

CHATBOT Development Using Python

¹Shweta Singh

¹Department of Computer Application Jain University, Bangalore, India

ABSTRACT

In this article, we center on chatbot utilizing python and convey the chatbot highlight to sites. conversation gatherings, online journals, wikis and so forth. Over recent years, informing applications have gotten more well-known than Social systems administration locales. Individuals are utilizing informing applications nowadays, for example, Facebook Messenger, Skype, Viber, Telegram, Slack and so forth. This is making different organizations accessible on informing stages prompts proactive collaboration with clients about their items. To collaborate on such informing stages with numerous clients, the organizations can compose a PC program that can talk like a human which is known as a chatbot.

1. INTRODUCTION

Advanced technology has become the integral part of our life [1]. To satisfy the need of the society, almost in each work, we use the technology [2] [3]. In current era computer science is major subject [4]. It has many real life applications such as cloud computing [5], artificial intelligence [6], remote monitoring [7], Wireless sensor network [8, 9, 10], internet of things [11, 12, 13], Neural network [14, 15], FSPP [16, 17, 18], NSPP [19, 20, 21, 22, 23], TP [24, 25, 26], internet Security [27], uncertainty [28, 29, 30, 31, 32] and so on. Technology is the mode by which user can store, fetch, communicate and utilize the information [33]. So, all the organizations, industries and also every individual are using computer systems to preserve and share the information [34]. The internet security plays a major role in all computer related applications. The internet security appears in many real-life applications, e.g., home security, banking system, education sector, defense system, Railway, and so on. In this manuscript we discuss about the protection of authentication which is a part of internet security.

A chatbot is a conversational agent where a computer program is designed to simulate an intelligent conversation. It can take user input in many formats like text, voice, sentiments, etc. For this purpose, many open source platforms are available. Artificial Intelligence Markup Language (AIML) is derived from Extensible Markup Language (XML) which is used to build up a conversational agent (chatbot) artificially. In this project, we are using Python, XML and AIML interpreter for the generation of the responses of users input. We have used this method for developing a chatbot which will interact with user using text and voice responses. Chatbots come in two kinds:

- Limited set of rules
- Machine learning

This sort of bots is extremely constrained to set of writings or orders. They have capacity to react just to those writings or orders. In the event that client asks something other than what's expected or other than the arrangement of writings or orders which are characterized to the bot, it would not react as wanted since it doesn't comprehend or it has not prepared what client inquired. These bots are not exceptionally savvy when contrasted with other sort of bots.

2. LITERATURE REVIEW

2.1 Natural Language Processing

Natural Language Processing (NLP) is the study of letting computers understand human languages. Without NLP, human language sentences are just a series of meaningless symbols to computers. Computers don't recognize the words and don't understand the grammars. NLP can be regarded as a "translator", who will translate human languages to computer understandable information.

Traditionally, users need to follow well-defined procedures accurately, in order to interact with computers. For example, in Linux systems, all commands must be precise. A single replace of one character or even a space can have significant difference. However, the emergence of NLP is changing the way of interacting. Apple Siri and Microsoft Cortana have made it possible to give command in everyday languages and are changing the way of interacting.

2.2 Machine Learning

Machine Learning (ML) is an area of computer science that "gives computers the ability to learn without being explicitly programmed". The parameter of the formulas is calculated from the data, rather than defined by the programmer. Two most common usage of ML is Classification and Regression. As shown in figure1[8], Classification means to categorize different types of data, while Regression means to find a way to describe the data. Basic ML program will have two stages, fitting and predicting. In the fitting stage, the program will be given a large set (at least thousands) of data. The program will try to adjust its parameter based on some statistical models, in order to make it "fit" the input data best. In the predicting stage, the program will give a prediction for a new input based on the parameters it just calculated out. For example, the famous Iris flower dataset [9] contains the measurement of several features of three different species of flowers, such as the length of sepals and petals. A well-defined ML program can learn the pattern behind this feature and give prediction accordingly.

