EVALUATING MACHINE LEARNING TECHNIQUES FOR DETECTING OFFENSIVE AND HATE SPEECH IN CHATBOT

Nazneen Fatima[1], Maseera Eram Rafai[2], Dr. Mohammed Jameel Hashmi[3]
[1,2]B. E Student, Department of CSE, ISL College of Engineering, India.
[3]Associate Professor & Head, Department of CSE, ISL College of Engineering, Hyderabad, India.

Abstract: In recent times, it has been witnessing insurgence of offensive and hate speech along with racial and ethnic dispositions in chatbots. Popular among them used is English. Although machine learning has been successfully used to detect offensive and hate speech in several English contexts, the distinctiveness of chatbots and the similarities among offensive, hate, and free speeches require domain-specific English corpus and techniques to detect offensive and hate speech. Thus, we developed an English corpus from chatbots and evaluated different machine-learning techniques to detect offensive and hate speech. Character n-gram, word n-gram, negative sentiment, syntactic-based features, and their hybrid were extracted and analyzed using hyper-parameter optimization, ensemble and multi-tier meta-learning models of support vector machine, logistic regression, random forest,and gradient boosting algorithms. The results showed that an optimized support vector machine with character n-gram performed best in the detection of hate speech, while optimized gradient boosting with word n-gram performed best in the detection of hate speech.

Index Terms— Machine learning, chatbot, hate speech, offensive speech.

1. INTRODUCTION

Among the most consequential inventions of the modern era are online social networks. Twitter is a well-liked social media site that lets users share content online with the use of emojis, photos, hyperlinks, special characters, and alphanumeric letters. Its features have evolved significantly throughout the years. For instance, in an effort to promote greater adaptability in engagement, the maximum character capacity for a tweet was recently raised from 140 to 280. Despite Twitter's terms of service explicitly forbidding hate speech and objectionable content, its dissemination has increased due to the widespread availability of free speech online. There is no universally accepted definition of hate speech; rather, it depends on incitement, defined as an outspoken and intentional call to hatred, bigotry, or violence, according to the UN strategy and plan of action against hate speech. The use of abusive slurs or insulting phrases in text has also been defined as offensive speech, which is sometimes mistaken for hate speech.

1.1. SCOPE

Machine learning has proven effective in identifying offensive and hate speech in various English contexts. However, due to the unique characteristics of Chatbot and the overlapping nature of offensive, hate, and free speech, it is necessary to utilize domain-specific English corpus and techniques to accurately detect offensive and hate

speech. Consequently, we created an English corpus sourced from Chatbot and assessed several machine-learning methods for identifying offensive and hate speech.

1.3 PURPOSE OF PROJECT

Twitter ranks as the third most popular social network in South Africa, with its members representing twenty percent of the total active social media users in the country. Recent media investigations have shown that racially inflammatory remarks were extensively disseminated over social media before to the 2019 elections. The report indicated a 170% surge in the number of nasty posts throughout the specified time frame. A recent research on hate speech across many domains has also confirmed the increase in provocative and hateful tweets, despite the existence of legal remedies. Therefore, there is a need for an extra solution to address the problem. We have devised a technique that is capable of classifying inflammatory and hate speech from a particular tweet.

1.4 KEYWORDS

Machine learning, South Africa, Twitter, hate speech, offensive speech.

2. OBJECTIVE OF THE PROJECT

A. Existing System:

Previous studies on hate speech identification have not specifically addressed the detection of offensive and hate speech in Chatbot systems. Nevertheless, several studies have used various characteristics and machine learning methods to identify ternary English tweets. Wazeem and Hovy meticulously analyzed 16,914 tweets pertaining to the Australian TV program 'My Kitchen Rules'. Out of these, 3,383 comments were identified as sexist, 1,972 tweets were identified as racist, and 11,559 tweets did not fall into either category. The evaluation included features such as character n-grams, which included unigram, bigram, trigram, and four-gram, along with gender, location, and tweet length (excluding spaces). Using Logistic Regression classification with 10-fold validation, the combination of n-gram and gender had the highest performance with an F1-score of 0.7393, closely followed by char n-gram with a score of 0.7389.

