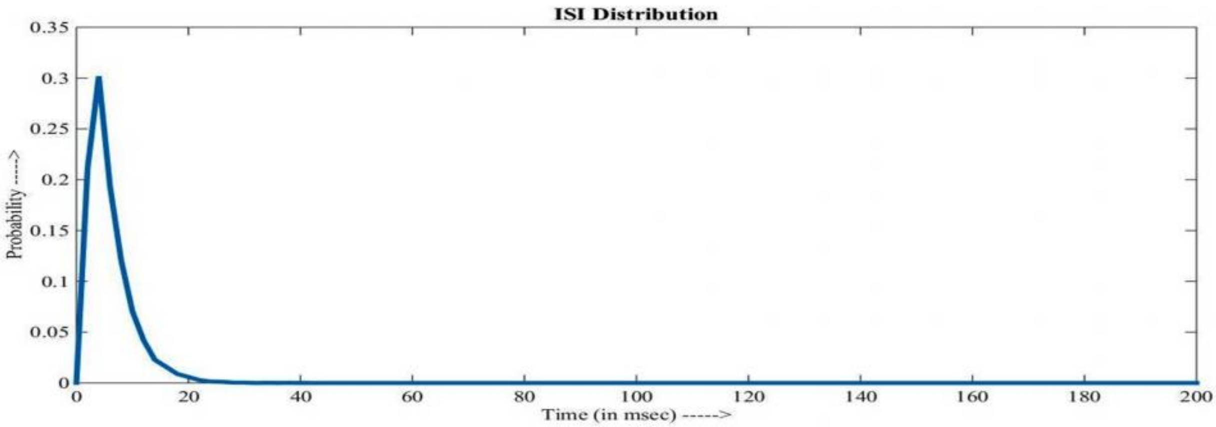


Fig. 7: Inter-Spike-Interval distribution for GLIF Model

Fig 1 to Fig 3 represent bursting nature in spiking pattern. This pattern is occurring due to small value of inhibitory input. When the value of inhibitory input (arrival rate and noise) is smaller, GLIF neuron achieves threshold value in quicker time as shown in Fig 1 and Fig 2. Further increment in inhibitory input as compared with excitatory input, the GLIF neuron exhibits noisy behavior in spiking pattern as shown in Fig 3. Here, inhibitory input moves neuron to opposite direction from firing threshold thus results into more random behavior in spiking pattern. Fig 4 and Fig 5 contains less spikes due to the further increase in inhibitory input than excitatory. Thus, it is noticed that inhibitory input (electro-chemicals) contributes to net membrane potential so that it move away from firing threshold. Fig 6 to Fig 10 represents ISI distribution for spiking activities of the GLIF neuron as shown in Fig 1 to Fig 5. Spiking patterns in Fig 1 to Fig 5 are shown only for initial 2000 msec time period whereas ISI distribution is obtained for 10000000 msec time duration. It is well illustrated in Fig 6 to Fig 9 that increase in inhibitory input leads GLIF neuron not to spike where as excitatory input enforces membrane potential to threshold value. This behavior of excitatory and inhibitory electro-chemicals increases the firing time of GLIF neuron so that the variance in ISI distribution increases. Fig 10 exhibits a situation when arrivals of excitatory and inhibitory inputs are approximately similar so that spiking rate of GLIF model reduces which decreases the spiking activity and increases the inter-spike-interval time duration. Fig 11 represents the ISI distribution on (shown in Fig 10) on Log-log scale. A straight line with slope 1.7102 is fitted on the scattered ISI distribution. Increase in the variance of the ISI time is due to the reason that electro-chemicals, working as a memory element, exhibits time dependent behavior.