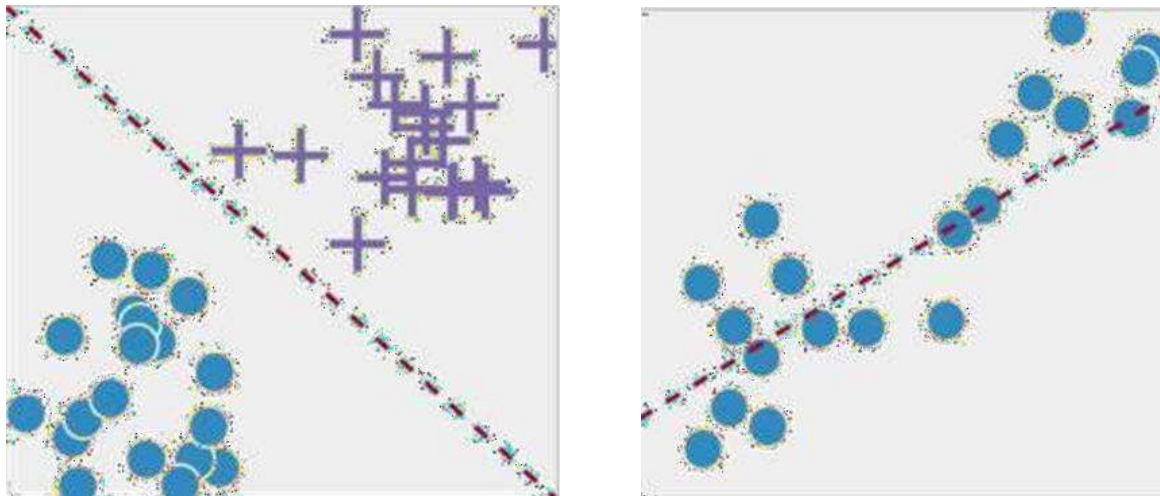


Fig.1 Data description in ML environment

3. PROBLEM IDENTIFICATION

Changing of prerequisites as a mechanical venture to assemble an item, we should follow the necessity from the client. In any case, in light of the fact that the task will probably be utilized by the business group, yet it is mindful by the specialized group, the necessity changed a ton in the center after a gathering with the business group. The business group need a straightforward bot that can give proposal right away. We needed to document what we had done previously and fabricate another one. Lacking of preparing information. The amount and nature of preparing information is basic to the exhibition to an AI model. Be that as it may, in light of the fact that some classified and protection reasons, the business group can't give enough information to us, and we needed to make up information by our own. For the AI model, we create some phony information dependent on our day by day educational experience, which is truly one-sided, in spite of the fact that with a decent precision on the phony dat. Temperamental API adaptation. Since API administration we are utilizing are still a work in progress, and we can't fix to a variant for the API, the API may change additional time. In addition, there are irregularities between the APIs and their records or test codes. Curious about the Python and AIML language. I had no past information with Python and AIML language. Programming in another dialect in such a gigantic system is very trying for me toward the start of the undertaking. Nonetheless, when I went to the later stages, I am increasingly used to that.

4. PROPOSED SOLUTION

4.1 AIML Scripting

I created the AIML files that only handle one particular pattern such as hello. aiml, inventions. aiml, geography. aiml and many more to make the bot realistic and answer some of the general inquiries. When we enter that command to the bot, it will try to load those. aiml. It won't work unless we actually create it. Here is what you can put inside hello. aiml. We will match that pattern and respond.

