

DE GRUYTER

# ALGORITHMS

BIG DATA, OPTIMIZATION TECHNIQUES, CYBER SECURITY

*Edited by Sushil C. Dimri, Abhay Saxena,  
Bhuvan Unhelkar and Akshay Kumar*

AUFLAGE U1

DE GRUYTER SERIES ON THE APPLICATIONS  
OF MATHEMATICS IN ENGINEERING AND  
INFORMATION SCIENCES

DE GRUYTER

# ALGORITHMS

## BIG DATA, OPTIMIZATION TECHNIQUES, CYBER SECURITY

*Edited by Sushil C. Dimri, Abhay Saxena,  
Bhuvan Unhelkar and Akshay Kumar*

AUFLAGE U1



## Algorithms

De Gruyter Series on the Applications of Mathematics in  
Engineering and Information Sciences

---

Edited by

Mangey Ram

Volume 17

# Algorithms

---

Big Data, Optimization Techniques, Cyber Security

Edited by

Sushil C. Dimri

Abhay Saxena

Bhuvan Unhelkar

Akshay Kumar

**DE GRUYTER**

ISBN 9783111228006

e-ISBN (PDF) 9783111229157

e-ISBN (EPUB) 9783111229638

Bibliographic information published by the Deutsche  
Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the  
Deutsche Nationalbibliografie; detailed bibliographic data are  
available on the Internet at <http://dnb.dnb.de>.

© 2024 Walter de Gruyter GmbH, Berlin/Boston

# Übersicht

## 1. Table of Contents

## Contents

### 1. Preface

2.

Rajiv Ranjan Giri, Richa Indu, Sushil Chandra Dimri  
Machine learning-enabled techniques for speech  
categorization

#### 1. 1 Introduction

#### 2. 2 Recent trends in speech categorization

#### 3. 3 Proposed methodology

##### 1. 3.1 Categorizing comments

##### 2. 3.2 Data acquisition and preprocessing

##### 3. 3.3 The proposed algorithm

#### 4. 4 Results and discussion

##### 1. 4.1 Hijab case (Karnataka)

##### 2. 4.2 Boycott movies dataset

##### 3. 4.3 Dharamsansad (Haridwar)

#### 5. 5 Conclusion

3.

Akash Dogra, Shiv Ashish Dhondiyal, Sushil Chandra Dimri

## Comprehensive study of cybersecurity issues and challenges

1. 1 Introduction
2. 2 Cybercrime
3. 3 Cyberspace threats
4. 4 Notable recent cybercrime
5. 5 Cybersecurity
6. 6 Cybersecurity policy
7. 7 Future direction
8. 8 Conclusion

4.

Chandrashekhar Patel, Bhanu Priya Yadav, Aditi Saxena  
An energy-efficient FPGA-based implementation of AES  
algorithm using HSTL IO standards for new digital age  
technologies

1. 1 Introduction
  1. 1.1 The Advanced Encryption Standard (AES)
  2. 1.2 FPGA
  3. 1.3 HSTL (high-speed transceiver logic) IO standard
2. 2 Literature review
3. 3 Design methodology
  1. 3.1 Simulation
  2. 3.2 RTL analysis
  3. 3.3 Synthesis
  4. 3.4 Implementation

5. 3.5 Program and debug
4. 4 Results and analysis
  1. 4.1 Computing total power consumption of HSTL I 18
  2. 4.2 Computing total power consumption of HSTL II
  3. 4.3 Computing total power consumption of HSTL II 18
5. 5 Conclusion
6. 6 Future scope

5.

Bhawmesh Kumar, Ashwani Kumar, Harendra Singh Negi,  
Ishwari Singh Rajput

A comparative study on security issues and clustering of  
wireless sensor networks

1. 1 Introduction
  1. 1.1 Popular research areas of WSNs
  2. 1.2 Comparison of clustering objectives and energy  
consumption of node
  3. 1.3 Security and privacy issues in WSN
2. 2 Critical literature review
3. 3 Research gaps identified
4. 4 Conclusion

6.

Deonarain Brijlall, Tauqeer Ahmed Usmani, Richa Indu  
Heuristic approach and its application to solve NP-complete  
traveling salesman problem

1. 1 Introduction

2. 2 Related work
3. 3 Techniques and mathematical models
  1. 3.1 The traveling salesman problem
  2. 3.2 Nearest neighbor approach
  3. 3.3 Greedy approach
  4. 3.4 The proposed heuristic solution
4. 4 Results and discussion
  1. 4.1 Identifying the optimal route of a milk van for 10 houses
  2. 4.2 Identifying the optimal route of a milk van for 20 houses
5. 5 Conclusion
7. Megha Shah, Akshay Kumar, Shristi Kharola, Mangey Ram  
Assessment of fake news detection from machine learning  
and deep learning techniques
  1. 1 Introduction
  2. 2 Literature survey
  3. 3 Methodology
    1. 3.1 Natural language processing\_(NLP)
    2. 3.2 Machine learning\_(ML)
    3. 3.3 Deep learning\_(DL)
    4. 3.4 Study framework
    5. 3.5 Datasets

6. 3.6 Data cleaning steps (preprocessing the dataset)
7. 3.7 Models utilized in the study
4. 4 Results and analysis of datasets
  1. 4.1 Results for dataset 1
  2. 4.2 Result for dataset 2
5. 5 Conclusion and future work
8.  
Harendra Singh Negi, Aditya Bhatt, Vandana Rawat  
Spam mail detection various machine learning methods and  
their comparisons
  1. 1 Introduction
  2. 2 Related works
  3. 3 Purposed work
  4. 4 Implementation and result
    1. 4.1 Performance metrics
    2. 4.2 Time elapsed
    3. 4.3 AUC-ROC
    4. 4.4 PR curve
  5. 5 Conclusion
9.  
Rahul Bijalwan, Vandana Rawat, Akshita Patwal, Sudhanshu  
Maurya  
Cybersecurity threats in modern digital world
  1. 1 Introduction
  2. 2 Literature review

3. 3 Issues in cybersecurity
4. 4 Attacks with their classification
  1. 4.1 Phishing
  2. 4.2 SQL injection
  3. 4.3 Brute force attack
  4. 4.4 Malware
  5. 4.5 Man-in-the-middle
5. 5 Data breaches
6. 6 Conclusion

10.

Parth Gautam, Apurva Omer, Jeetendra Pande, Devesh Bora  
Mechanism to protect the physical boundary of organization  
where the private and public networks encounter

1. 1 Introduction
2. 2 Literature review
3. 3 Purpose and objectives
4. 4 Overview of private and public networks
5. 5 Security considerations for private and public networks
  1. 5.1 Network segmentation
  2. 5.2 Segmentation techniques
  3. 5.3 Use of firewalls and intrusion prevention systems (IPS)
  4. 5.4 Access control mechanisms
  5. 5.5 Network access control (NAC) solutions

6. 5.6 Physical security measures
6. 6 Future trends and considerations
7. 7 Conclusion

11.

Bhawmesh Kumar, Aditya Bhatt, Neeraj Panwar

By combining binary search and insertion sort, a sorting method for small input size

1. 1 Introduction
2. 2 Related works
3. 3 Proposed work
  1. 3.1 Algorithm
  2. 3.2 Time complexity
  3. 3.3 In the worst case of binary insertion sort
  4. 3.4 For average case of binary insertion sort
  5. 3.5 Time complexity analysis
4. 4 Numerical illustration
5. 5 Conclusion

12. Index

1. V
2. 1
3. 2
4. 3
5. 4
6. 5

7. 6
8. 7
9. 8
10. 9
11. 10
12. 11
13. 12
14. 13
15. 14
16. 15
17. 16
18. 17
19. 18
20. 19
21. 20
22. 21
23. 22
24. 23
25. 24
26. 25
27. 26
28. 27
29. 28
30. 29

31. 30

32. 31

33. 32

34. 33

35. 34

36. 35

37. 36

38. 37

39. 38

40. 39

41. 41

42. 42

43. 43

44. 44

45. 45

46. 46

47. 47

48. 48

49. 49

50. 50

51. 51

52. 52

53. 53

54. 55

55. 56

56. 58

57. 59

58. 60

59. 61

60. 62

61. 63

62. 64

63. 65

64. 66

65. 67

66. 69

67. 70

68. 71

69. 72

70. 73

71. 74

72. 75

73. 76

74. 77

75. 78

76. 79

77. 80

78. 81

- 79. [82](#)
- 80. [83](#)
- 81. [84](#)
- 82. [85](#)
- 83. [86](#)
- 84. [87](#)
- 85. [88](#)
- 86. [89](#)
- 87. [90](#)
- 88. [91](#)
- 89. [92](#)
- 90. [93](#)
- 91. [94](#)
- 92. [95](#)
- 93. [96](#)
- 94. [97](#)
- 95. [98](#)
- 96. [99](#)
- 97. [100](#)
- 98. [101](#)
- 99. [102](#)
- 00. [103](#)
- 01. [104](#)
- 02. [105](#)