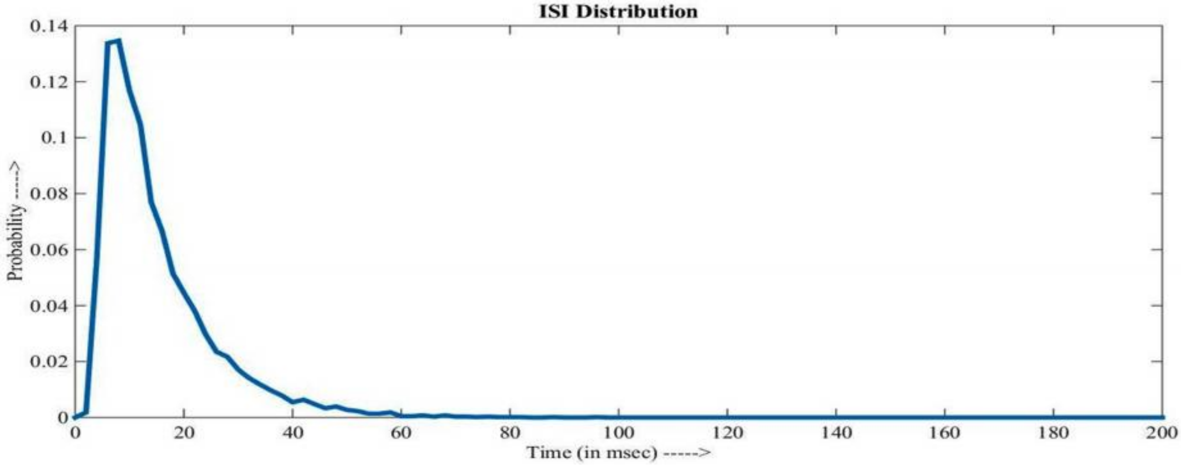


Fig. 8: Inter-Spike-Interval distribution for GLIF Model

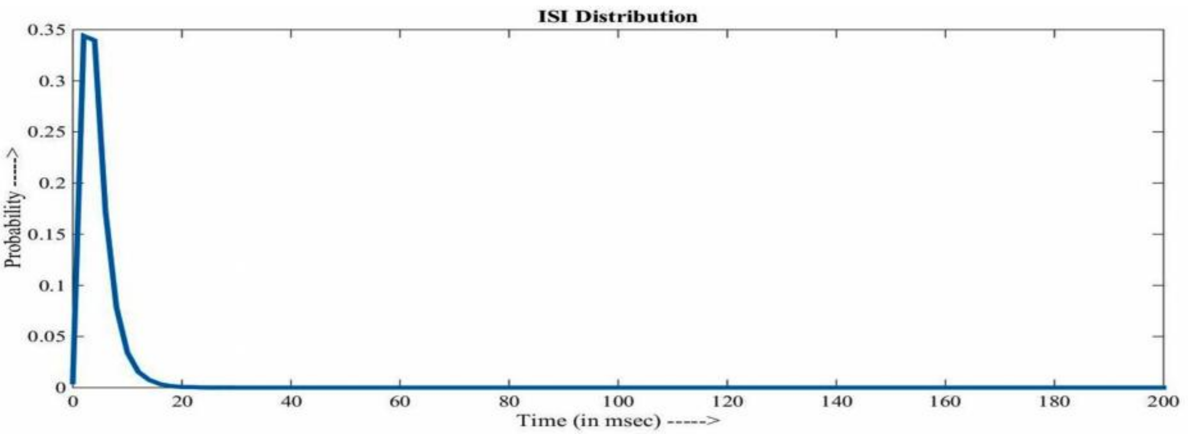


Fig. 9: Inter-Spike-Interval distribution for GLIF Model

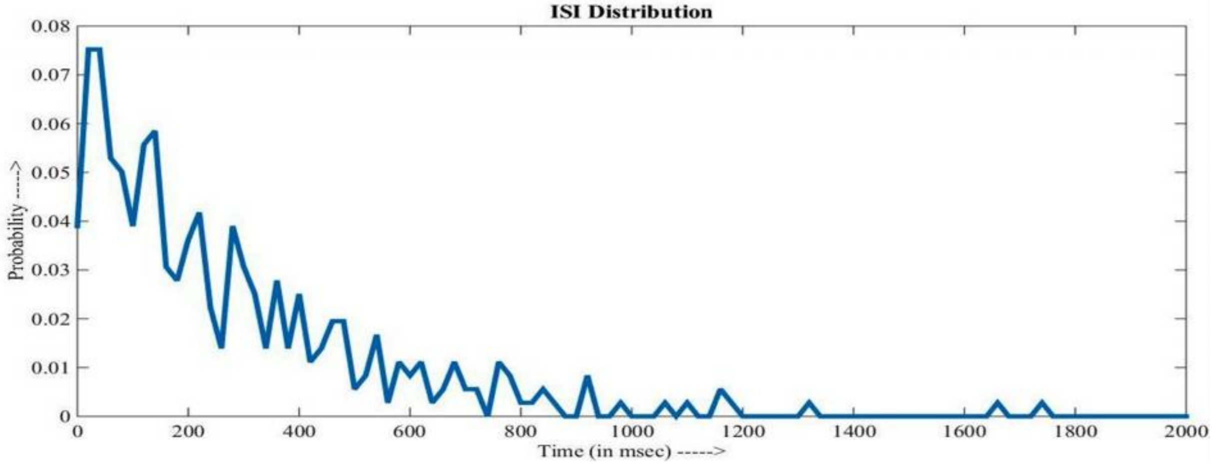


Fig. 10: Inter-Spike-Interval distribution for GLIF Model

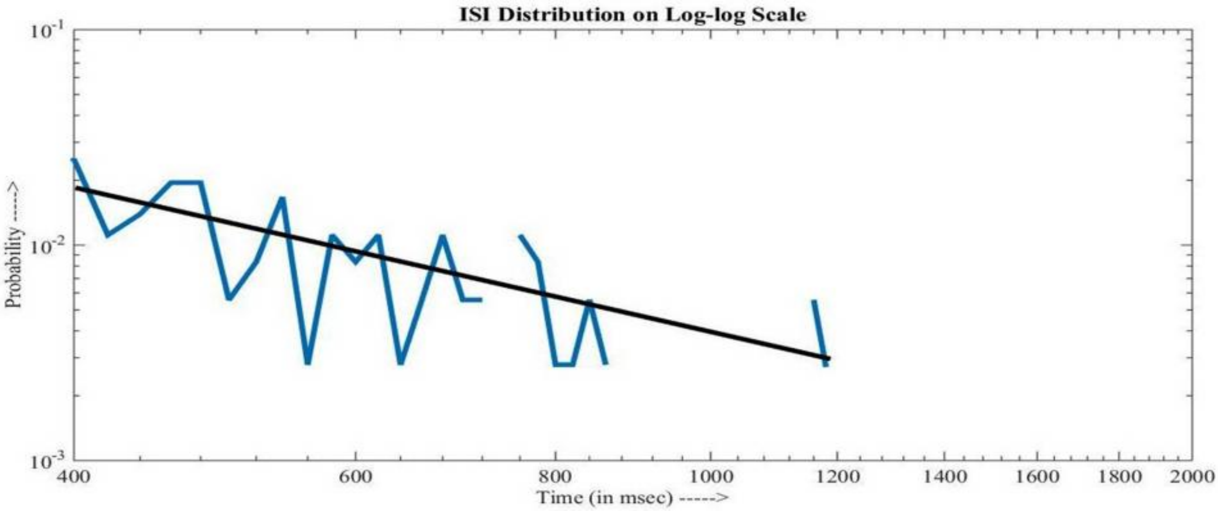


Fig. 11: Inter-Spike-Interval distribution on Log-log Scale for GLIF Model

5. Discussion

The power law is one of the prominent features for many-body problems [18]. A random variable having power-law behavior, can exhibit it in entire domain space or in a sub-domain space; which can be defined by the following generalized form [18]

$$p(x) = \begin{cases} \sim_1(x); x \in (-\infty, a) \\ x^{-\sim}; x \in [a,b] \\ \sim_2(x); x \in (b, \infty) \end{cases} \tag{16}$$

Here, a random variable A with an attribute x has probability distribution function $p(x)$. It has three sub-domains $(-\infty, a)$, $[a,b]$ and (b,∞) . $\pm_1(x)$ and $\pm_2(x)$ are two different functions in two different sub-domains $(-\infty, a)$ and (b,∞) . $x^{-\sim}$ is a third function in sub-domain $[a,b]$ with power-law behavior which has as a normalization constant and $\sim (> 0)$ as a constant exponent. In this way attribute x for random variable A have three different behaviors in their three respective sub-domains as defined in Eq. (16). It is a challenging task to compute parameter values like $(a,b,\sim)$ for a probability distribution. In order to obtain the value of constant exponent $\sim$ as defined in Eq. (16), maximum likelihood technique provides a way to examine the power-law behavior which maximizes logarithm of probability distribution. The power law behavior can also be examined on Log-log scale scattered distribution [18]. A straight line can be fitted in sub-domain having power law behavior on scattered distributions. This line fitting also helps to examine system's attribute evolution dynamics [8, 18]. A positive slope value of fitted straight line suggests linear increase in attribute whereas negative slope depicts the linear decrease. An attribute which has constant behavior in time domain, will have a slope value 0 for fitted straight line.

GLIF neuron has a kind of gamma distribution for membrane potential in its stationary state with reflecting boundary conditions as shown in Eq. (13). When the membrane resistance is very small as compared to the membrane potential, ISI distribution as shown in Eq. (13), reduces to exhibit the power-law behavior, which is noticed in Eq. (14). Occurrence of the power-law behavior for stationary state membrane potential suggest the long-range dependency for electro-chemicals moving to-and-fro from the synapses.