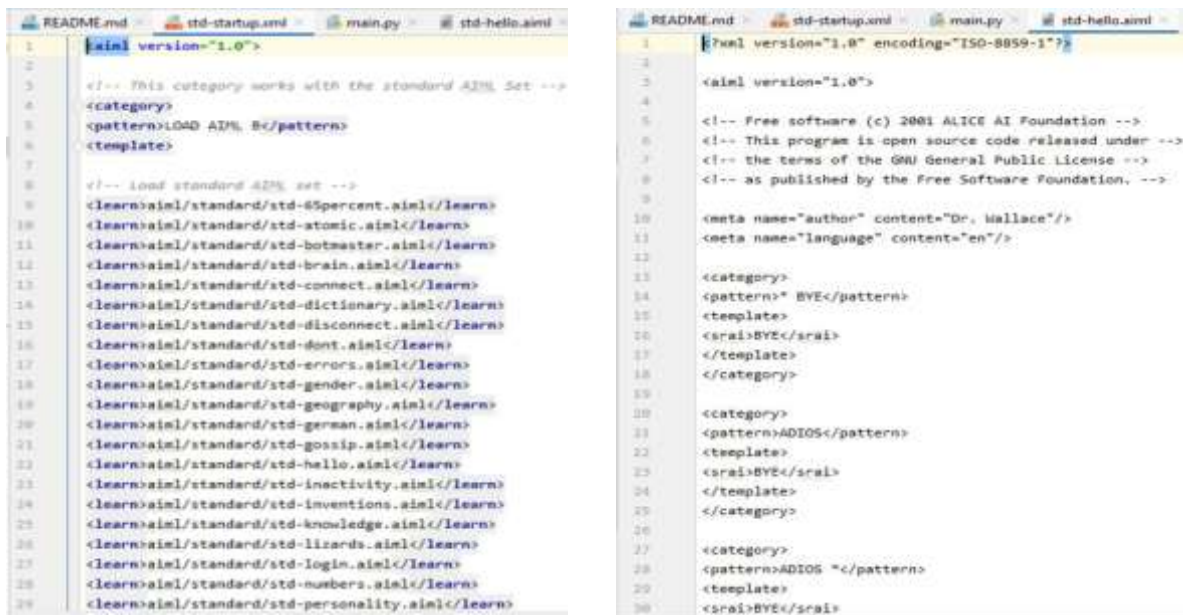


Fig 2 Loading Aiml

4.2 Creating a Startup File

It is standard to make a startup document called sexually transmitted disease startup.xml as the principle section point for stacking AIML records. For this situation we will make an essential document that matches one example and makes one move. We need to coordinate the example load aiml b, and have it load our aiml cerebrum accordingly. Making Interface As appeared in the figure, there are six parts in our task. The interfaces are the front-end visit box for client to converse with the bot, which can be the Bot Portal, Skype, Facebook, and so forth. The connector fills in as a typical entryway for all the interfaces. The outbound side calls diverse APIs to various front end; however, the inbound APIs saved the equivalent for our bot to associate. Luckily, this connector has just been executed by the jar structure, we just need to appropriately arrange them. The bot part contains the fundamental stream control of our undertaking. It is answerable for divert the contribution to various models, parse the arrival esteems, and figures out what to do straightaway. It is likewise associated with the database to recover and refresh values.

4.3 Speeding up Brain Load

At the point when you begin to have a great deal of AIML records, it can set aside a long effort to learn. This is the place cerebrum documents come in. After the bot learns all the AIML records it can spare its mind straight forwardly to a document which will radically accelerate load times on resulting runs.