03. [106](#)
04. [107](#)
05. [108](#)
06. [109](#)
07. [110](#)
08. [111](#)
09. [112](#)
10. [113](#)
11. [114](#)
12. [115](#)
13. [116](#)
14. [117](#)
15. [118](#)
16. [119](#)
17. [120](#)
18. [121](#)
19. [122](#)
20. [123](#)
21. [124](#)
22. [125](#)
23. [126](#)
24. [127](#)
25. [128](#)
26. [129](#)

27. [130](#)

28. [131](#)

29. [132](#)

30. [133](#)

31. [134](#)

32. [135](#)

33. [137](#)

34. [138](#)

35. [139](#)

36. [140](#)

37. [141](#)

38. [142](#)

39. [143](#)

40. [144](#)

41. [145](#)

42. [146](#)

43. [147](#)

44. [149](#)

45. [150](#)

46. [151](#)

47. [153](#)

48. [156](#)

49. [157](#)

50. [159](#)

- 51. [160](#)
- 52. [161](#)
- 53. [162](#)
- 54. [165](#)
- 55. [166](#)
- 56. [167](#)
- 57. [168](#)
- 58. [169](#)
- 59. [170](#)
- 60. [171](#)
- 61. [172](#)
- 62. [173](#)
- 63. [174](#)
- 64. [175](#)
- 65. [176](#)
- 66. [177](#)
- 67. [183](#)
- 68. [184](#)

## Preface

Algorithms play a vital role in all sciences, especially in computer science. We are always in search of efficient

algorithms that give the results in less amount of time and that consume less space for large input size.

Cybersecurity and big data are two prominent areas of computer science, where the role of algorithms is vital.

Algorithms provide the best security mechanism to protect the data from all types of attacks. There are several types of attacks such as threats to data safety, data security, and on the mechanisms to store and retrieve the data.

Cybercrime is a hard reality nowadays, and to deal with that we require full proof of algorithms which provides best data security, data safety, and protection from all kinds of attacks and crimes, which are changing their shapes, nature, and methodology with each passing moment.

Big data is a complicated scattered huge data which increases in size with time. Data processing, data safety, data operations, data information extraction, and so on are all equally important. Big data is the area that deals with huge amount of data that needs sound high-performance algorithms. To extract the relevant information, data manipulation, access, and retrieval are big challenges.

Optimization is a common term used almost everywhere, including our daily life. This book comprises chapters on the study of different algorithms used in cybersecurity and big data, and also puts light on optimization methods used in computer science. A comparative study of different algorithms is also provided. This book is helpful to students, especially for those who work in areas of cybersecurity and big data.

# Machine learning-enabled techniques for speech categorization

**Rajiv Ranjan Giri**

**Richa Indu**

**Sushil Chandra Dimri**

## **Abstract**

In the modern era trolling, abusive language, violation of the right to freedom of speech in the name of free speech, and bigotry, to incite hatred and disharmony among people or groups on various grounds are ubiquitous on social platforms. Therefore, this work not only categorizes comments on such platforms into hostile speech but also identifies religion-oriented, offensive, violence-provocating, and normal speeches. In this way, our proposed algorithm uses traditional yet simple word-matching criteria, with complexity  $O(n)$  to categorize speech into five classes, by identifying the words in comments identical to three predesigned sets, named  $S_R$  (set of religion-oriented words),  $S_O$  (set of offensive words), and  $S_V$  (set of violence-provocating words). These words are selected on the basis of the different IPC sections to tackle the cases of hate speeches. To test our algorithm, we extracted comments from three different incidents, that is, the Hijab case (Karnataka),

Boycott movies, and Dharmasansad (Haridwar) from different YouTube news channels and their respective Facebook pages, and created three datasets. These datasets are then preprocessed to maintain anonymity and obtain clean data for categorization. Furthermore, the total similarity index is also computed to determine the kind of words used most in these datasets. However, after preprocessing the final sample sizes are 116, 372, and 861 for three datasets in respective order, where the misclassification rate of the present approach is 2.58%, 1.344%, and 0.813% of the total sample sizes, conveying that the presented approach generalizes better with large samples. To handle imbalanced data SVMSMOTE is used, whose output is fed to linear-, polynomial-, and RBF-kernel support vector machine, logistic regression, Gini index and entropy criteria decision tree, and random forest classifiers to estimate the efficacy of the proposed algorithm. The highest and the lowest efficacy of 99.03% and 92.95% for the Dharmasansad (Haridwar) dataset, 98.66% and 92.95% for the Boycott movies dataset, and 97.42% and 89.87% for the Hijab case (Karnataka) dataset are attained with random forest and polynomial-kernel SVM, respectively. However, the work also has some limitations like the inability to categorize comments with deep sarcasm, skepticism, and a lack of comprehension of the context of

statements endorsing violence, where the explicit use of violence-provocative phrases is absent.

**Keywords:** Hostile speech, religion-oriented, violence-provoking, offensive, speech categorization,

## 1 Introduction

In a society, it is natural to have disagreements and agreements of opinions. Gradually, as a consequence of the political and social milieu, underlying stereotypes, and the rest, which we see in the guise of news and entertainment turns disagreements into hatred leading to certain negative repercussions like polarization and demoralization. In this day and age, hate speech, trolling, abusing, and violating the right to freedom of speech in the name of free speech are quite common on various social platforms like Facebook, Twitter, and YouTube channels. Targeting people and cohorts on the grounds of religion, community, caste, gender, profession, race, and so on is referred to as hate (or hostile) speech [[→1](#)]. Such speeches spread hatred, and threatened trolled people to feel isolated, helpless, and afraid for their safety. Moreover, sometimes this kind of social behavior ends in riots and merciless killing of the victimized people, generating a hate crime affecting the complex relationship in a society to live indifferently together with peace in a country. Thus, it can be said that any kind of

detesting speech leaves a devastating impact on human society like increased casteism, communalism, disharmony, and increasing disbelief among people from different ideologies. Consequently, such speeches require timely identification and neutralization before any unfortunate incident may occur or be made to happen as a cause-and-effect chain.

Depending on intellect, environment, and mindset, humans can effectively identify sarcasm, hatred, abuse, praise, denial, neutrality, and any emotion behind the speech. But for a machine, it is merely a combination of ASCII characters. The machines are unable to identify the sentiments behind a sentence. Thus, to impart such knowledge to machines, several machine learning, deep learning, or natural language processing (NLP) algorithms are utilized. Traditionally, machine learning used several supervised classifiers for detecting hate speech, such as naive Bayes (NB), support vector machines (SVMs), extreme gradient boosting (XGBoost), logistic regression (LR), random forest (RF), k-nearest neighbors (kNNs), and decision tree (DT), which, later on, is advanced with usage of multilayer perceptron (MLP), long short-term memory (LSTM) networks, Bi-LSTM, convolutional neural networks (CNNs), and NLP techniques [[→2](#), [→3](#)]. These days, a combination of NLP tools and deep learning techniques is the most popular, such as bag-of-words, term frequency-inverse document frequency (TF-

IDF), word embeddings, word2vec, Glove, and FastText. These methods turn a word into a vector representation capturing its essence. Further, the modern transformer-based embedding techniques, such as Bidirectional Encoder Representations from Transformers (BERT), Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA), and ALBERT (or A Lite BERT) can also be used with deep learning models built using LSTM, Bi-LSTM, and CNN. Thence, these algorithms not only identify fake news and hostile speeches but also determine the tone of a speech, that is, joke, sarcasm, offense, and so on.

Besides this, detecting hate speech is a complex and difficult task. First, a sentence that may hurt one person's sentiments may not necessarily offend others [[→3](#)]. Secondly, a huge amount of data is required to be preprocessed for filtering emojis, shortcuts, jumbled words, mixed-code languages, etc., which is time- as well as effort-consuming [[→4](#)]. This increases the number of comparisons and thus the complexity of the work. Furthermore, the problem of misclassification arose due to the presence of ironical statements. Lastly, a clear understanding of the domain of speech is a must. Statistics show an upsurge in the incidents of hate speech in India in recent years [[→5](#), [→6](#), [→7](#)], but unfortunately, currently, there has been no law defining hate speech. However, such related

matters are dealt with under IPC under Sections 124A, 153, 295A, 298, and 505 [[→1](#)]. The commonality among these provisions is the promoting disharmony, hatred, or insults on the basis of religion, ethnicity, culture, language, region, caste, community, race, gender, and so on, via spoken, written, or visual is a punishable offense, either as financial penalty or prison. Therefore, we propose a speech categorization method based on traditional word-matching criteria from a predefined set of words, implemented with the help of the Natural Language Toolkit (NLTK) in Python. This approach distinguishes between five kinds of speeches, which are hostile, offensive, religious, violenceful, and normal.

Further, the work is organized into five sections, where [→Section 2](#) comprises recent trends in speech categorization, and [→Section 3](#) consists of legal provisions for dealing with hate speech, data acquisition, and preprocessing along with the proposed work. Results are discussed in [→Section 4](#), followed by the conclusion in Section 5.