6. Conclusion and Future Scope

We investigate the GLIF model with stochastic synaptic conductance. Noise generated via multiple kinds of electro-chemicals, molecules and ions is captured via well established Winner process. Incorporation of Wiener process into the GLIF model, turn it into a system with colored noise where noise generates huge fluctuations as it becomes multiple of current value of the state variable [16, 23]. It also provides an essential parameter for long-range dependency of the membrane potential on the information. This long range dependency behavior in information processing occurs due to the memory elements such as electro chemicals. These elements contribute to membrane potential for large time interval results the power-law behavior in stationary state membrane potential distribution and ISI distribution of the GLIF model with stochastic synaptic conductance.

We have investigated membrane potential distribution in stationary state and temporal coding technique of information processing of the neuron. It will be interesting to investigate multiple other neuro-dynamical features of the proposed GLIF model with or without DDF.

References

[1] Abbott, L. F. and Dayan, P. "*Theoretical Neuroscience: Computational and mathematical modeling of neural systems*", The MIT press, 2001 [ISBN-10: 9780262541855].

[2] Burkitt, A. N. "*A Review of the Integrate-and-Fire Neuron Model: I. Homogeneous Synaptic Input", BiologicalCybernetics*, Vol. 95, No.1, pp.1-19, 2006, [DOI: 10.1007/s00422-006-0082-8].

[3] Burkitt, A. N. "*A Review of the Integrate-and-Fire Neuron Model: II. Inhomogeneous synaptic Input and Network Properties*", Biological Cybernetics, Vol. 95, No. 2, pp. 97-112, 2006.

[4] Choudhary, S. K., Singh, K. and Bharti, S. K. "*Variability in Spiking Pattern of Leaky Integrate-and-Fire Neuron Due to Excitatory and Inhibitory Potentials*", 2nd International Conference on "Computing for Sustainable Global Development", pp. 2025-2030, March 2015 [**INSPEC Accession Number:** 15110012**].**

[5] Choudhary, S. K. and Bharti, S. K., "Information Processing in Neuron with Exponential Distributed Delay*",International Journal of Machine Learning and Networked Collaborative Engineering,* Vol. 02 No. 02, 2018,pp 58-66. doi: https://doi.org/10.30991/IJMLNCE.2018v02i02.003.

[6] Choudhary, S. K. and Solanki, V. K. "*LIF Neuron with Hypo-exponential Distributed Delay: Emergence of Unimodal, Bimodal, Multi-Model ISI Distribution with long tail*" in Recent Patent on Engineering, Vol. 4, Issue, 4, 2019.

[7] Choudhary, S. K. and Solanki., V. K. "*Spiking Activity of a LIF Neuron in Distributed Delay Framework*," International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 3, No. 7, pp. 70-76, June 2016, [DOI: 10.9781/ijimai.2016.3710].

[8] Clauset, A., Shalizi, C. R. and Newman, M. E. J. "*Power-law distributions in empirical data*", SIAM Review, 51(4) (2009) 661-703.

[9] Destexhe, A. and Lilith, M. R. "*Neuronal Noise*," Springer Series in Computational Neuroscience, 2012.

[10] Gerstner, W. and Kistler, W. M. "*Spiking Neuron Models: Single Neurons, Populations, Plasticity*," Cambridge University Press, 2002, [ISBN-10: 0262514206].

[11] Gabbiani, F. and Koch, C. "*Principles of Spike Train Analysis. In: Koch C, Segev* I (eds) Methods in Neuronal Modeling: From Ions to Networks," MIT Press, Cambridge, 1998.

[12] Goris, R. L. T., Movshon, A. and Simoncelli, E. P. "*Partitioning Neuronal Variability*", Nature Neuroscience, Vol. 17, No. 6, pp. 858-867, 2014.

[13] Hruby, P. "*Analysis of Bursting in Stein's Model with Realistic Synapses*", Gen. Physiol. Biophys. pp. 305-311, 1995.

[14] Izhikevich, E. M. "*Which Model to Use for Cortical Spiking Neurons?*" IEEE Transactions on Neural Networks, Vol. 15, No. 5, pp. 1063-1070, 2004.

[15] Kloeden, P. E. and Platen, E, "*Numerical Solution of Stochastic Differential Equations*," Springer, Berlin, 1992.

[16] Koch, C. "*Biophysics of Computation: Information Processing in Single Neurons"*, Oxford University Press, 1998.

[17] Lansky, P. "*On Approximation of Stein's Neuronal Model*", Journal of Theoretical Biology, Vol. 107, pp. 631-647, 1984.

[18] Mansfield, M. L. "*Numerical tools for obtaining the power-law representations of heavy-tailed datasets*", The European Physical Journal B, 89:16 (2016) 1-13.

[19] McDonnell, M. D., Ikeda, S. and Manton, J. H. "*An Introductory Review of Information Theory in the Context of Computational Neuroscience*", Biological Cybernetics, Vol. 105, pp. 55-70, 2011.

[20] Richardosn, M. J. and Gerstener, W. "*Synaptic Shot Noise and Conductance Fluctuations Affect the Membrane Voltage with Equal Significance*", Neural Computation, Vol. 17, No. 4, pp. 923-947. 2005.

[21] Ross, S. M. "*Introduction to Probability Models*," 9th Edition, Academic Press, 2007.

[22] Smaili, K., Kadri, T. and Kadry, S. *Hypoexponential Distribution with Different Parameters*," Applied Mathematics, Vol. 4, pp. 624-631, 2013.

[23] Stevens, C. F. and Zador, A. "*Information through a Spiking Neuron"*, Advances in Neural Information Processing Systems, pp. 75-81, MIT Press. 1996.

[24] Teeter, C. et. al. "*Generalized Leaky Integrate-and-Fire Models Classify Multiple Neuron Types*", Nature Communications, Vol. 9, Article No. 709, 2018.

[25] Tuckwell, H. C. "*Introduction to Theoretical Neurobiology: Volume 2 Non Linear and Stochastic Theories*," Cambridge University Press, 1988, [ISBN-10: 0521022223].

[26] Tuckwell, H. C. "*Frequency of firing of Stein's Model Neuron with Application to cells of the Dorsal Spinocerebellar Tract*", Brain Research, Vol. 116, p. 323-328, 1976.

[27] Stein, R. B. "*A Theoretical Analysis of Neuronal Variability*", Biophysical Journal, Vol. 5, No. 2, pp. 173-194,

1965.

[28] Stein, R. B. "*Some Models of Neuronal Variability*", Biophysical Journal, Vol. 7, pp. 37-68, 1967.