4.4 Loading Brain

This is the least complex program we can begin with. It makes the aiml object, learns the startup document, and afterward stacks the remainder of the aiml records. From that point forward, it is prepared to talk, and we enter a limitless circle that will keep on provoking the client for a message. You should enter an example the bot perceives. The examples perceived rely upon what AIML records you stacked. We make the startup record as a different element so we can add more aiml documents to the bot later without changing any of the programs source code. We can simply add more documents to learn in the startup xml record.

```

28 <learn>aiml/standard/std-numbers.aiml</learn>
29 <learn>aiml/standard/std-personality.aiml</learn>
30 <learn>aiml/standard/std-pickup.aiml</learn>
31 <learn>aiml/standard/std-politics.aiml</learn>
32 <learn>aiml/standard/std-profile.aiml</learn>
33 <learn>aiml/standard/std-religion.aiml</learn>
34 <learn>aiml/standard/std-robot.aiml</learn>
35 <learn>aiml/standard/std-sales.aiml</learn>
36 <learn>aiml/standard/std-sextalk.aiml</learn>
37 <learn>aiml/standard/std-sports.aiml</learn>
38 <learn>aiml/standard/std-srai.aiml</learn>
39 <learn>aiml/standard/std-suffixes.aiml</learn>
40 <learn>aiml/standard/std-that.aiml</learn>
41 <learn>aiml/standard/std-turing.aiml</learn>
42 <learn>aiml/standard/std-yesno.aiml</learn>
43
44
45 <!-- Load additional AIML set -->
46 <learn>aiml/additional/computer.aiml</learn>
47
48 </template>
49 </category>
50
51 </aiml>
    
```

```

1 <!DOCTYPE html>
2 <html>
3   <head lang="en">
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Chatbot</title>
7     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
8     <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
9     <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">
10    <style type="text/css">
11      .fixed-panel {
12        min-height: 400px;
13        max-height: 400px;
14        background-color: #191919;
15        color: white;
16        overflow: auto;
17      }
18      .media-list {
19        overflow: auto;
20        clear: both;
21        display: table;
22        overflow-wrap: break-word;
23        word-wrap: break-word;
24        word-break: normal;
25        line-break: strict;
26      }
27      .panel {
28        margin-bottom: 20px;
29        background-color: #fff;
    
```

```

C:\Users\Projects\chatbot-using-aiml\venv\Scripts\python.exe "C:\Users\Projects\chatbot-using-aiml\venv\Scripts\python.exe"
* Restarting with stat
* Debugger is active!
* Debugger PID: 285-555-837
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [24/Mar/2020 19:37:55] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Mar/2020 19:37:56] "GET /favicon.ico HTTP/1.1" 404 -
Loading brain from bot_brain.brn... done (23482 categories in 0.86 seconds)
Kernel bootstrap completed in 0.86 seconds
127.0.0.1 - - [24/Mar/2020 19:38:05] "POST /ask HTTP/1.1" 200 -
Loading brain from bot_brain.brn... done (23482 categories in 0.84 seconds)
Kernel bootstrap completed in 0.84 seconds
127.0.0.1 - - [24/Mar/2020 19:38:11] "POST /ask HTTP/1.1" 200 -
Loading brain from bot_brain.brn... done (23482 categories in 0.86 seconds)
Kernel bootstrap completed in 0.86 seconds
127.0.0.1 - - [24/Mar/2020 19:38:20] "POST /ask HTTP/1.1" 200 -
Loading brain from bot_brain.brn... done (23482 categories in 0.88 seconds)
Kernel bootstrap completed in 0.88 seconds
127.0.0.1 - - [24/Mar/2020 19:38:28] "POST /ask HTTP/1.1" 200 -
Loading brain from bot_brain.brn... done (23482 categories in 0.86 seconds)
Kernel bootstrap completed in 0.86 seconds
127.0.0.1 - - [24/Mar/2020 19:38:39] "POST /ask HTTP/1.1" 200 -
Loading brain from bot_brain.brn... done (23482 categories in 0.84 seconds)
Kernel bootstrap completed in 0.84 seconds
127.0.0.1 - - [24/Mar/2020 19:38:52] "POST /ask HTTP/1.1" 200 -
127.0.0.1 - - [24/Mar/2020 19:39:00] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Mar/2020 19:41:48] "GET / HTTP/1.1" 200 -
    
```

```

1 from flask import Flask, render_template, request, jsonify
2 import aiml
3 import os
4
5 app = Flask(__name__)
6
7 @app.route("/")
8 def hello():
9     return render_template("chat.html")
10
11 @app.route("/ask", methods=['POST'])
12 def ask():
13     message = request.form['messageText'].encode('utf-8').strip()
14
15     kernel = aiml.Kernel()
16
17     if os.path.isfile("bot_brain.brn"):
18         kernel.bootstrap(brainFile = "bot_brain.brn")
19     else:
20         kernel.bootstrap(learnFiles = os.path.abspath("aiml/std-startup;
21         kernel.saveBrain("bot_brain.brn")
22
23     # kernel now ready for use
24     while True:
25         if message == "quit":
26             exit()
27         elif message == "save":
28             kernel.saveBrain("bot_brain.brn")
29         else:
    
```