.**Disadvantages**

- It doesn’t focus on offensive and hate speech detection for Chatbot.
- Only the logistic regression algorithm is used with cross-validation**.**

B. Proposed system:

Although machine learning has been successfully used to detect offensive and hate speech in several English contexts, the distinctiveness of Chatbot and the similarities among offensive, hate, and free speeches require domain-specific English corpus and techniques to detect the offensive and hate speech. Thus, we developed an English corpus from Chatbot and evaluated different machine-learning techniques to detect offensive and hate

speech. Character n-gram, word n-gram, negative sentiment, syntactic-based features, and their hybrid were extracted and analyzed using hyper-parameter optimization, ensemble and multi-tier meta-learning models of support vector machine, logistic regression, random forest, and gradient boosting algorithms.

Advantages:

- The proposed system is totally focused on offensive and hate speech detection for Chatbot.
- Multiple machine learning algorithm is used for predicting the classes of tweets.

C. Modules Description

NLTK: NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

Lemmatization: Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word.

Tokenization: Tokenization is the process of tokenizing or splitting a string, text into a list of tokens. One can think of token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph.

LITERATURE SURVEY

The proliferation of toxic online information has emerged as a significant concern in contemporary society, mostly driven by the exponential growth in internet use with individuals from diverse cultural and educational backgrounds. Distinguishing between hate speech and offensive language is a significant obstacle in the automated identification of harmful textual material. This study presents a method for automatically categorizing messages on Twitter into three distinct classes: hostile, offensive, and clean. Utilizing the Twitter dataset, we conduct experiments by treating n-grams as features and supplying their term frequency-inverse document frequency (TFIDF) values to several machine learning models. We conduct a comparative study of the models, taking into account various values of n in n-grams and TFIDF normalizing approaches. Upon optimizing the model to get the most favorable outcomes, we attain a 95.6% accuracy while assessing it using the test data. Additionally, we developed a module that functions as an intermediary between the user and Twitter.

Instances of hate speech, characterized by racist and sexist comments, are often seen on social media platforms. Due to this, several social media platforms tackle the issue of recognizing hate speech. However, the definition of hate speech differs significantly and is mostly done manually (BBC, 2015; Lomas, 2015). We use a set of standards derived from critical race theory to annotate a publicly accessible collection of over 16,000 tweets. We examine the influence of several non-linguistic characteristics together with character n-grams for identifying hate speech.

Additionally, we provide a lexicon that is derived from the most representative terms found in our dataset. The proliferation of social networks and microblogging platforms has facilitated direct connection between individuals of diverse cultural and psychological backgrounds, leading to an increase in cyber disputes among them. As a result, hate speech is increasingly being employed, to the extent that it has become a significant issue infiltrating these public areas. Hate speech is the act of using aggressive, violent, or abusive words to specifically target a certain group of people who share a similar characteristic, such as their gender (sexism), their ethnic group or race (racism), or their beliefs and religion. Although many online social networks and microblogging services have policies against hate speech, the sheer scale of these platforms makes it very difficult to monitor and regulate all of the information posted on them. Hence, it is essential to automatically identify and censor any communication that contains hostile words or language that encourages hatred. This study presents a methodology for identifying hate speech on the Twitter platform. The foundation of our technique relies on unigrams and patterns that are gathered automatically from the training set. These patterns and unigrams are then used, among other things, as features to train a machine-learning system. The results of our experiments on a test set consisting of 2010 tweets demonstrate that our approach achieves an accuracy of 87.4% in determining whether a tweet is offensive or not (binary classification), and an accuracy of 78.4% in determining whether a tweet is hateful, offensive, or clean (ternary classification).

TECHNOLOGY DESCRIPTION AND IMPLEMENTATION

Introduction To Python Framework: An overview of Django This book focuses on Django, a Web development framework that enhances efficiency and brings pleasure to the process of Web development. Django enables the creation and management of top-notch Web applications with minimum effort. Web development may be a stimulating and imaginative process when done well, but it can also become monotonous and exasperating when not executed well. Django allows you to concentrate on the enjoyable aspects of your web application, which are the most important parts, while reducing the difficulty of the repetitive components. By doing this, it offers sophisticated representations of typical Web development patterns, time-saving methods for common programming jobs, and explicit guidelines for problem-solving. Simultaneously, Django endeavors to avoid interfering with your work, allowing you to operate outside the framework's limitations when necessary. The objective of this book is to transform you into a proficient Django practitioner. The emphasis is dual. Initially, we provide a comprehensive explanation of the functionalities and capabilities of Django, as well as the process of constructing Web applications using this framework. Additionally, we explore more advanced principles, if relevant, addressing the query "How can I proficiently utilize these tools in my personal projects?" Through the act of reading this book, you will acquire the necessary expertise to efficiently create robust websites, using code that is both organized and simple to maintain.