[29] Wilbur, W. J. and Rinzel, J. "*An Analysis of Stein's Model for Stochastic Neuron Excitation*", Biological Cybernetics, Vol. 45, pp. 107-114, 1982.

Author'sBiography

Mr. VDS Baghela received his graduation degree in Statistics from BHU in 1999 & completed MCA degree from AAIDU, Allahabad in 2004. He obtained M.Tech (CSE) degree from AKTU, Lucknow in 2010. He is pursuing Ph.D (CSE) under the supervision of Dr. S. K. Bharti. Mr. Baghela has served many engineering colleges of NCR as HoD & Dean. At present he is working as a Chief Operating Officer (COO) in EMVIDYA EDUCATION INDIA PVT. LTD., Delhi.

Dr. Sunil Kumar Bharti receives Master's degree (Master of Computer Application) and Ph.D from School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. His primary research area are in Modeling and Simulation, Computational (Neuroscience), image processing and computer vision. He has published various research papers (in reputed journals, international and national conferences.

Saket Kumar Choudhary obtained his master degrees in Mathematics from the University of Allahabad, Allahabad, India in 2005, Master of Computer Application (MCA) from UPTU, Lucknow, India in 2010, Master of Technology (M.Tech) from Jawaharlal Nehru University, New Delhi, India in 2014. He is Ph.D (Computer Science and Technology) School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. Currently, he is working as an assistant professor in MRIIRS, Faridabad. His research interest includes mathematical modeling and simulation, dynamical systems, computational neuroscience: modeling of single and coupled neurons, computer vision, digital image processing, machine learning and artificial intelligence.

How to Cite

Baghela, Vishwadeepak Singh, Bharti, Sunil Kumar and Choudhary , Saket Kumar (2019). Information Processing in GLIF Neuron Model with Noisy Conductance. *International Journal of Machine Learning and Networked Collaborative Engineering*, 3(02) pp 102-113.

doi: https://doi.org/10.30991/IJMLNCE.2019v03i02.004

Vol. 03, No. 2,(2019),114–124, **ISSN:2581-3242**

IJMLNCE
JOURNAL

International Journal of Machine Learning and Networked Collaborative Engineering

Journal Homepage: http://www.mlnce.net/home/index.html

DOI : https://doi.org/10.30991/IJMLNCE.2019v03i02.005

Deep Neural Network with Stacked Denoise Auto Encoder for Phishing Detection

[a]Kothandan Sumathi[*], [b]Vijayan Sujatha

[a]Research scholar, Department of Computer Applications, CMS College of Science and Commerce, Coimbatore, Tamilnadu, India

[b]Associate Professor, Department of Computer Applications, CMS College of Science and Commerce, Coimbatore, Tamilnadu, India

[a]nksmba@rediffmail.com, (https://orcid.org/ 0000-0001-8761-6905)

[b]sujathapadmakumar4@gmail.com, (https://orcid.org/0000-0002-5473-0362)

Abstract

Sensitive information such as credit card information, username, password and social security number etc, can be stolen using a fake page that imitates trusted website is called phishing. The attacker designs a similar webpage either by copying or making small manipulation to the legitimate page so that the online user cannot distinguish the legitimate and fake websites. A Deep Neural Network (DNN) was introduced to detect the phishing Uniform Resource Locator (URL). Initially, a 30-dimension feature vector was constructed based on URL-based features, Hypertext Markup Language (HTML)-based features and domain-based features. These features were processed in DNN to detect the phishing URL. However, the irrelevant, redundant and noisy features in the dataset increase the complexity of DNN classifier. So the feature selection is required for efficient phishing attack detection. But feature selection is a time-consuming process since it is an independent process. So in this paper, a feature vector is generated by DNN itself using Stacked Denoise Auto Encoder (SDAE). Moreover, the noisy data such as missing features affect the efficiency of phishing detection so the SDAE is trained to reconstruct a clean input feature vector. The initial input feature vector is corrupted by setting some feature vectors as zero. Then, the corrupted feature vector is then plotted with basic auto encoder, to a hidden representation from that the input feature vector is reconstructed. The reconstructed features are given as input to DNN which selects the most relevant features and predicts the phishing URL. Hence the sparse feature representation of SDAE increases the classification accuracy of DNN. The experiments are conducted in Ham, Phishing Corpus and Phishload datasets to prove the effectiveness of DNN-SDAE.

Keywords

Phishing

Deep Neural Network

Auto encoder

Stacked Denoise

[*] Corresponding author

Kothandan Sumathi

Email: nksmba@rediffmail.com

1. Introduction

Phishing [1] is a treacherous attempt wherein the main intention of the attacker is to capture a victim's sensitive information. Generally a classic phishing attack is performed based on using a proxy or man-in-the-middle attacks or making use of browser weakness. However, the most general method is creating a web page which is more similar to the one which is familiar to the user. Hence, phishing still causes an important security threat and a large number of internet users faces this problem. Such attacks are also causing trouble for companies that provide online services.

Various techniques such as Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes and Artificial Neural Network (ANN) [2] were utilized for phishing attack detection. Good quality of training data is obtained by using deep learning technique. So, a Deep Neural Network (DNN) [3] was introduced for phishing attack detection. Initially, a feature extractor was used to extract 30-dimension feature vector from the URLs. Then, the extracted features were processed in DNN to classify the URL as a phishing URL or legitimate URL. The dataset may consist of redundant, irrelevant and noisy features. By using such features in DNN, the classification accuracy of DNN is affected.

So in this paper, DNN with Stacked Denoise Auto Encoder (SDAE) [4, 5] is proposed to remove the irrelevant, redundant and noisy data (missing features) for phishing URL detection. The feature selection process can remove irrelevant, redundant and noisy (missing features) features by selecting the most important features. But feature selection is a time-consuming process since it is an independent process. So, the feature vector is generated by DNN through SDAE. SDAE reconstructs the features from the corrupted version of the features by using a stochastic mapping. The reconstructed features are fed into the DNN to choose the most relevant features and detect the phishing URLs. The SDAE constructs high-level features which increase the classification accuracy of DNN.

2. Literature Survey

Hybrid feature selection method [6] was proposed for phishing email detection. This method was based on the grouping of behavior-based and content-based phishing detection approaches. Based on the email header, this method mined the behavior of the attacker. By analyzing the attacker behavior, it came to know that phishing email which had tended to generated from more than one domain. It indicated abnormal activity. However, this model concentrated only on email headers.

A novel approach was proposed [7] to detect and prevent from phishing URL. It combined Webpage similarity and URL-based based detection methods. URL-based phishing detection involved the extraction of actual URL and the result generated by the approach proceeded to the next phase. The approach proceeded to the visual similarity-based detection when the URL-based detection doesn't detect phishing. The webpage similarity-based detection used a threshold value for phishing attack detection. However, the efficiency of this approach depends on the threshold value.