Fig 3 Speeding up Brain, Creating Interface and Execution of main file

5. IMPLEMENTATION

This section covers the design and implementation of different module of the bot, which contains the design of the PYTHON module, and the AIML module. Chatbot Interface

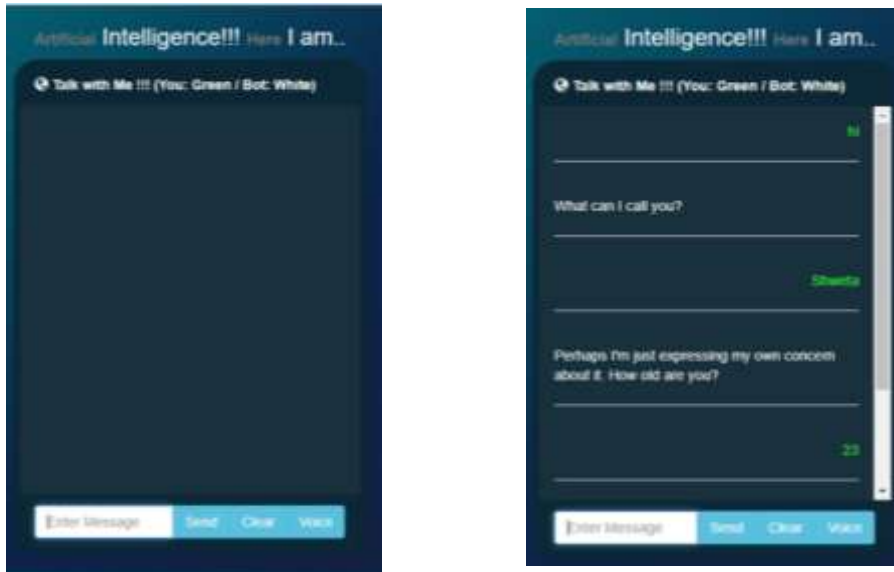


Fig 4 Chatbot Interface and Chatbot Conversation

6. CONCLUSION

Chatbots are the new Apps! As we have talked about in the above expectations, this venture brings the influence of chatbots to sites and advances its convenience. Chatbots in sites can give a human like touch to certain viewpoints and make it a getting a charge out of discussion. What's more, they are centered completely on giving data and finishing assignments to the people they associate with. The previously mentioned usefulness in all the expectations is executed and pushed in to Heroku server. By actualizing the previously mentioned expectations, I had the option to include fundamental chatbot usefulness in to the Website.

Following stage towards building chatbots include helping individuals to encourage their work and connect with PCs utilizing regular language or utilizing Machine Learning. Eventual fate of chatbots, upheld by AI innovation, will have the option to recall past discussions and gain from them to answer new ones. The test would speak with numerous bot clients and different client

7. FUTURE SCOPE

There are limitations to what has been currently achieved with chatbots. The limitations of data processing and retrieval are hindering chatbots to reach their full potential. It is not that we lack the computational processing power to do so. However, there is a limitation on "How" we do it. One of the biggest examples is the retail customer market. Retail customers are primarily interested in interacting with humans because of nature of their needs. They don't want bots to process their needs and respond accordingly.