## **2 Recent trends in speech categorization**

Valnkar et al. [[→3](#)] discussed major challenges in detecting hate speech by forming hierarchal grouping. These major challenges are the lack of existence of a universal definition for hate

speech, comments written in multiple languages, use of symbols, and jumbled text, identifying the sarcasm, joke, irony, and other tones of speeches, and the need for the background knowledge of the context of the comment. In this series, Kindermann provided a thorough discussion regarding hate speech and also provided several pieces of evidence supporting the use of the term “discriminatory speech” instead of “hate speech” [→8]. Now, for incorporating context-aware mechanisms in the problem of hate speech detection in the Roman Urdu language, Bilal et al. [→9] implemented a transformer-based approach named the BERT-RU model. With 173,714 text messages, their first Roman Urdu pretrained BERT model claimed the superiority of BERT-RU not only over the traditional ML algorithms but also over the deep learning approaches like LSTM, BiLSTM, BiLSTM with attention layer, and CNN, by attaining 96.70%, 97.25%, 96.74%, and 97.89% accuracy, precision, recall, and *F*-measure, respectively. Moreover, William et al. [→10] attained 79% accuracy in determining hate speech messages using the Bigram feature set with the SVM.

Similarly, Boishakhi, Shill, and Alam [→11] proposed a multimodel system to detect hate speech from video content by extracting feature images, and feature values can be extracted from the audio and text by using machine learning and NLP. A

Bengali language dataset comprising 30,000 crowd-sourced and verified comments from YouTube and Facebook for detecting hate speech in the Bengali language was curated by Romim et al. [[→12](#)]. Using word embeddings such as Word2Vec, FastText, and BengFastText, they experimentally illustrated 87.5% accuracy with SVM and deep learning approach. In another work associated with the Bengali language, Ishmam and Sharmin [[→13](#)] classified 5,126 annotated Bengali comments into 6 classes, viz. hate speech, communal attack, inciteful, religious hatred, political comments, and religious comments. Using RF only 52.2% accuracy was achieved, which was improved to 70.1% when the gated recurrent unit (GRU)-based model was used. Identification of cyberbullying and hate speech from 14,000 comments extracted from the most visited Facebook pages, written as a mixture of the local Chadian and French languages, was demonstrated in [[→14](#)]. Out of these, four categories (hate, offense, insult, and neutral) were identified and classified using SVM, LR, RF, and kNN, after employing the Word2Vec, Doc2Vec, and FastText word embeddings on the NLP-cleaned data. The experimental results revealed that the FastText feature representation fed to the SVM classifier was 95.4% accurate for predicting the insult category statement, followed by 93.9% for hate statements. Furthermore,

work embodied in [[→2](#), [→15](#)] reviewed the methodologies for automated hate speech detection and classification of tweets.

Taking into account that manual detection of hatred-inducing tweets is a challenging operation, and very few efforts were made to detect the same in South Asian languages. Khan et al. [[→16](#)] undertook hate speech detection in Roman Urdu text. Out of 5,000 Roman Urdu tweets filtered from a scrap of more than 90,000 tweets, an iterative approach was developed to classify this corpus at three levels: neutral-hostile, simple-complex, and offensive-hate speech. During the experimental evaluation of their suggested approach, LR surpassed other methods including deep learning methods with an *F*-measure of 90.6% and 75.6% for discriminating between neutral and hostile tweets, and offensive and hate speech tweets, respectively. A religion, ethnicity, nationality, and gender-based hate speeches textual corpus from Twitter was curated by Alsafari et al. [[→17](#)] using contextual word embeddings for feature extraction. With the aid of several ML and deep learning techniques for classification, they developed an effective Arabic hate and offensive speech detection framework. Abbasi et al. [[→18](#)] focused on analyzing and classifying the multilabel toxic comments based on religion, race, and ethnicity, such as Muslim, Jewish, White, and Black. Providing high toxicity ratings to these four words was disagreed by them as the same

can be used in different contexts and thus dataset can appear biased. For word embeddings, they applied GloVe, Word2vec, and FastText before deploying NN, CNN, LSTM, BiLSTM, GRU, and BiGRU for the final classification. The maximum accuracies achieved by Glove word embeddings with the CNN model were 96.59%, 92.91% with Word2Vec, and 92.07% using FastText word embedding. However, as per precision-recall and *F1* score, RNN outperformed the rest by attaining scores of 96.72% and 96.77%, respectively.

Another work focusing to detect cyber hate speech in the Arabic language was discussed in [[→19](#)]. They used 843 hate tweets and 790 neutral tweets for positive-negative classification including emotions on the ground of racism, journalism, sledging, terrorism, and Islam, using NLP, TF-IDF, and classification algorithms like SVM, NB, DT, and RF. Again, Paul [[→20](#)] considered Twitter data for separating speech into hate speech, offensive, and normal language, using SVM, LR, ANN, and NB, where NN achieved the highest 94% accuracy and 96% *F*-measure. Bohra et al. [[→21](#)] demonstrated the detection of hate speech on the 4,575 Hindi-English code-mixed tweets. After preprocessing the tweets into vectors, and performing feature identification and extraction using character N-grams, word N-grams, punctuations, lexicon, negations, and all features, a total of 177 words were classified as hate speech. The experiential

outcome revealed that the SVM performs better than the RF classifier by attaining the highest accuracy of 71.7% when all features are used. Meanwhile, character N-grams proved to be most efficient with SVM, while word N-grams resulted in high accuracy scores when used with RF. Bhardwaj et al. [→22] detected 1,638, 1,132, 1,071, and 810 posts as fake, hate, offensive, defame, and nonhostile content from 8,200 annotated online posts in the Hindi language using the CONSTRAINT-2021 dataset. Their fine- and coarse-grained evaluation inferred SVM as the best detector of hate (47.49%), offensive (41.98%), and defamation (43.57%) posts, whereas LR outperforms others in detecting fake posts with an *F1* score of 68.15%. Moreover, the best weighted *F1* score of 84.11% was given by SVM, whereas 83.98%, 83.45%, and 79.79% weighted *F1* scores were achieved by LR, MLP, and RF, respectively.

### 3 Proposed methodology

According to the 267th report of the Law Commission of India, hate speech is defined as “the incitement to hatred towards a group of people based on their race, ethnicity, gender, sexual orientation, or other characteristics” [→1]. Laws in India do not exclusively define any section against hate speech, though the Indian Penal Code (IPC) has some legal provisions to prohibit

the violation of the right of freedom of speech. These are Sections 124A, 153, 295A, 298, and 505 [[→1](#), [→23](#), [→24](#), [→25](#), [→26](#), [→27](#)] constituting:

*Section 124A:* Any person(s) can be penalized with imprisonment for life (which is extendable to 3–5 years) and/or financial penalty, if he/she/they are found involved in provoking disrespect, enmity, or hatred between different classes of the citizens of India on grounds of religion, race, caste, community, or language, by verbal, audio, or visual media.

*Section 153:* Any person(s) if found wantonly provoking people or group(s) for riots, shall be castigated for life imprisonment (which is extendable for 6 months) along with a fine. Similarly, disturbing the harmony as well as public tranquility between different communities based on their religion, race, caste, community, etc. is punishable under Sections 153A and 153B respectively. For this, the convicted person(s) shall be imprisoned for 3–5 years along with a fine.

*Section 295A:* If any person(s) is found guilty of intending to outrage the religious feelings of any class of citizens of India by any means, then she/he/they can be imprisoned for 3 years along with a financial penalty.

*Section 298:* This section deals with penalizing the culprit with imprisonment for a term extendable for a year, who

either by gesture, expression, sound, or by placing any objectionable object in the sight of the targeted person(s) intentionally hurt the religious feelings of other people.

*Section 505:* Any person(s) circulating rumor or fake news, report, or statement intended to hurt the sentiments of people and conducting mischievous activities to disintegrate social harmony can be penalized with a fine and/or imprisonment for 3 years, which can also be extended up to 5 years, depending on the repercussions of the statement/report.

Further, Sections 123(3A) and 125 have been provisioned to punish any propaganda that insults or generates ill will on grounds of religion, race, caste, language, or community during elections. Besides these, the Protection of Civil Rights Act (1955), the Religious Institutions (Prevention of Misuse) Act (1988), the Cable Television Network Regulation Act (1995), and the Cinematograph Act (1952) are also provisioned to protect and punish against the spread of hatred via different mediums.

### **3.1 Categorizing comments**

Keeping in view the aforementioned juridical provisions, we can categorize the instances of comments into five categories, namely, hostile comment (*H*), normal comment (*N*), offensive

comment (O), religious comment (R), and violenceful comment (V). For this, we extracted a set of words in both English and Hindi languages, where Hindi is written in an English manuscript. These selected words and/or phrases are tailored to manifest any violation of the above-said IPC sections and acts.

The first set includes a set of words identifying a religion like  $S_R = \{muslims, Hindu, Sikhs, Buddhist, Hindutva, Christians, Islam, Islamic, baudh \text{ (indicating Buddhism in Hindi language)}, Mulla, maulana, Rohingyas, god, bhagwan, Allah, InShaAllah, orange/green/red flag, Islamist, maulvis, Dharma \text{ (represents Religion in Hindi language)}, jehadi, halaala, musalman \text{ (signifies Muslim in Hindi)}, Buddh, Jain, Jainism\}$ .