A dynamic evolving neural network [8] model was proposed based on reinforcement learning for the detection of an online phishing email. It detected phishing emails by combined reinforcement learning and neural network as a single framework. It could adjust itself to produce email detection system. The dynamic model accepted the concept of reinforcement learning which dynamically enhanced the performance of the dynamic model. By including additional dataset to the offline dataset, the richness of this model could be increased.

An ensemble model [9] was proposed for detection of phishing attacks with Remove-Replace Feature Selection Technique (RRFST). It reduced the number of features by randomly selecting a feature and removing that feature when the attack detection accuracy was unchanged. On other hand, features were replaced to its original feature space when attack detection accuracy was increased. The selected features were used in Random Forest (RF), C4.5 and Classification and Regression Tree (CART). Even though the ensemble method had high detection accuracy, it consumed more space to store trees.

An efficient phishing website detection model [10] was proposed based on improved Back Propagation (BP) Neural Network (BPNN) and dual feature evaluation. A grey wolf algorithm was introduced to fine tune the BPNN and dual feature to sensibly choose starting parameters. After that, a dual feature evaluation mechanism was utilized to estimate the results of improved BPNN. The phishing website recognition accuracy was improved by using the dual feature evaluation mechanism. By using more features, the accuracy of this model could be increased.

A fuzzy rough set feature selection method [11] was introduced to enhance phishing attack detection. Fuzzy Rough Set theory was used to select the most discriminative features in the dataset. The selected features were fed into RF, multipreceptron and Sequential Minimal Optimization (SMO) classifiers to find the phishing websites. However, the accuracy of this method is low.

Hybrid Ensemble Feature Selection (HEFS) method [12] was proposed for phishing detection. Initially, primary feature subsets were generated by using a Cumulative Distribution Function gradient (CDF-g) algorithm and those features were acted as input to the data perturbation ensemble method. It produced secondary feature subsets. Then, a group of features primary and secondary features were obtained by employing a function perturbation ensemble method. These features were processed by Random Forest (RF) to distinguish the phishing and legitimate websites. However, the complexity of RF is high due to the creation of more number of trees.

A Case-Based Reasoning Phishing Detection System (CBR-PDS) [13] was introduced to detect the phishing websites. It primarily based on the CBR which act as an important part of phishing detection system. This system was highly flexible and active as it can easily detected latest phishing attacks. The CBR classifier classified websites with a relatively small dataset but other classifiers required to be trained in advance before classifying the websites. Initially CBR-PDS process checks OPT of current URL and checks whether the OPT was exist or not. If the OPT was present, the proposed CBR-PDS flag it as phishing website otherwise extracted features of that URL and it was formulated a new case to be tested. Then it starts CBR process which retrieves the most similar cases. However, it was failed to implement in integrated web-based CBR-PDS system.

3. Proposed Methodology

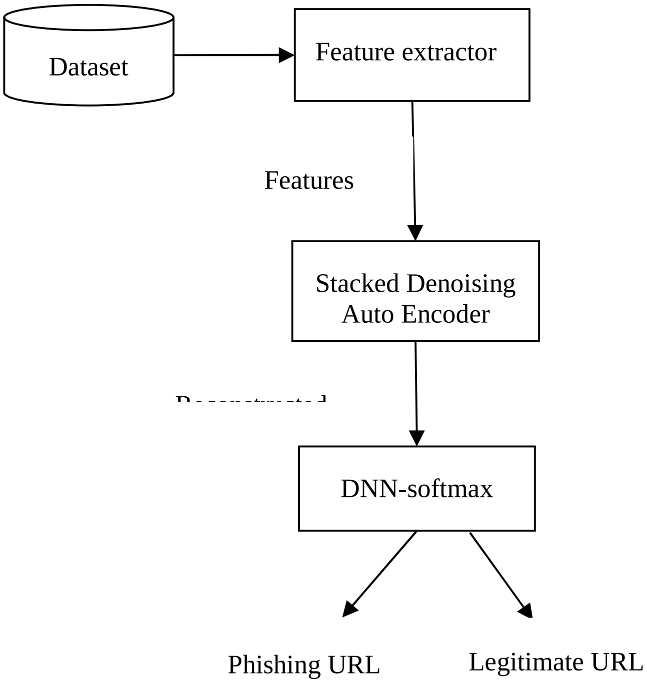


Fig.1: Workflow of Proposed Methodology

Here, the proposed Deep Neural Network with Stacked Denoising Auto Encoder (DNN-SDAE) for phishing attack detection is described in detail. Initially, a feature extractor [3] obtains URLs as an input and returns a vector that consists of thirty features. Then, the extracted features are reconstructed by SDAE to obtain high-level features for phishing detection. The reconstructed features are used in DNN which selects most significant features and predict the phishing URL. Fig. 1 shows the work flow of the proposed methodology.

3.1. Reconstruction of features using Stacked Denoise Auto Encoder

Stacked Denoising Auto Encoder (SDAE) gets the extracted features as input. SDAE has three layers are input layer, hidden layer and output layer. The hidden layer and output layer of SDAE is called as encoder layer and decoder layer respectively. SDAE consists of two encoding and two decoding layers. Initially, SDAE generates a vector $\tilde{x}$ by assigning some feature vectors as 0. Then, the SDAE uses $\tilde{x}$ as input feature vector. The number of units in the input layer is equal to the dimension of $\tilde{x}$ and the dimension of $\tilde{x}$ is denoted as d. In the encoding layer, the result of the first encoding layer acts as input to the second encoding layer. Suppose, there are L hidden layers in the encoding part, there is the activation function of the kth encoding layer is,

$$y^{(k+1)} = f_e\left(W^{(k+1)}y^{(k)} + b^{(k+1)}\right), k = 0,1, \dots , L - 1 \qquad (1)$$

In (1), the input y^0 is the original data x, $W^{(k+1)}$ is the input-to-hidden weights of the $k+1$th encoding layer, $b^{(k+1)}$ denotes the bias of the $k+1$th encoding layer and $f_e()$ is the activation function of the hidden layer. $f_e()$ is formulated as,

$$f_e\left(W^{(k+1)}y^{(k)} + b^{(k+1)}\right) = max\left(0, W^{(k+1)}y^{(k)} + b^{(k+1)}\right) \qquad (2)$$

If $W^{(k+1)}y^{(k)} + b^{(k+1)} < 0$, the output of the hidden layer will be zero. Hence it generates a sparse feature representation which may have better partition ability. The result of the last encoding layer is the high-level features extorted by SDAE. In the decoding part, the result of first decoding layer acts as input to the second decoding layer. The decoding function of the kth decode layer is given as follows:

$$z^{(k+1)} = f_d\left(W^{(L-k)T}z^{(k)} + b'(k+1)\right), k = 0,1, \dots L - 1 \qquad (3)$$