8. ACKNOWLEDGEMENT

I would like to express my profound gratitude to professor Dr. MN Nachappa and project coordinators for their patient, encouragement and valuable assessments of this research work.

9. REFERENCES

- [1] M. BM and H. Mohapatra, "Human centric software engineering," International Journal of Innovations & Advancement in Computer Science (IJIACS), vol. 4, no. 7, pp. 86-95, 2015.
- [2] H. Mohapatra, C Programming: Practice, Vols. ISBN: 1726820874, 9781726820875, Kindle, 2018.
- [3] H. Mohapatra and A. Rath, Advancing generation Z employability through new forms of learning: quality assurance and recognition of alternative credentials, ResearchGate, 2020.
- [4] H. Mohapatra and A. Rath, Fundamentals of software engineering: Designed to provide an insight into the software engineering concepts, BPB, 2020.
- [5] V. Ande and H. Mohapatra, "SSO mechanism in distributed environment," International Journal of Innovations & Advancement in Computer Science, vol. 4, no. 6, pp. 133-136, 2015.
- [6] H. Mohapatra, "Ground level survey on sambalpur in the perspective of smart water," EasyChair, vol. 1918.
- [7] H. Mohapatra, S. Panda, A. Rath, S. Edalatpanah and R. Kumar, "A tutorial on powershell pipeline and its