The second set  $S_V$  comprises words and phrases which can provoke violence or talk about violence, that is,  $\{Bloodbath, Kill, Burnt Alive, Chopped, Raped, Mocked, Terrorism, Terrorist, Aag \text{ (Fire)}, Badla \text{ (Revenge)}, Beheading Heads, Cutting Tongues, Looted, Goondaism \text{ (Thuggery)}, Massacre, Threat, Destroying, Swords, Shoot, Violence, Genocide, Katna \text{ (To Cut)}, Marna \text{ (To Beat)}, Jan Se Marna \text{ (To Kill)}, Khoon ki Nadiyan Bhahana \text{ (Bloodshed)}\}$ .

The final set  $S_O$  consists of abusive and offensive words and phrases, which are  $\{Kuttey \text{ (Dogs)}, Foolhardy, Bewkoof \text{ (Stupid)}, False, Fake, Fool, Foolish, Disrobe, Blind, Dumb, Deaf, Goons, Idiot(S), Bloody, Idiotic, Nalayak \text{ (Useless)}, Absurd, Scoundrel,$

*Impotent, Rascals, Vaishya (Women or Girl Who Works in the Redlight Area), Bandoroun (Monkeys), Hijra (Transgender), Sala (Brother-of-Wife), Gobar (Cow Dung), Suar (Pig), Kala (Black), Gorey (White), Kshatriya, Shudra (SC/ST), Brahmin, OBC, Wtf, abuse}.*

More words can be added to these sets depending on the context of the speeches.

*Religious comments (R):* The words or phrases addressing a specific religion, which is not necessarily an ill-will but talks about a religion, its customs, traditions, values, ideals, and so on. Such phrases even include slogans particular to a religion like *Jay Shree Ram (Hindus)*, *Allah hu Akbar (Muslims)*, *Wahe guru ji da khalsa wahe guru ji di fateh (Sikhs)*, *Zindabad*, and *Jai ho*.

This group comprises comments having religion-specific words and phrases as mentioned in the set  $S_R$  above.

*Violenceful comments (V):* The words or phrases promoting the conceptual as well as physical incitement of any kind of violence and even death against any group, community, gender, or religion are considered under this category. This group consists of the most sensitive words that are questionable if stated against any religion, caste, race, gender, community, and so on as also aforesaid in IPC sections. Such words mentioned in the set  $S_V$  can be used to discuss or

mention the brutality that happened at some point in history unless they are specifically targeting a group or person.

*Offensive comments (O):* Abusing or insulting persons or a group of persons, living or nonliving entity, dehumanizing, defaming, trolling, and comparing them to subhuman entities, such as pigs, rats, dogs, shit, monkeys, germs, and so forth, on the ground of caste, creed, religion, gender, or with the words in the set  $S_O$ , are considered under  $O$  category.

*Hostile comments (H):* Generally, these comments are religiously offensive, inciting violence against an out-group, offensive as well as provoking violence, or a combination of all. The words and phrases in  $S_R$ ,  $S_O$ , and  $S_V$  categories alone do not categorize a statement or comment as hostile speech. Thus, comments that are posted intended to insult a religion, gender, community, caste, person, or group of people by utilizing words from at least two different above-said groups are regarded as hostile.

*Normal or neutral comments (N):* The comments involving agreements, disagreements, nonhostile thoughts, free and normal forms of speech, sarcasm, metaphors, and so on are categorized as neutral or normal comments.

### **3.2 Data acquisition and preprocessing**



The data used to test the efficiency of the proposed work has been extracted from the popular YouTube news channels in India and their Facebook pages. To keep the anonymity of the data, the channel, subscriber, and commenter's name and gender are removed during preprocessing. In this way, a list of comments from anonymous users written in English and Hindi languages are obtained, where Hindi comment is inscribed in the Roman manuscript. Three different kinds of data are extracted and selected from Indian news platforms in this series:

1. Hijab case (Karnataka): This dataset comprises the comments of debates over wearing of hijab in schools and colleges, which became the talk of the town in 2022 after the verdict of the Supreme Court.
2. Boycott movies: This dataset comprises comments regarding the boycott of movies. First is Padmavat (2018) on the ground of hurting the sentiments of a community belonging to a region in the name of freedom of direction. The second is Pathaan (2023), which was chosen as a boycott candidate for offending some groups by using certain colors (or color-based discrimination).
3. Dharmasansad (Haridwar): This dataset comprises the comments regarding the Dharmasansad held in Haridwar from December 17 to 19, 2021, by Hindu sages. It has been an

issue of huge controversy over most of the news channels in India during the last year due to the reported hostile speeches and few arrests.

Before feeding data to selected classifier, pre - processing is required. Besides converting the comments into lowercase, the emojis are converted to their alternate texts, the abbreviated words are written in full form, and the misspelled words are corrected. Moreover, these misspelled words are mostly written intentionally, which comprises a mixture of lowercase and uppercase letters, introducing unnecessary letters in the word, and amalgamating words from multiple languages. For instance, writing *gorment* instead of government, *jeL* for jail (prison), *Midiya* for media, \* *Dhanny Bad* \* for dhanyavaad (thank you), *jeneralism* or *jarnilisam* for journalism, *pulice* for police, and so on. Finally, we also removed all the punctuation marks and replaced the indecent abuses with the word “abuse.” Some examples of preprocessed comments are mentioned in [→Table 1.](#)

**Table 1:** Instances of data preprocessing in the proposed work.

S. no.	Original comments	Processed comments
1	Ye sarkari Manav hi. inko jeL nhi ho skti	Ye sarkari Manav hai inko jail nahi ho sakti (He is a government man, he cannot be jailed)
2	We support  and love   .	We support Heart Fire Flag Flag Flag and love Heart Folded Hands Fire flag flag
3	Sir pulice to nind k dawa kha kr so Gaya h jo matre 30 me bikta h	Sir police to neend ki dawa kha kar so Gayi hai, jo matr 30 me bikti hai (Sir, the police have fallen asleep after consuming sleeping medicine, which is sold for only 30)

S. no.	Original comments	Processed comments
4	Thanks to hilights this issue on your air.	Thanks to highlight this issue on your air
5	Salute you sir for your fair and square jeneralism ., ... 🇮🇳 This is the real jarnilisam	Salute you sir for your fair and square journalism Folded hands This is the real journalism
6	Truth' 🇮🇳 🌟	Truth 100%
7	Jai Hindutva 🇮🇳 ✂️ 🚩 🚩 🚩	Jai Hindutva Hindu Symbol Om Swords Red Flag Red Flag Red Flag

### 3.3 The proposed algorithm

Once the extracted data is preprocessed, the Natural Language ToolKit (NLTK) library [[→28](#)] of Python version 3.11 is used for tokenizing the comments into words and identifying the different kinds of nouns in the comments. For non-English comments (i.e., comments written in Hindi), the identification of foreign words is performed. The count of each noun and foreign word has been recorded simultaneously for each comment. Alike traditional spam detection approaches, the proposed algorithm uses very traditional word-matching criteria to assort the comments into different categories. Therefore, each noun in a comment is searched for its appearance in either set, which is  $S_R$  (religious),  $S_O$  (offensive), and  $S_V$  (words illustrating violence). Now, if nouns in a comment are found to be in more than one or all three sets, then it would be categorized under category  $H$ , that is, hostile speech. Similarly, if the noun appears identical to any word or phrase in the set  $S_R$  only, then the comment will be assorted under category  $R$  (religious comment). The comments will be assigned to categories  $V$  (violenceful comment) and  $O$  (offensive comment) if a noun matches with the words/phrases in the sets  $S_V$  and  $S_O$ , respectively. In case no noun or foreign word is found in common with any of these sets, then that comment will be assigned category  $N$ . Furthermore, a total similarity index

(tot\_sim\_index) is also computed for each comment representing its similarity with four of the assigned categories excluding  $N$ , since the tot\_sim\_index for category  $N$  is always 0. The upper and lower limits of tot\_sim\_index are 1 and 0, respectively. It can be given as

$$\text{tot\_sim\_index} = (C_R + C_V + C_O)/n \quad (1)$$

where  $n$  is the total number of nouns and foreign words in the comment,  $C_R$  is the number of words common with set  $S_R$ ,  $C_V$  is the number of words common with the set  $S_V$ , and  $C_O$  is the number of words common with  $S_O$ . The pseudocode of the proposed work is:

### **Speech Categorization**

***Input:*** Pre\_processed Data

***Output:*** Categories  $\{H, N, O, V, R\}$

***Step 1.*** Tokenize words using NLTK

***Step 2.*** Identify nouns and foreign words and count  $n$

***Step 3.*** Match the occurrence of each noun or foreign word with its presence in the sets  $S_R$ ,  $S_V$ , and  $S_O$ . And keep a count of each occurrence in  $C_R$ ,  $C_V$ , and  $C_O$ .