In (3), the input of the first decoding layer is represented as $z^{(0)}$ and the output of the last encoding layer is represented as $y^{(L)}$. Here, softplus function is used as decoding function $f_d()$ which is given as follows,

$$f_d(a) = \begin{cases} \log(1+e^a), & x \in [0,1] \\ a, & otherwise \end{cases} \qquad (4)$$

In (4), $a = W^{(L-k)T}z^{(k)} + b'(k+1)$. The output $z^{(L)}$ of the last decoding layer is the reconstruction of the original input feature x. The objective function (reconstruction error) is given as follows,

$$Obj = \begin{cases} -\frac{1}{k}\sum_{i=0}^{k}\sum_{j=1}^{d}\left[x_j^{(i)}\log\left(z_j^{(i)}\right) + \left(1-x_j^{(i)}\right)log\left(1-z_j^{(i)}\right)\right] + \frac{\lambda}{2}||W||^2, & x \in [0,1] \\ \frac{1}{k}\sum_{i=1}^{k}\left\|x^{(i)} - z^{(i)}\right\|^2\frac{\lambda}{2}||W||^2, & otherwise \end{cases} \qquad (5)$$

The process of SDAE while training is given as follows,

1. Get the features from the feature extractor

2. Train the initial encoding and the final decoding layer.

3. Get the $W^{(l)}, b^{(l)}$and the features $y^{(l)}$ which are the output of the initial encoding layer.

4. Exploit $y^{(k)}$as the input data of the $(k+1)$th encoding layer.

5. Train the $(k+1)$th denoise auto encoder and get $W^{(k+1)}$ and $b^{(k+1)}$ and the features $y^{(k+1)}$.

The SDAE is trained independently and the features, weight and bias values in the last decoding layer acted as the inputs of the DNN.

3.2. Deep Neural Network with Softmax

DNN with softmax is used to minimize the objective function Obj. Given an input training set with n features $\{x^{(i)}\}_{i=1}^{n}$ which are selected by DNN and the feature's label set $\{t^{(i)}\}_{i=1}^{n}$, where $t^{(i)}$ is either -1 or 1, -1 represents the legitimate URL and 1 represents the phishing URL. Softmax estimates the probability of each feature vector belonging to each class (legitimate or phishing URL). The probability is given by,

$$P\left(t^{(i)} = 2|x^{(i)};\theta\right) = \frac{1}{e^{\theta_1^T x^{(i)}} + e^{\theta_2^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \end{bmatrix} \qquad (6)$$

In (6), θ is the parameter of softmax parameter, $\frac{1}{e^{\theta_1^T x^{(i)}} + e^{\theta_2^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \end{bmatrix}$ normalizes the distribution so that the summation of the probability is one. The objective function of softmax is given as follows:

$$Obj(\theta) = -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{h} 1\{t^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{h} e^{\theta_h^T x^{(i)}} + e^{\theta_1^T x^{(i)}}}\right] \qquad (7)$$

In (7), $h = 1,2$, $1\{.\}$ is the indicator function which returns 1 if the condition is true. Otherwise, it returns 0. By minimizing the objective function, legitimate URLs are detected effectively. The softmax is used to generate the classes of the features. The DNN together with SDAE and softmax is achieved an improved classification result.

4. Result and Discussion

The efficiency of DNN and DNN-SDAE is evaluated in this section based on accuracy, precision, recall and f-measure performance metrics. The DNN and DNN-SDAE based phishing detection in MATLAB 2018a by using Ham, Phishing Corpus and Phishload datasets. The Ham dataset is utilized for a baseline evaluation which consists of 4,150 legitimate e-mail communication and 1897 spam-based e-mails. The Phishing Corpus is utilized for its complication of phishing e-mail communication which consists of 4,559 phishing e-mail messages. The Phishload dataset is utilized for its raw web-based coding structures that consist of 1,185 legitimate and 3,718 phishing URLs.

4.1. Accuracy

Accuracy is the fraction of the total number of correct phishing URL detections to the actual dataset size. It measures the overall rate of correctly detected legitimate and phishing URLs.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \qquad (8)$$

In (8), True Positive is the percentage of phishing URLs in the training dataset that is properly differentiated as phishing URLs

True Negative is the percentage of legitimate URLs in the training dataset that is properly differentiated as legitimate URLs

False Positive is the percentage of legitimate URLs that is improperly differentiated as phishing URLs

False Negative is the percentage of phishing URLs that is improperly differentiated as legitimate URLs.

Table 1 shows the comparison between DNN and DNN-SDAE in terms of accuracy for different datasets.

Table 1: Comparison of Accuracy

Datasets	**DNN**	**DNN-SDAE**
Ham	0.9	0.92
Phishing Corpus	0.92	0.94
Phishload	0.89	0.914

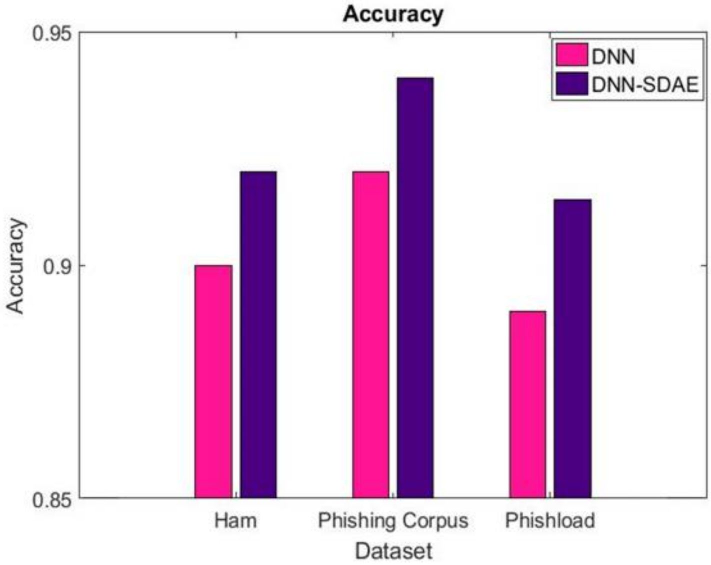


Fig.2: Comparison of Accuracy

Fig. 2 shows the comparison of accuracy between DNN and DNN-SDAE based phishing URL detection for Ham, Phishing Corpus and Phishload datasets. The datasets are taken in X-axis and the accuracy is taken in Y-axis. For Ham dataset, the accuracy of DNN-SDAE based phishing URL detection is 2.22% greater than DNN based phishing URL detection. Similarly, for Phishing Corpus dataset, the accuracy of DNN-SDAE based phishing URL detection is 2.17% greater than DNN based phishing URL detection. The accuracy of DNN-SDAE based phishing URL detection is 2.7% greater than DNN based phishing URL detection for Phishload dataset. From this analysis, it is proved that the proposed DNN-SDAE based phishing URL detection has high accuracy than DNN based phishing URL detection.