A web framework is a software framework that provides a structure and tools for developing web applications. It offers a set of libraries, components, and patterns that help streamline the development process and enhance the

efficiency and scalability of web applications.

Django is a major constituent of a contemporary cohort of Web frameworks. Could you please provide a precise definition for that term? In order to address this inquiry, let us examine the architecture of a Web application developed using the Common Gateway Interface (CGI) standard, which was a widely used method for creating Web applications around the year 1998. During that time period, while developing a CGI application, it was necessary to handle all aspects of the process independently, similar to preparing a cake entirely from basic ingredients. As an example, presented below is a basic CGI script, implemented in Python, that displays the 10 most recently released books from a database.

```
import MySQLdb

print "Content-Type: text/html"
print
print "<html><head><title>Books</title></head>"
print "<body>"
print "<h1>Books</h1>"
print "<ul>"

connection = MySQLdb.connect(user='me', passwd='letmein', db='my_db')
cursor = connection.cursor()
cursor.execute("SELECT name FROM books ORDER BY pub_date DESC LIMIT 10")
for row in cursor.fetchall():
    print "<li>%s</li>" % row[0]

print "</ul>"
print "</body></html>"

connection.close()
```

This code is simple and easy to understand. Initially, it outputs a line stating the "Content-Type", followed by an empty line, as mandated by CGI. The code generates initial HTML content, establishes a connection to a database, and conducts a query to get the 10 most recent books. Iterating over those books, it produces an HTML unordered list. Lastly, it outputs the closing HTML and terminates the database connection.

For a unique and dynamic website like this, using the write-it-from-scratch technique may be a viable option. Firstly, this code is easily understandable - even an inexperienced developer can comprehend just 16 lines of Python and grasp its whole functionality, from beginning to end. There is no more knowledge to acquire; no additional programming language to study. Deploying this code is straightforward: store it in a file named latestbooks.cgi, upload the file to a Web server, then access the page using a browser. However, as a Web application expands beyond being simple or basic, this method becomes ineffective, and you encounter many issues:
Is it necessary for a developer to be concerned with publishing the "Content-Type" line and ensuring the database connection is closed? This kind of standardized template decreases the efficiency of programmers and creates possibilities for errors. It would be most efficient to delegate these chores linked to setting up and dismantling to a shared infrastructure.

When this code is repeated in various contexts, each with a distinct database and password, the following will occur:

At this juncture, it is crucial to have certain configuration settings that are relevant to the context.

CheckedWhat occurs when a Web designer lacking proficiency in Python coding desires to revamp the webpage? It would be ideal if the logic of the page, which involves retrieving books from the database, could be separated from the HTML presentation. This would allow a designer to make changes to the appearance of the page without impacting the functionality.The purpose of a Web framework is to specifically address these issues. A Web framework offers a programming foundation for applications, allowing developers to concentrate on building efficient and manageable code without the need to create everything from scratch. Essentially, it is the core function of Django.

Python

What Is A Script?

Up to this point, I have concentrated on the interactive programming capability of Python. This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode

Scripts are reusable

Basically, a script is a text file containing the statements that comprise a Python program. Once you have created the script, you can execute it over and over without having to retype it each time.

Scripts are editable

Perhaps, more importantly, you can make different versions of the script by modifying the statements from one file to the next using a text editor. Then you can execute each of the individual versions. In this way, it is easy to create different programs with a minimum amount of typing.

You will need a text editor

Just about any text editor will suffice for creating Python script files.

You can use *Microsoft Notepad, Microsoft WordPad, Microsoft Word,* or just about any word processor if you want to.

IMPLEMENTATION

Implementation is the process of converting a new or revised system design into operational one. There are three types of Implementation:

- Implementation of a computer system to replace a manual system. The problems encountered are converting files, training users, and verifying printouts for integrity.
- Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned there can be many problems.
- Implementation of a modified application to replace an existing one using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in the files. Implementation in Generic tool project is done in all modules. In the first module User level identification is done. In this module every user is identified whether they are genuine one or not to access the database and also generates the session for the user. Illegal use of any form is strictly avoided.

In the Table creation module, the tables are created with user specified fields and user can create many table at a time. They may specify conditions, constraints and calculations in creation of tables. The Generic code maintain the user requirements through out the project.

In Updating module user can update or delete or Insert the new record into the database. This is very important module in Generic code project. User has to specify the filed value in the form then the Generic tool automatically gives whole filed values for that particular record.