**Step 4.** Assigning the respective categories based on the nature of word:

1. Category 'H' is assigned to a comment, if
$$C_R > 0 \text{ and } C_V > 0 \text{ and } C_O > 0, \text{ or if}$$
$$C_R > 0 \text{ and } C_V > 0, \text{ or if}$$
$$C_V > 0 \text{ and } C_O > 0, \text{ or if}$$
$$C_R > 0 \text{ and } C_O > 0.$$
2. Category 'R' is assigned to a comment, if
$$1 \leq C_R$$
3. Category 'O' is assigned to a comment, if
$$1 \leq C_O$$
4. Category 'V' is assigned to a comment, if
$$1 \leq C_V$$
5. Category 'N' is assigned to a comment, if
$$C_R = 0 \text{ and } C_V = 0 \text{ and } C_O = 0.$$

**Step 5.** Compute Total Similarity Index for each category using Equation (1).

After categorizing the comments for each dataset, the proposed work's feasibility is evaluated using various machine learning methods like Linear, Polynomial, Radial Basis Function (RBF) kernel Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Entropy, and Gini Index criteria Decision

Tree (DT). [[→29](#)]. To overcome the problem of imbalanced data, SVM synthetic modeling and oversampling technique (SVMSMOTE) is used before the data is fed to these classifiers, which generates new samples for the minority class near borderlines using SVM instead of kNN [[→30](#)].

## 4 Results and discussion

### 4.1 Hijab case (Karnataka)

From 116 total preprocessed comments in this dataset, 7, 26, 7, 69, and 7 comments are categorized under the categories hostile comment (*H*), religious comment (*R*), violenceful comment (*V*), normal comment (*N*), and offensive comment (*O*), respectively. Out of these categorizations, the following three comments are misclassified:

*1. There was a sati pratha in Hindu religion where the woman was to be burnt alive on pyre of her dead husband but by law it was banned in India*

(This comment is assigned to category H since it talks about an ancient act resembling violence against women of the Hindu religion in India. However, factually it should be assigned to category N).

*2. leave islam and be a human*

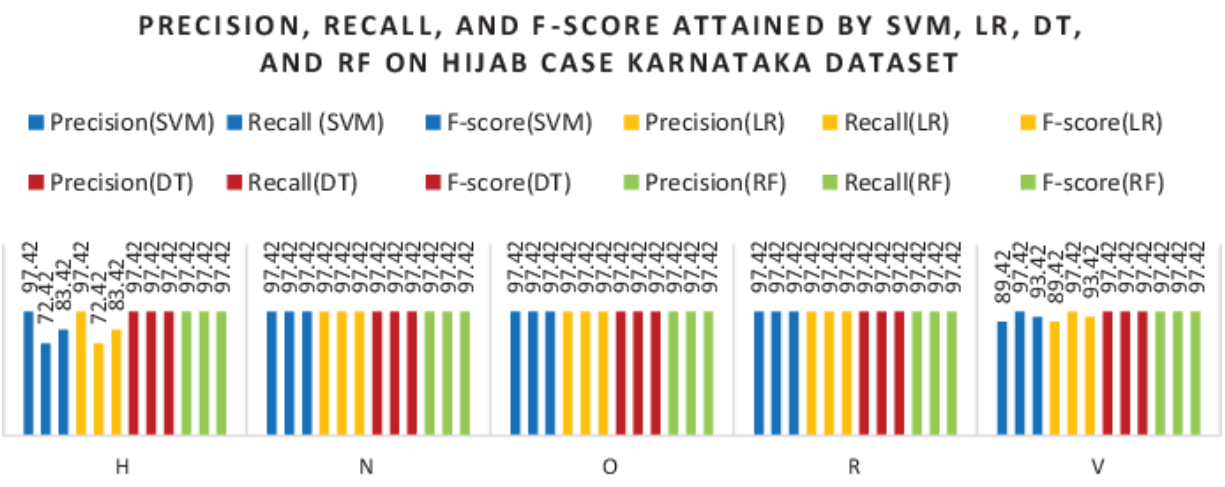
(This comment is assigned to category  $R$  because the proposed algorithm finds an identical word “Islam,” while this comment is religiously offensive. Thus, can be categorized as a hostile comment ( $H$ )).

*3. where are these people when girls are raped?*

(Since the comment again talks about violence against women, thus categorized as a violenceful comment by the proposed algorithm as  $V$ , whereas this comment has a touch of sarcasm, which should have been categorized under normal comment ( $N$ )).

In the Hijab case (Karnataka), the total similarity index (tot\_sim\_index) ranges from 0.285 to 0.6 for the  $H$ , 0.1 to 1 for  $R$ , 0.143 to 0.2 for  $V$ , and 0.076 to 0.6 for  $O$  categories. Since tot\_sim\_index for  $R$  reaches 1, it implies excluding the  $N$  category; this dataset includes more religion-oriented comments than hostile, offensive, or violenceful comments. Now, using SVM SMOTE the data with 116 samples is oversampled to 259 samples, where the recategorization is {“ $N$ ”: 69, “ $R$ ”: 69, “ $V$ ”: 44, “ $O$ ”: 40, “ $H$ ”: 37}, such that 69–69 comments belong to categories  $N$  and  $R$ , 44 comments are categorized under  $V$ , 40 comments under  $O$ , and 37 under the category  $H$ . These 259 resampled data are fed to four supervised classification algorithms, namely SVM, LR, DT, and RF. About 20% of this sample size is reserved as test samples (for

predicting categories), whereas the remaining 80% is used for training these classifiers. The precision, recall, and *F*-score for each category, that is, *H*, *N*, *O*, *V*, and *R*, are depicted in [→Figure 1](#). As evident, DT and RF are perfect classifiers for this dataset considering the misclassification instances by the proposed categorization approach, which is around 2.58% of the total sample size. Therefore, actual performance measures will be the difference between the percentages of the predicted value and the misclassified instances.



**Figure 1:** Precision, recall, and *F*-score attained by linear-kernel SVM (represented in blue color), LR (shown in yellow color), DT with both Gini index and entropy criteria (illustrated in red color), and RF (depicted in green color) on Hijab case (Karnataka) dataset in identifying each category, that is, *H* (hostile comment), *N* (normal comment), *O* (offensive comment), *R* (religious comment), and *V* (violenceful comment) after adapting the portrayed scores with misclassified instances.

The accuracy achieved by different classifiers with different parameters is 95.49%, 89.87%, and 91.65% with linear-, polynomial-, and RBF-kernel SVMs, respectively. Similarly, using

the Gini and entropy criteria of DT, where the depth varies from 3 to 9, and using RF with 100 estimators, an accuracy rate of 97.42% is obtained. However, LR behaved identical to linear-kernel SVM. Averagely, the precision scores achieved by linear-, polynomial-, and RBF-kernel SVMs, LR, DT (with both criteria), and RF are 95.65%, 91.31%, 92.08%, 95.65%, 97.42%, 97.42%, and 97.42%. Likewise, the recall scores are 0.9549, 0.8987, 0.9165, 0.9549, 0.9742, 0.9742, and 0.9742 for linear-, polynomial-, and RBF-kernel SVMs, LR, DT (with both criteria), and RF classifiers. The harmonic mean of the recall and precision scores are 0.954, 0.8943, 0.9107, 0.954, 0.9742, 0.9742, and 0.9742 for the same order of classifiers. However, the category-wise classification of polynomial-kernel SVM yields the precision, recall, and  $F$ -score of 97.42%, 64%, and 78% for the  $H$  category, 97.42%, 97.42%, and 97.42% for the  $N$  and  $O$  categories, 79.42%, 97.42%, and 87.4% for  $R$  category, and 77.42%, 97.42%, and 86.4% for  $V$  category. Similarly, using RBF-kernel SVM, the precision, recall, and  $F$ -score achieved with  $H$ ,  $N$ ,  $O$ ,  $R$ , and  $V$  categories are, respectively, 97.42%, 50.4%, and 67.42%, 91.42%, 97.42%, and 94.4%, 97.42%, 87.42%, and 93.4%, 89.4%, 97.42%, and 93.4%, and 89.42%, 97.42%, and 93.42%.

## 4.2 Boycott movies dataset

The comments categorized by the proposed algorithm under the categories hostile comment (*H*), religious comment (*R*), violenceful comment (*V*), normal comment (*N*), and offensive comment (*O*) are 9, 17, 25, 292, and 29, respectively, from 372 total preprocessed comments. However, the misclassification instances by the proposed categorization approach on the Boycott movies dataset are approximately 1.344% of the total sample size. Hence, the actual performance measures will be obtained by subtracting the predicted percentage from 1.344% (i.e., the miscategorization percentage of the proposed algorithm). The following comments are misclassified in the proposed categorization:

*1. so at last the language they understand, though no civilized society can agree with violence. The question arises that why these liberals and so called moderates target the ideals of Hindu religion again and again on the name of freedom of expression.*

*2. what is doing our law minister he is encouraging all to make Hindu terrorism*

(These comments are miscategorized as *H* since they talk about religion and violence. Nonetheless, both are questioning, the former regarding targeting the ideals of the Hindu religion, and the latter from the law minister.)