- loopholes," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 4, 2020.
- [8] H. Mohapatra and A. Rath, "Fault tolerance in WSN through PE-LEACH protocol," *IET Wireless Sensor Systems*, vol. 9, no. 6, pp. 358-365, 2019.
- [9] H. Mohapatra, S. Debnath and A. Rath, "Energy management in wireless sensor network through EB-LEACH," *International Journal of Research and Analytical Reviews (IJRAR)*, pp. 56-61, 2019.
- [10] H. Mohapatra and A. Rath, "Fault-tolerant mechanism for wireless sensor network," *IET Wireless Sensor Systems*, vol. 10, no. 1, pp. 23-30, 2020.
- [11] H. Mohapatra and A. Rath, "Detection and avoidance of water loss through municipality taps in india by using smart tap and ict," *IET Wireless Sensor Systems*, vol. 9, no. 6, pp. 447-457, 2019.
- [12] M. Panda, P. Pradhan, H. Mohapatra and N. Barpanda, "Fault tolerant routing in heterogeneous environment," *International Journal of Scientific & Technology Research*, vol. 8, pp. 1009-1013, 2019.
- [13] D. Swain, G. Ramkrishna, H. Mahapatra, P. Patra and P. Dhandrao, "A novel sorting technique to sort elements in ascending order," *International Journal of Engineering and Advanced Technology*, vol. 3.
- [14] H. Mohapatra, "HCR using neural network," 2009.
- [15] V. Nirgude, H. Mahapatra and S. Shivarkar, "Face recognition system using principal component analysis & linear discriminant analysis method simultaneously with 3d morphable model and neural network BPNN method," *Global Journal of Advanced Engineering Technologies and Sciences*, vol. 4, p. 1, 2017.
- [16] R. Kumar, S. Edalatpanah, S. Jha, S. Gayen and R. Singh, "Shortest path problems using fuzzy weighted arc length," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, pp. 724-731.
- [17] R. Kumar, S. Jha and R. Singh, "A different approach for solving the shortest path problem under mixed fuzzy environment," *International Journal of fuzzy system Applications*, vol. 9, no. 2, pp. 132-161, 2020.
- [18] R. Kumar, S. Jha and R. Singh, "Shortest path problem in network with type-2 triangular fuzzy arc length," *Journal of Applied Research on Industrial Engineering*, vol. 4, pp. 1-7, 2017.
- [19] S. Broumi, A. Dey, M. Talea, A. Bakali, F. Smarandache, D. Nagarajan, M. Lathamaheswari and R. Kumar, "Shortest path problem using Bellman algorithm under neutrosophic environment," *Complex & Intelligent Systems*, vol. 5, pp. 409--416, 2019.
- [20] R. Kumar, S. Edalatpanah, S. Jha, S. Broumi, R. Singh and A. Dey, "A multi objective programming approach to solve integer valued neutrosophic shortest path problems," *Neutrosophic Sets and Systems*, vol. 24, pp. 134-149, 2019.
- [21] R. Kumar, A. Dey, F. Smarandache and S. Broumi, "A study of neutrosophic shortest path problem," in *Neutrosophic Graph Theory and Algorithms*, F. Smarandache and S. Broumi, Eds., IGI-Global, 2019.
- [22] R. Kumar, S. Edalatpanah, S. Jha and R. Singh, "A novel approach to solve gaussian valued neutrosophic shortest path problems," *International Journal of Engineering and Advanced Technology*, vol. 8.
- [23] R. Kumar, S. Edalatpanah, S. Jha, S. Broumi and A. Dey, "Neutrosophic shortest path problem," *Neutrosophic Sets and Systems*, vol. 23, pp. 5-15, 2018.
- [24] R. Kumar, S. Edalatpanah, S. Jha and R. Singh, "A Pythagorean fuzzy approach to the transportation problem," *Complex and Intelligent System*, vol. 5, pp. 255-263, 2019.
- [25] J. Pratihari, R. Kumar, A. Dey and S. Broumi, "Transportation problem in neutrosophic environment," in *Neutrosophic Graph Theory and Algorithms*, F. Smarandache and S. Broumi, Eds., IGI-Global, 2019.
- [26] J. Pratihari, S. E. R. Kumar and A. Dey, "Modified Vogel's Approximation Method algorithm for transportation problem under uncertain environment," *Complex & Intelligent Systems (Communicated)*.
- [27] J. Sakhnini, H. Karimipour, A. Dehghantanha, R. Parizi and G. Srivastava, "Security aspects of Internet of Things aided smart grids: A bibliometric survey," *Internet of Things*, pp. 100-111, 2019.
- [28] S. Gayen, F. Smarandache, S. Jha and R. Kumar, "Interval-valued neutrosophic subgroup based on interval-valued triple t-norm," in *Neutrosophic Sets in Decision Analysis and Operations Research*, M. Abdel-Basset and F. Smarandache, Eds., IGI-Global, 2019, p. 300.
- [29] S. Gayen, F. Smarandache, S. Jha, M. Singh, S. Broumi and R. Kumar, "Introduction to plithogenic subgroup," in *Neutrosophic Graph Theory and Algorithm*, F. Smarandache and S. Broumi, Eds., IGI-Global.
- [30] S. Gayen, S. Jha, M. Singh and R. Kumar, "On a generalized notion of anti-fuzzy subgroup and some characterizations," *International Journal of Engineering and Advanced Technology*, vol. 8.
- [31] S. Gayen, F. Smarandache, S. Jha, M. K. Singh, S. Broumi and R. Kumar, "Introduction to plithogenic hypersoft subgroup," *Neutrosophic Sets and Systems*, vol. 33, p. Accepted, 2020.
- [32] S. Gayen, S. Jha and M. Singh, "On direct product of a fuzzy subgroup with an anti-fuzzy subgroup," *International Journal of Recent Technology and Engineering*, vol. 8, pp. 1105-1111, 2019.
- [33] Behura and H. Mohapatra, "IoT Based Smart City with Vehicular Safety Monitoring," *EasyChair*, vol. 1535.
- H. Panda, H. Mohapatra and A. Rath, "WSN-Based Water Channelization: An Approach of Smart Water," *Smart Cities—Opportunities and Challenges. Lecture Notes in Civil Engineering*, vol. 58, pp. 157-166, 2020.