SNO	Test case Title	Pre-requisites	Action	Expected Result	Test Result(pass/fail)
Test case 1	Software requirements	Python version 3.6.5	python --version (Checking the version)	Python 3.6.5 present in your system	Pass
Test case 2	Idle requirements	Jupyter notebook	CMD--(jupyter notebook)	jupyter file should run on the local host	Pass
Test case 3	packages need	pandas, numpy,sklearn,matplotlib,nltk	ls(list of packages)	all packages should import	Pass
Test case 4	Import the datset	Import the dataset by using pandas	Datset found (uploaded into jupyter)	show the dataset in the ipynb file	Pass
Test case 5	Data cleaning	All data should be read from dataset	Apply stemming, lemetization on the data	It will give clean text suitable for machine learning algorithms.	Pass
Test case 6	Prediction	Model should be trained with training dataset	Use trainded model on test dataset for prediction	It will display the predicted results	Pass

INPUT AND OUTPUT DESIGN

The following some are the projects inputs and outputs.

Inputs:

- Importing the all required packages like numpy, pandas, matplotlib, scikit – learn and required machine learning algorithms packages.
- Setting the dimensions of visualization graph.
- Downloading and importing the dataset and convert to data frame.

Outputs:

- Preprocessing the importing data frame for imputing nulls with the related information.
- All are displaying cleaned outputs.
- After applying machine learning algorithms it will give good results and visualization plots.

INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization,
- Internal Outputs whose destination is within organization and they are the
  - User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly with

The outputs were needed to be generated as a hard copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

OUTPUT SCREENS

```
def tweet_rater(tweet, clf):
    tweet = clean_tweet(tweet)
    tweet = lemmatize(tweet)
    tweet = remove_stop(tweet)

    print(tweet)
    rating = clf.predict([tweet])
    probability = clf.predict_proba([tweet])
    print(probability)
    if rating == 0:
        print('I\'m {:2.4}% sure that\'s not offensive.'.format(probability[0][0]*100))
    elif rating == 1:
        print('I\'m {:2.4}% sure that\'s offensive.'.format(probability[0][1]*100))
    else:
        print('I\'m {:2.4}% sure that\'s hate speech.'.format(probability[0][2]*100))
```

```
tweet_rater('how are you man', pipe_rf)
```

```
man
[[0.82 0.11 0.07]]
I'm 82.0% sure that's not offensive.
```

```
tweet_rater('Bobby flay in this bitch', pipe_xgb)
```

```
Bobby flay bitch
[[0.06747387 0.8297399  0.10278623]]
I'm 82.97% sure that's offensive.
```

```
tweet_rater('Bobby flay in this bitch', pipe_lr)
```

```
Bobby flay bitch
[[0.03965848 0.84157017 0.11877135]]
I'm 84.16% sure that's offensive.
```

FUTURE ENHANCEMENTS

Our future efforts will be dedicated to enhancing the identification of hate speech and inflammatory language. This will be achieved by combining improved Support Vector Machine and Gradient Boosting classifiers with multi-tier meta-learning classifiers. The suggested algorithms and deep neural network techniques, such as LSTM, will be used to design and evaluate novel word dense embeddings for the purpose of detecting offensive and hate speech in Chatbots.

CONCLUSION

This study involves the compilation of an English corpus specifically designed for the purpose of detecting offensive and hate speech in Chatbot interactions. After the tweets were tokenized and preprocessed, four separate feature sets and their combinations were retrieved. The study utilized three types of enhanced machine learning models, namely hyper-parameter optimization, ensemble, and multi-tier meta-learning. These models were applied to various machine learning algorithms, including Logistic Regression, Support Vector Machine, Random Forest, and Gradient Boosting. The objective was to classify tweets into three categories: hate speech, offensive speech, or free speech. The results of the experiment demonstrated that the Support Vector Machine, Random Forest, and Gradient Boosting multi-tier meta-learning model had the highest level of consistency and balance in detecting offensive and hate speech.