*3. this is not a hindu issue. This is a group of goons talking dangerous nonsense*

(Again, this comment is wrongly categorized as *H*, though it should have been considered under *N*.)

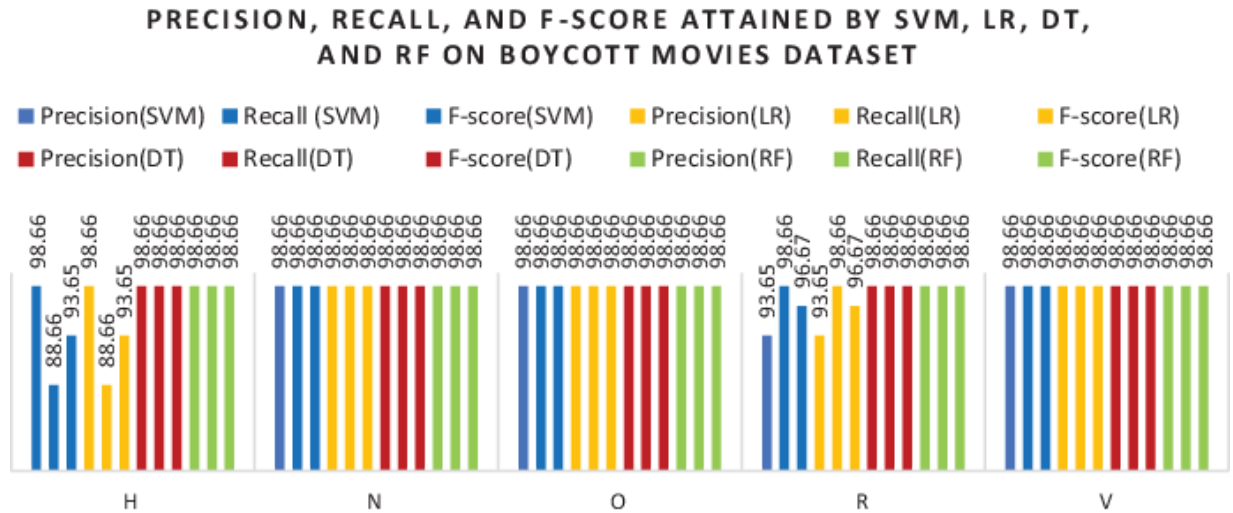
*4. a question to law and order – would these men be arrested for giving death threat?*

(This comment is miscategorized as *V* since it comprises the words “death threat,” while it is a question. Thus to be categorized under *N*.)

*5. they are ready to kill someone because of a movie? Really?? joblessness in India has come to this level.* (Since the comment comprises the word “kill,” it is misidentified under *V* category. While it is sarcasm and has to be categorized under *N*.)

The `tot_sim_index` for the Boycott movies dataset ranges between 0.2 and 1, 0.07 and 1, 0.1 and 0.75, and 0.07 and 0.6 for categories *H*, *R*, *V*, and *O*, respectively. It means this dataset comprises more religion-oriented and hostile comments than violenceful and offensive comments. Again, the sample size of 372 is over- and resampled using SVM SMOTE to 1,313 samples, such that {“*V*”: 292, “*N*”: 292, “*R*”: 292, “*O*”: 292, “*H*”: 145}. This resampled data is fed to SVM, LR, DT, and RF classification algorithms, where 20% of this sample size is reserved as test samples (for predicting categories), and the rest 80% is used for training these classifiers. The accuracy attained is 97.51% by linear-kernel SVM and LR, 92.95%, 95.99% by polynomial- and

RBF-kernel SVM, and 98.66% by RF and DT (with Gini index and entropy criteria). The rest of the performance measures are illustrated in →[Figure 2](#). As evident, DT and RF are perfect classifiers for this dataset as well.



**Figure 2:** Precision, recall, and  $F$ -score attained by linear-kernel SVM (represented in blue color), LR (shown in yellow color), DT with entropy criteria (illustrated in red color), and RF (depicted in green color) on Boycott movies dataset in identifying each category, that is,  $H$  (hostile comment),  $N$  (normal comment),  $O$  (offensive comment),  $R$  (religious comment), and  $V$  (violenceful comment) after adapting with the misclassification rate of 1.344% of the total sample size.

The average weighted precision for all categories is 97.63, 93.61, 96.09, 97.63, 98.66, and 98.66 for linear-, polynomial-, and RBF-kernel SVM, LR, DT with both criteria (with a depth ranging between 4 and 9) and RF, respectively. Likewise, 97.56%, 92.6%, 96.09%, 97.56%, 98.66%, and 98.66% are the average weighted recall scores for the same respective order of classifiers. Again, the  $F$ -measure of LR and linear-kernel SVM are identical, which

is 97.5%. About 92.95% and 95.98% are the  $F$ -scores for the polynomial- and RBF-kernel SVMs with degree 3 and autoscaled gamma parameters, respectively. Both RF and DT achieved the harmonic mean between precision and recall scores equivalent to 98.69%. In the category-wise classification, polynomial-kernel SVM, respectively, attained precision, recall, and  $F$ -scores of 0.987, 0.887, and 0.937; 0.967, 0.8165, and 0.887; 0.806, 0.967, and 0.89; 0.937, 0.987, and 0.967; 0.987, 0.987, and 0.987 for classifying  $H$ ,  $N$ ,  $O$ ,  $R$ , and  $V$  categories. Similarly, the RBF-kernel SVM has achieved precision, recall, and  $F1$  measures of 0.987, 0.857, and 0.916 for classifying instances of category “ $H$ ”; 0.947, 0.987, and 0.965 for classifying category “ $N$ ”; 0.987, 0.967, and 0.97 for category “ $O$ ”; 0.926, 0.957, and 0.937 for category “ $R$ ”; and 0.987, 0.987, and 0.987 for classifying comments in  $V$  category.

### **4.3 Dharamsansad (Haridwar)**

Out of 861 comments, the proposed algorithm categorizes 174 comments under religious comment ( $R$ ), 11 comments under offensive comments ( $O$ ), 22 comments under violenceful comment ( $V$ ), 17 comments under hostile comment ( $H$ ), and the remaining 637 comments under the category normal comment ( $N$ ). However, with this dataset, the misclassification rate using the proposed algorithm is 0.813% of the total sample size. It

means the final efficiency metrics after considering the misclassification instances of the proposed algorithm is the difference between the predicted value and the misclassification rate. The wrongly categorized comments are as follows:

*1. I'm not as disturbed by calls for muslim genocide as I should be the turning point was the 2014 elections the 2014 results and the happenings in months preceding it were the most traumatic political events. I witnessed what's happening now is just the inevitable*

(Since this comment talks about the violence against one community, it is miscategorized in the *H* category, whereas it is a statement of the *N* category, which is considering the political events responsible for such events.)

*2. If state and central government are so impotent that they cannot act against such elements which promote violence, we Indians should immediately boycott him.*

(This comment uses words from offensive as well as violenceful categories; therefore it is misclassified for *H* category, while it should be in the category *O*.)

*3. when a certain peaceful book encourages and orders that disbelievers should be mercilessly killed nobody is in objecting to its ideology. Why this discrimination? Everyone should be punished for inciting violence*

(The comment is misclassified for the *V* category, while it should be under category *N*.)

*4. we suport dharm sansad we support yati narasimhanand saraswati*

*5. agar hame mita doge to hamara parcham kahi or lehrata dekhoge lekin agar tumhe mita diya to kya hoga*

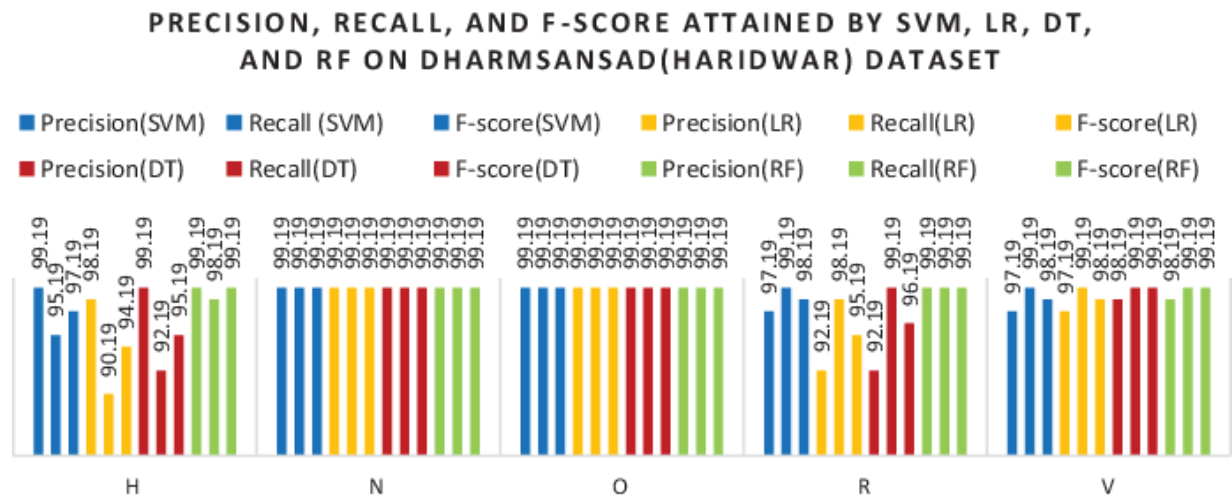
*6. aab samay aa chuka hai jagne ka flag folded hands*  
(Meaning: It is the time to wake up.)