4.2. Precision

Precision measures the exactness of the classifier, i.e., what percentage of URLs that the classifier labeled as phishing URLs and it is given by,

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad (9)$$

Table 2 shows the comparison between DNN and DNN-SDAE in terms of precision for different datasets.

Table 2: Comparison of Precision

Datasets	DNN	DNN-SDAE
Ham	0.88	0.91
Phishing Corpus	0.89	0.916
Phishload	0.87	0.90

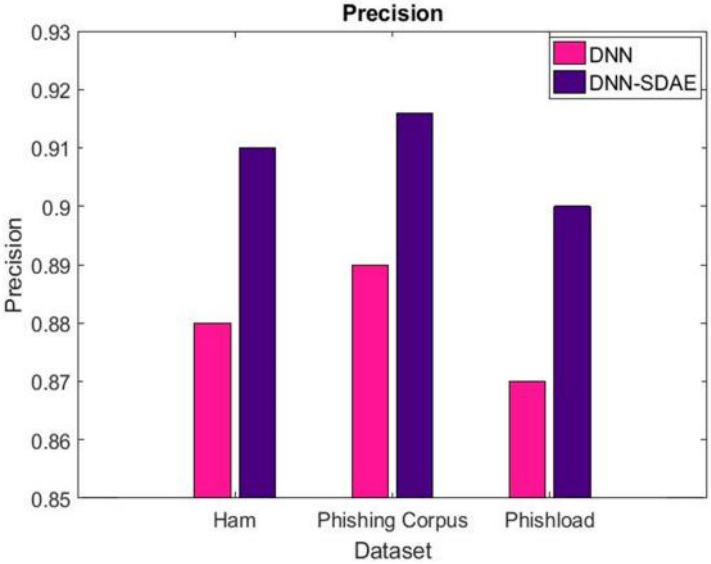


Fig.3: Comparison of Precision

Fig. 3 shows the comparison of precision between DNN and DNN-SDAE based phishing URL detection for Ham, Phishing Corpus and Phishload datasets. The datasets are taken in X-axis and the precision is taken in Y-axis. For Ham dataset, the precision of DNN-SDAE based phishing URL detection is 3.41% greater than DNN based phishing URL detection. Similarly, for Phishing Corpus dataset, the precision of DNN-SDAE based phishing URL detection is 2.92% greater than DNN based phishing URL detection. The precision of DNN-SDAE based phishing URL detection is 3.45% greater than DNN based phishing URL detection for Phishload dataset. From this analysis, it is proved that the proposed DNN-SDAE based phishing URL detection has high precision than DNN based phishing URL detection.

4.3. Recall

Recall measures the completeness of the classifier results, i.e., what percentage of phishing URLs did the classifier label as phishing, and is given by,

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \qquad (10)$$

Table 3 shows the comparison between DNN and DNN-SDAE in terms of recall for different datasets.

Table 3: Comparison of Recall

Datasets	**DNN**	**DNN-SDAE**
Ham	0.87	0.9
Phishing Corpus	0.88	0.915
Phishload	0.88	0.92

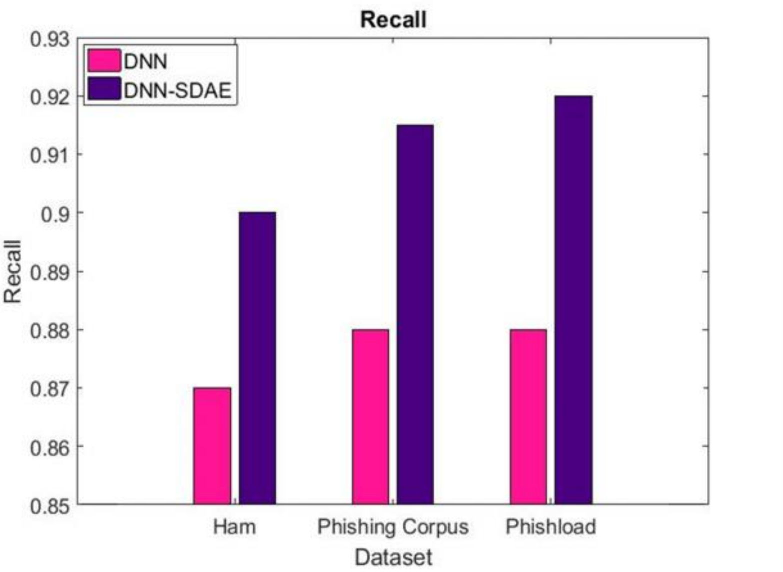


Fig.4: Comparison of Recall

Fig. 4 shows the comparison of recall between DNN and DNN-SDAE based phishing URL detection for Ham, Phishing Corpus and Phishload datasets. The datasets are taken in X-axis and the recall is taken in Y-axis. For Ham dataset, the recall of DNN-SDAE based phishing URL detection is 3.45% greater than DNN based phishing URL detection. Similarly, for Phishing Corpus dataset, the recall of DNN-SDAE based phishing URL detection is 3.98% greater than DNN based phishing URL detection. The recall of DNN-SDAE based phishing URL detection is 4.55% greater than DNN based phishing URL detection for Phishload dataset. From this analysis, it is proved that the proposed DNN-SDAE based phishing URL detection has high recall than DNN based phishing URL detection.

4.4. F-measure

F-measure is computed as the mean of precision and recall. It is calculated as,

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

Table 4 shows the comparison between DNN and DNN-SDAE in terms of f-measure for different datasets.