BIBLOGRAPHY

For software installation:

https://www.anaconda.com/download/

https://www.python.org/downloads/release/python-360/

References:

S. Dredge. (2014). Twitter Changes: 20 Hits and Misses From the Social Network's History. The Guardian. Accessed: Sep. 10, 2019. [Online].
Available:https://www.theguardian.com/technology/2014/oct/22/twitterchanges-hits-misses history

1. Gaydhani,V. Doma, S.Kendre, and L. Bhagwat, ``Detecting hate speech and offensive language on Twitter using machine learning: An N-gram and TFIDF based approach," in *Proc. IEEE Int. Advance Comput. Conf.*, Sep. 2018, pp. 1_5.
2. H. Watanabe, M. Bouazizi, and T. Ohtsuki, ``Hate speech on Twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE Access*, vol. 6, pp. 13825_13835, 2018.
3. Ijteba Sultana, Dr. Mohd Abdul Bari ,Dr. Sanjay," *Routing Performance Analysis of Infrastructure-less Wireless Networks with Intermediate Bottleneck Nodes*", International Journal of Intelligent Systems and Applications in Engineering, ISSN no: 2147-6799 IJISAE,Vol 12 issue 3, 2024, Nov 2023
4. Md. Zainlabuddin, "*Wearable sensor-based edge computing framework for cardiac arrhythmia detection and acute stroke prediction*", Journal of Sensor, Volume2023.
5. Md. Zainlabuddin, "*Security Enhancement in Data Propagation for Wireless Network*", Journal of Sensor, ISSN: 2237-0722 Vol. 11 No. 4 (2021).
6. Dr MD Zainlabuddin, "*CLUSTER BASED MOBILITY MANAGEMENT ALGORITHMS FOR WIRELESS MESH NETWORKS*", Journal of Research Administration, ISSN:1539-1590 | E-ISSN:2573-7104 , Vol. 5 No. 2, (2023)
7. Vaishnavi Lakadaram, " Content Management of Website Using Full Stack Technologies", Industrial Engineering Journal, ISSN: 0970-2555 Volume 15 Issue 11 October 2022
8. Dr. Mohammed Abdul Bari,Arul Raj Natraj Rajgopal, Dr.P. Swetha ," *Analysing AWSDevOps CI/CD Serverless Pipeline Lambda Function's Throughput in Relation to Other Solution*", International Journal of Intelligent Systems and Applications in Engineering , JISAE, ISSN:2147-6799, Nov 2023, 12(4s), 519–526
9. Ijteba Sultana, Mohd Abdul Bari and Sanjay," *Impact of Intermediate per Nodes on the QoS Provision in Wireless Infrastructure less Networks*", Journal of Physics: Conference Series, Conf. Ser. 1998 012029 , CONSILIO Aug 2021

10. Naif Ismail Ibrahim, Mohd Kamran Khadeer, Mohammed Abdul Kareem, Vikas Kumar Tiwari, Mohammad Rafeeq, Mr.B.Tejavardhan, Design Optimization And Analysis Of Aircraft Landing Gear, International Journal of Multidisciplinary Engineering in Current Research - IJMEC Volume 8, Issue 6, June-2023, http://ijmec.com/, ISSN: 2456-4265.
11. Altaf Nawaz Khan, Mendy Madhavi, Mohammad Abdul Hai, Mohammed Faiz Ur Rahman, Md Abdulla Zaid, Mr. N. Vinay Kumar, Design And Structural Analysis Of Hovercraft Using Ansys Software, International Journal of Multidisciplinary Engineering in Current Research - IJMEC Volume 8, Issue 6, June-2023, http://ijmec.com/, ISSN: 2456-4265.
12. Gurumoorthy P, Adike Shiva Kumar, G. Prashanth, V. Ashok, Mr. B.Narendar Rao, , Design And Structural Analysis Of Tri Powered Go Kart, International Journal of Multidisciplinary Engineering in Current Research - IJMEC Volume 8, Issue 6, June-2023, http://ijmec.com/, ISSN: 2456-4265.
13. Bapanpalli Madhu, Challa Srikar, Diviti Ravi, Gavinolla Madhavi, K. Rajesh Kumar, Mr.N. Vinay Kumar, Design And Thermal Analysis Of Portable Solar Electric Air Cooler, International Journal of Multidisciplinary Engineering in Current Research - IJMEC Volume 8, Issue 6, June-2023, http://ijmec.com/, ISSN: 2456-4265.
14. Ijteba Sultana, Dr. Mohd Abdul Bari ,Dr. Sanjay," *Routing Performance Analysis of Infrastructure-less Wireless Networks with Intermediate Bottleneck Nodes*", International Journal of Intelligent Systems and Applications in Engineering, ISSN no: 2147-6799 IJISAE,Vol 12 issue 3, 2024, Nov 2023
15. Md. Zainlabuddin, "*Wearable sensor-based edge computing framework for cardiac arrhythmia detection and acute stroke prediction*", Journal of Sensor, Volume2023.
16. Md. Zainlabuddin, "*Security Enhancement in Data Propagation for Wireless Network*", Journal of Sensor, ISSN: 2237-0722 Vol. 11 No. 4 (2021).
17. Dr MD Zainlabuddin, "*CLUSTER BASED MOBILITY MANAGEMENT ALGORITHMS FOR WIRELESS MESH NETWORKS*", Journal of Research Administration, ISSN:1539-1590 | E-ISSN:2573-7104 , Vol. 5 No. 2, (2023)
18. Vaishnavi Lakadaram, " Content Management of Website Using Full Stack Technologies", Industrial Engineering Journal, ISSN: 0970-2555 Volume 15 Issue 11 October 2022
19. M.A.Bari, Sunjay Kalkal, Shahanawaj Ahamad," *A Comparative Study and Performance Analysis of Routing Algorithms*", in 3rd International Conference ICCIDM, Springer - 978- 981-10-3874-7_3 Dec (2016)
20. Mohammed Rahmat Ali,: BIOMETRIC: AN e-AUTHENTICATION SYSTEM TRENDS AND FUTURE APLLICATION", International Journal of Scientific Research in Engineering (IJSRE), Volume1, Issue 7, July 2017
21. Mohammed Rahmat Ali,: BYOD.... A systematic approach for analyzing and visualizing the type of data and information breaches with cyber security", NEUROQUANTOLOGY, Volume20, Issue 15, November 2022