*7. just wake up...support your dharma communal signs.*

The comments from 4 to 7 are also an example of misclassification under the category *N* and should have been classified under category *H*. But due to the lack of any such words or mechanism in the algorithm that would help the machine to understand the context of the statement, these comments are misclassified.

The tot\_sim\_index for Dharmsansad (Haridwar) dataset for categories *H*, *R*, *V*, and *O* ranges from 0.189 to 1, 0.076 to 1, 0.1 to 0.5, and 0.083 to 1, respectively, which means the majority of religion-oriented, offensive, and hostile words are used in comments of this dataset than the brutal words. Using SVM SMOTE oversampling approach, the 861 samples were resampled to 3,210 samples, where {"N": 642, "R": 642, "V": 642, "O": 642, "H": 642}. The accuracy scores achieved on this dataset

with linear-, polynomial-, and RBF-kernel SVMs, LR, and RF are 98.41%, 92.95%, 95.92%, 97.16%, and 99.03%, respectively. DT classifier with entropy criteria provides absolute results at depths of 4 and 6, whereas for the rest of the depths, that is, 3%, 5%, 7%, 8%, and 9%, 99.03% accuracy is attained. Similarly, using the Gini index criteria of DT, 97.63% accuracy is obtained at depths of 3, 5, and 8, and the absolute accuracy is achieved at depths 4, 6, 7, and 9. The category-wise precision, recall, and  $F$ -scores are represented in [→Figure 3](#), where yet again RF and DT are highly efficient classifiers in speech categorization.



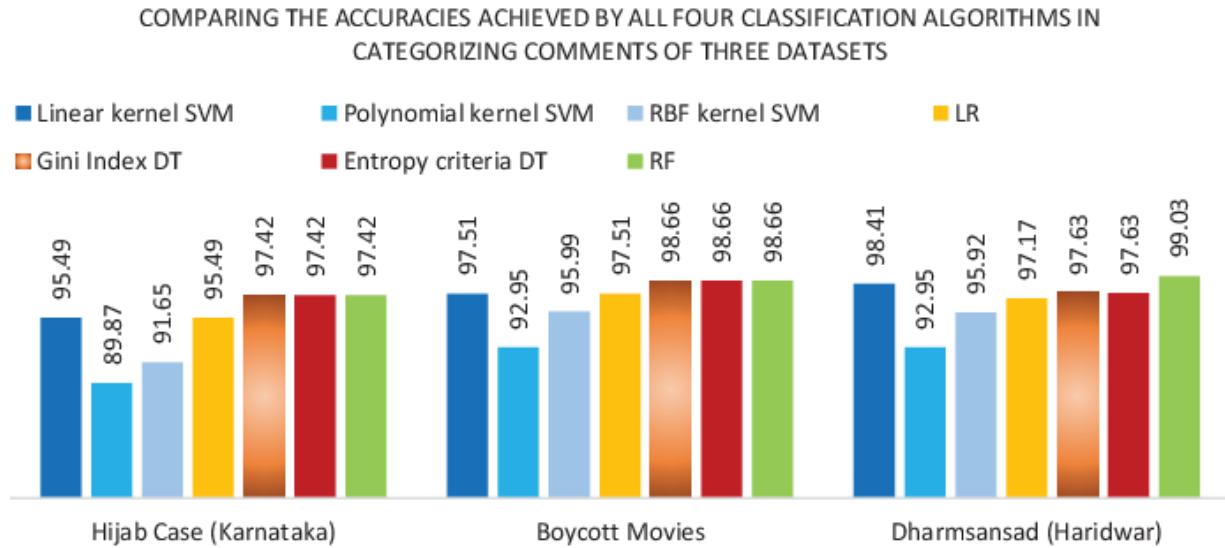
**Figure 3:** Precision, recall, and  $F$ -score attained by linear-kernel SVM (represented in blue color), LR (shown in yellow color), DT with Gini index criteria at depths of 3, 5, and 8 (illustrated in red color), and RF (depicted in green color) on Dharmansad (Haridwar) dataset in identifying each category, that is,  $H$  (hostile comment),  $N$  (normal comment),  $O$  (offensive comment),  $R$  (religious comment), and  $V$  (violenceful comment) after adapting with the misclassification rate.

The average weighted precision is 0.9819, 0.922, 0.962, 0.971, 0.9819, 0.971, and 0.9819 respectively for linear-, polynomial-,

and RBF-kernel SVMs, LR, entropy criteria DT at depths of 3, 5, 7, 8, and 9, Gini index criteria DT at a depth of 3, 5, and 8, and RF respectively. While at the rest of the depths for both criteria, the precision score is 1. The weighted recall scores for the same order and parameters of classifiers are 0.982, 0.932, 0.971, 0.982, 0.971, and 0.982. Similarly, 0.982, 0.932, 0.962, 0.971, 0.982, 0.971, and 0.982 are the respective  $F$ -scores of linear-, polynomial-, and RBF-kernel SVMs, LR, entropy criteria DT at depths of 3, 5, 7, 8, and 9, Gini index criteria DT at a depth of 3, 5, and 8, and RF. The category-wise precision, recall, and  $F1$  scores of the polynomial-kernel SVM are 97.19%, 83.2%, and 89.2% for  $H$  category; 91.2%, 99.2%, and 95.187% for  $N$  category; 85.19%, 85.19%, and 85.19% for  $R$  category; 93.2%, 99.2%, and 96.18% for  $V$  category; and absolute for  $O$  category. Likewise, the RBF-kernel SVM achieved 99.2%, 83.2%, and 91.2% precision, recall, and  $F$ -score for category  $H$ . The same for categories  $R$  and  $V$  are 86.18%, 99.2%, and 92.18% and 97.19%, 99.2%, and 98.18%, respectively, whereas for categories  $O$  and  $N$ , the absolute precision, recall, and  $F$ -measures are attained.

In summary, the RF and entropy-criteria DT classifiers are identified as the most efficient algorithms for classifying YouTube comments with  $F1$  scores of 97.42% for the Hijab case (Karnataka) dataset, 98.69% for the Boycott movies dataset, and 98.2% for Dharmasanasad (Haridwar) dataset. Among various

variants of SVM, linear-kernel has better measures of efficiency than polynomial- and RBF-kernels. Correspondingly, the polynomial-kernel SVM achieved the least  $F$ -measure, which is 89.43%, 92.95%, and 93.2% in the above-said respective order of datasets, while RBF-kernel SVM has a moderate performance, where 91%, 95.98%, and 96.2% are the respective  $F$ -scores for the same order of datasets. Moreover, in Hijab case (Karnataka) and the Boycott movies dataset, LR is almost identical to linear-kernel SVM in terms of efficacy. However, on the Dharmasansad (Haridwar) dataset, a slight dip in the performance metrics of LR can be observed. Additionally, the misclassification made by the proposed algorithm is 2.58%, 1.344%, and 0.813% of the total sample sizes on the Hijab case (Karnataka), Boycott movies, and Dharmasansad (Haridwar) datasets, respectively. It shows that the proposed speech categorization approach generalizes better with more samples at hand than the smaller sample sizes. Therefore, considering the misclassification rate in the accuracies obtained on these three datasets using linear-kernel SVM, LR, DT (with entropy criteria), and RF (with 100 estimators), the final classification accuracies of the SVM-SMOTE oversampled data are shown in [→Figure 4](#).



**Figure 4:** Comparison of the accuracy achieved by all four classification algorithms in categorizing comments on all three datasets.

Finally, the algorithm is simple and less complex with a complexity of  $O(n)$ , where  $n$  is the number of comparisons. As far as the misclassification instances are considered, the proposed work has a lack of understanding of the context of those statements, which were used to support violence without the explicit use of violence-provoking words. Furthermore, this work is not so efficient to handle certain comments involving deep sarcasm and questioning, which we will try to address in future works.