Table 4: Comparison of F-measure

Datasets	DNN	DNN-SDAE
Ham	0.87	0.905
Phishing Corpus	0.885	0.915
Phishload	0.878	0.904

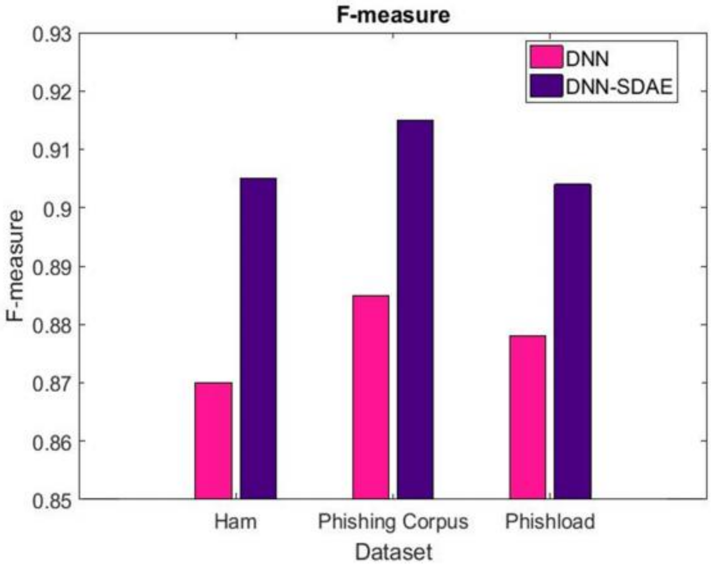


Fig.5: Comparison of F-measure

Fig. 5 shows f-measure value of DNN and DNN-SDAE based phishing URL detection for Ham, Phishing Corpus and Phishload datasets. The datasets are taken in X-axis and the f-measure is taken in Y-axis. For Ham dataset, the f-measure of DNN-SDAE based phishing URL detection is 4.02% greater than DNN based phishing URL detection. Similarly, for Phishing Corpus dataset, the f-measure of DNN-SDAE based phishing URL detection is 3.39% greater than DNN based phishing URL detection. The f-measure of DNN-SDAE based phishing URL detection is 2.96% greater than DNN based phishing URL detection for Phishload dataset. From this analysis, it is proved that the proposed DNN-SDAE based phishing URL detection has high f-measure than DNN based phishing URL detection.

5. Conclusion

In this paper, DNN with SDAE is introduced for efficient phishing URL detection. A feature extractor extracts URL-based features, web-based features and domain-based features. The extracted features are reconstructed by SDAE which returns high-level features. These features are acted as input to DNN which selects the most important features and classifies the URLs as legitimate or phishing URLs by using softmax classifier. The SDAE returns sparse features which increase the classification accuracy of DNN. The experimental results prove that the proposed DNN-SDAE has high accuracy, precision, recall and f-measure for Ham, Phishing Corpus and Phishload datasets.

References

[1]. Basnet, R.B.; Mukkamala, S.; Sung, A.H. “Detection of Phishing Attacks: A Machine Learning Approach”. *Soft Comput. Appl. Ind.* **2008,** *226*, 373-383. doi: 10.1007/978-3-540-77465-5_19.

[2]. Chin, T.; Xiong, K.; Hu, C. "Phishlimiter: A Phishing Detection and Mitigation Approach Using Software-Defined Networking". *IEEE Access*, **2018,** *6*, 42516-42531. doi: 10.1109/ACCESS.2018.2837889.

[3]. Sumathi, K.; Sujatha, V. "Deep learning based phishing attack detection". **2019**. *In Progress.*

[4]. Du, B.; Xiong, W.; Wu, J.; Zhang, L.; Zhang, L.; Tao, D.; "Stacked convolutional denoising auto-encoders for feature representation". *IEEE trans. cybern.* **2016**, *47*, 1017-1027. doi: 10.1109/TCYB.2016.2536638.

[5]. Masci, J.; Meier, U.; Cireşan, D.; Schmidhuber, J.; "Stacked convolutional auto-encoders for hierarchical feature extraction". *Int. Conf. Artif. Neural Netw.* Springer, Berl., Heidelb., **2011**, 52-59. doi: 10.1007/978-3-642-21735-7_7.

[6]. Hamid, I. R. A.; Abawajy, J. "Hybrid feature selection for phishing email detection. *Intern. Conf. Algorithms Archit. Parallel Process"., Springer, Berl., Heidelb.* **2011,** 266-275. doi: 10.1007/978-3-642-24669-2_26.

[7]. Shekokar, N. M.; Shah, C.; Mahajan, M.; Rachh, S. "An ideal approach for detection and prevention of phishing attacks". *Procedia Comput. Sci.* **2015,** *49,* 82-91. doi: 10.1016/j.procs.2015.04.230.

[8]. Smadi, S.; Aslam, N.; Zhang, L. "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning". *Decis. Support Syst.* **2018,** *107,* 88-102. doi: 10.1016/j.dss.2018.01.001.

[9]. Hota, H.S.; Shrivas, A.K.; Hota, R. "An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique". *Procedia Comput. Sci.* **2018,** *132,* 900-907. doi: 10.1016/j.procs.2018.05.103.

[10]. Zhu, E.; Liu, D.; Ye, C.; Liu, F.; Li, X.; Sun, H. "Effective Phishing Website Detection Based on Improved BP Neural Network and Dual Feature Evaluation". *IEEE Intl Conf Parallel & Distrib. Process. Appl., Ubiquitous Comput. Commun., Big Data & Cloud Computing, Social Computing & Networking, Sustainable Comput. Commun.* **2018,** 759-765. doi: 10.1109/BDCloud.2018.00114.

[11]. Zabihimayvan, M.; Doran, D. "Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection". *arXiv preprint arXiv:1903.05675* **2019**. doi: 10.1109/BDCloud.2018.00114.

[12]. Chiew, K.L.; Tan, C.L.; Wong, K.; Yong, K.S.; Tiong, W. K. "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system". *Inf. Sci.***2019,** *484*, 153-166. doi: 10.1016/j.ins.2019.01.064.

[13]. Abutair, H. Y.; Belghith, A. "Using Case-Based Reasoning for Phishing Detection". *Procedia Comput. Sci.*, **2017,** *109*, 281-288. doi: 10.1016/j.ins.2019.01.064.

Author's Biography

Kothandan Sumathi, Ph.d Research scholar, CMS College of Computer Applications, CMS College of Science and Commerce, Chinnavedampatty, Coimbatore, Tamilnadu, India.

Kothandan Sumathi is a student of CMS college of science and commerce, affiliated to Bharathiar university, Coimbatore, Tamilnadu, India. She is pursing Ph.D in Computer Science. She is doing research in the area of information security.

Vijayan Sujatha, Dean-Administration, CMS College of Computer Applications, CMS College of Science and Commerce, Chinnavedampatty, Coimbatore, Tamilnadu, India.

Vijayan Sujatha has 16 years of teaching experience and 2 years of IT Industrial experience. Her area of specialization is web mining, IoT and Big Data Analysis. She has published 24 research articles in National and International Journals and also presented papers in several National Conferences, Seminars and Workshops. She is currently guiding M.Phil and Ph.D Scholars. She has an ideal knowledge in programming languages, DOT NET frameworks and has developed two live projects using Visual programming. She also sets question papers for universities in TamilNadu.

How to Cite

Kothandan Sumathi and V. Sujatha (2019). Deep Neural Network with Stacked Denoise Auto Encoder for Phishing Detection. *International Journal of Machine Learning and Networked Collaborative Engineering*, 3(02) pp 114-124.

doi : https://doi.org/10.30991/IJMLNCE.2019v03i02.005.