22. Mohammed Rahmat Ali, Computer Forensics -An Introduction of New Face to the Digital World, International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169-453 – 456, Volume: 5 Issue: 7
23. Mohammed Rahmat Ali, Digital Forensics and Artificial Intelligence ...A Study, International Journal of Innovative Science and Research Technology, ISSN:2456-2165, Volume: 5 Issue:12.
24. Mohammed Rahmat Ali, Usage of Technology in Small and Medium Scale Business, International Journal of Advanced Research in Science & Technology (IJARST), ISSN:2581-9429, Volume: 7 Issue:1, July 2020.
25. Dr. Mohammed Abdul Bari,Arul Raj Natraj Rajgopal, Dr.P. Swetha ," *Analysing AWSDevOps CI/CD Serverless Pipeline Lambda Function's Throughput in Relation to Other Solution*", International Journal of Intelligent Systems and Applications in Engineering , JISAE, ISSN:2147-6799, Nov 2023, 12(4s), 519–526
26. Ijteba Sultana, Mohd Abdul Bari and Sanjay," *Impact of Intermediate per Nodes on the QoS Provision in Wireless Infrastructure less Networks*", Journal of Physics: Conference Series, Conf. Ser. 1998 012029 , CONSILIO Aug 2021
27. M.A.Bari, Sunjay Kalkal, Shahanawaj Ahamad," *A Comparative Study and Performance Analysis of Routing Algorithms*", in 3rd International Conference ICCIDM, Springer - 978- 981-10-3874-7_3 Dec (2016)
28. Mohammed Rahmat Ali,: BIOMETRIC: AN e-AUTHENTICATION SYSTEM TRENDS AND FUTURE APLLICATION", International Journal of Scientific Research in Engineering (IJSRE), Volume1, Issue 7, July 2017
29. Mohammed Rahmat Ali,: BYOD.... A systematic approach for analyzing and visualizing the type of data and information breaches with cyber security", NEUROQUANTOLOGY, Volume20, Issue 15, November 2022
30. Mohammed Rahmat Ali, Internet of Things (IOT) Basics - An Introduction to the New Digital World, International Journal on Recent and Innovation Trends in Computing and Communication, ISSN: 2321-8169-32-36, Volume: 5 Issue: 10
31. Mohammed Rahmat Ali, Internet of things (IOT) and information retrieval: an introduction, International Journal of Engineering and Innovative Technology (IJEIT), ISSN: 2277-3754, Volume: 7 Issue: 4, October 2017.
32. Mohammed Rahmat Ali, How Internet of Things (IOT) Will Affect the Future - A Study, International Journal on Future Revolution in Computer Science & Communication Engineering, ISSN: 2454-424874 – 77, Volume: 3 Issue: 10, October 2017.
33. Mohammed Rahmat Ali, ECO Friendly Advancements in computer Science Engineering and Technology, International Journal on Scientific Research in Engineering(IJSRE), Volume: 1 Issue: 1, January 2017

34. Ijteba Sultana, Dr. Mohd Abdul Bari ,Dr. Sanjay, "*Routing Quality of Service for Multipath Manets, International Journal of Intelligent Systems and Applications in Engineering"*, JISAE, ISSN:2147-6799, 2024, 12(5s), 08–16;