## 5 Conclusion

This work is designed to categorize the comments on social platforms into one out of five categories, which are hostile,

religious, offensive, violenceful, and normal. Traditional and simple word-matching criteria to perform feature categorization is used, based on the occurrence of words in either one or all of three sets, named  $S_R$ ,  $S_O$ , and  $S_V$  containing religion- oriented, offensive, and violence-provocating words, respectively, as per different IPC sections. Comments were extracted from the YouTube and Facebook pages of various news channels reporting three different incidents: (i) the Hijab case (Karnataka), (ii) the Boycott movies, and (iii) Dharmsansad (Haridwar), which are then prepared as three datasets, and preprocessed to maintain anonymity and to obtain clean data for categorization. Furthermore, the total similarity index is calculated to demonstrate the kinds of words used most in these datasets. Finally, to estimate the feasibility of the algorithm, linear-, polynomial-, and RBF-kernel SVMs, LR, Gini index and entropy criteria DT, and RF classifiers are used, after handling the imbalanced data with SVMSMOTE. Considering the misclassification made by the proposed algorithm, which is 2.58%, 1.344%, and 0.813% of the total sample sizes on the Hijab case (Karnataka), Boycott movies, and Dharmsansad (Haridwar) datasets, respectively, the proposed speech categorization approach obtained the highest efficacy of 99.03% for the Dharmsansad (Haridwar) dataset, 98.66% for the Boycott movies dataset, and 97.42% for Hijab case (Karnataka)

dataset with RF classifier. Furthermore, the misclassification rate imparts that the presented approach generalizes better with more samples than the smaller sample sizes. However, the work also has certain limitations revealed while scrutinizing the misclassified instances, like the inability of the work to categorize comments with deep sarcasm, skepticism, or questioning, and understanding of the context of violence-supporting statements, where the explicit use of violence-provoking words is missing. In the future work, we would work to rectify these lacunae and try to identify more tones of speech behind the comments.

## References

[1] Dr. Justice B. S. Chauhan, "267. The Hate Speech (घृणा भाषण)," Twenty-First Law Commission of India, March 23, 2017. Available from:

→[https://lawcommissionofindia.nic.in/report\\_twentyfirst/](https://lawcommissionofindia.nic.in/report_twentyfirst/) [a](#), [b](#), [c](#), [d](#)

[2] F. Alkomah and X. Ma, "A Literature Review of Textual Hate Speech Detection Methods and Datasets," *Information*, vol. 13, no. 6, 2022. →<https://doi.org/10.3390/info13060273> [a](#), [b](#)

[3] A. Velankar, H. Patil, and R. Joshi, "A Review of Challenges in Machine Learning Based Automated Hate Speech Detection," *Computation and Language, Machine Learning*, arXiv:2209.05294. →<https://doi.org/10.48550/arXiv.2209.05294> a, b, c

[4] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate Speech Detection: Challenges and Solutions," *PLoS ONE*, vol. 14, no. 8, 2019. →<https://doi.org/10.1371/journal.pone.0221152> →

[5] N. Barton, "89 Instances of Hate Crimes, Hate Speech across Six North Indian States in Four Months, Communalism," *The Wire*, 9 March 2022. Available from: →<https://thewire.in/communalism/89-instances-of-hate-crimes-hate-speech-across-six-north-indian-states-in-four-months> →

[6] N. Jacob, "Data Check: In Seven Years, India has Seen a 500% Rise in Cases Filed Under Its Hate-speech Law," *Scroll.in*, 23 June 2022. Available from: →<https://scroll.in/article/1026701/data-check-in-seven-years-india-saw-a-500-rise-in-cases-filed-under-its-hate-speech-related-law> →

[7] S. Hrishikesh, "Why People Get Away with Hate Speech in India?," *BBC News*, 14 Apr 2022. Available from:

[→https://www.bbc.com/news/world-asia-india-61090363](https://www.bbc.com/news/world-asia-india-61090363) →

**[8]** D. Kindermann, "Against 'Hate Speech,'" Journal of Applied Philosophy, Wiley Online, pp. 1–23.

[→https://doi.org/10.1111/japp.12648](https://doi.org/10.1111/japp.12648) →

**[9]** M. Bilal, A. Khan, S. Jan, S. Musa, and S. Ali, "Roman Urdu Hate Speech Detection Using Transformer-Based Model for Cyber Security Applications," Sensors (Basel), vol. 23, no. 8, 2023.

[→https://doi.org/10.3390/s23083909](https://doi.org/10.3390/s23083909) →

**[10]** P. William, R. Gade, R. e. Chaudhari, A. B. Pawar, and M. A. Jawale, "Machine Learning based Automatic Hate Speech Recognition System," in 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2022, pp. 315–318.

[→https://doi.org/10.1109/ICSCDS53736.2022.9760959](https://doi.org/10.1109/ICSCDS53736.2022.9760959) →

**[11]** F. T. Boishakhi, P. C. Shill, and M. G. R. Alam, "Multi-modal Hate Speech Detection Using Machine Learning," in 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 4496–4499.

[→https://doi.org/10.1109/BigData52589.2021.9671955](https://doi.org/10.1109/BigData52589.2021.9671955) →

**[12]** N. Romim, M. Ahmed, H. Talukder, and M. S. Islam, "Hate Speech Detection in the Bengali Language: A Dataset and Its

Baseline Evaluation,” in M. S. Uddin and J. C. Bansal (eds.), Proceedings of International Joint Conference on Advances in Computational Intelligence, Algorithms for Intelligent Systems, Singapore: Springer, 2021, pp. 457–468.

[→https://doi.org/10.1007/978-981-16-0586-4\\_37](https://doi.org/10.1007/978-981-16-0586-4_37) →

**[13]** A. M. Ishmam and S. Sharmin, “Hateful Speech Detection in Public Facebook Pages for the Bengali Language,” in 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 2019, pp. 555–560.

[→https://doi.org/10.1109/ICMLA.2019.00104](https://doi.org/10.1109/ICMLA.2019.00104). →

**[14]** M. S. Adoum Sanoussi, C. Xiaohua, G. K. Agordzo, M. L. Guindo, A. M. Al Omari, and B. M. Issa, “Detection of Hate Speech Texts Using Machine Learning Algorithm,” in 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV (USA), 2022, pp. 0266–0273.

[→https://doi.org/10.1109/CCWC54503.2022.9720792](https://doi.org/10.1109/CCWC54503.2022.9720792) →

**[15]** A. Arango, J. Pérez, and B. Poblete, “Hate Speech Detection is Not as Easy as You May Think: A Closer Look at Model Validation,” in Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval – SIGIR'19, Paris, France, 2019, pp. 45–54.

[→https://doi.org/10.1145/3331184.3331262](https://doi.org/10.1145/3331184.3331262) →

**[16]** M. M. Khan, K. Shahzad, and M. K. Malik, "Hate Speech Detection in Roman Urdu," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 20, no. 1, pp. 1–9, 2021. →<https://doi.org/doi:10.1145/3414524> →

**[17]** S. Alsafari, S. Sadaoui, and M. Mouhoub, "Hate and Offensive Speech Detection on Arabic Social Media," Online Social Networks and Media, vol. 19, 2020.  
→<https://doi.org/10.1016/j.osnem.2020.100096> →

**[18]** A. Abbasi, A. R. Javed, F. Iqbal, N. Kryvinska, and Z. Jalil, "Deep Learning for Religious and Continent-based Toxic Content Detection and Classification," Scientific Reports, vol. 12, 2022.  
→<https://doi.org/10.1038/s41598-022-22523-3> →

**[19]** I. Aljarah, M. Habib, N. Hijazi, H. Faris, R. Qaddoura, B. Hammo, M. Abushariah, and M. Alfawareh, "Intelligent Detection of Hate Speech in Arabic Social Network: A Machine Learning Approach," Journal of Information Science, vol. 47, no. 4, 2020.  
→<https://doi.org/10.1177/0165551520917651> →

**[20]** C. Paul, "Hate Speech in Social Networks and Detection Using Machine Learning Based Approaches," in 2023 International Conference on Intelligent Systems, Advanced

Computing and Communication (ISACC), Silchar, India, 2023, pp. 1–7. doi: 10.1109/ISACC56298.2023.10084222. →

**[21]** A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava, “A Dataset of Hindi-English Code- Mixed Social Media Text for Hate Speech Detection,” in Proceedings of the Second Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 36–41.

→<https://doi.org/10.18653/v1/W18-1105> →

**[22]** M. Bhardwaj, M. S. Akhtar, A. Ekbal, A. Das, and T. Chakraborty, “Hostility Detection Dataset in Hindi,” Association for the Advancement of Artificial Intelligence, 2020, arXiv:2011.03588v1. →

**[23]** IPC Chapter VI-Section 124A.Sedition.

→<https://devgan.in/ipc/section/124/> →

**[24]** IPC Chapter VIII-Section 153 Wantonly giving provocation with intent to cause riot, 153A. Promoting enmity between classes and groups and doing acts prejudicial to harmony, 153B. Imputations, assertions prejudicial to national integration.

→<https://devgan.in/ipc/section/153/> →

**[25]** IPC Chapter XV-Section 295.A deliberate and malicious intention of outraging the religious feelings of any class by insulting its religion or religious beliefs.

[→https://devgan.in/ipc/section/295A/ →](https://devgan.in/ipc/section/295A/)

**[26]** IPC Chapter XV-Section 298. Uttering words, ...etc. with deliberate intent to wound religious feelings.

[→https://devgan.in/ipc/section/298/ →](https://devgan.in/ipc/section/298/)

**[27]** IPC Chapter XXII-Section 505.Statements conducing public mischief. [→https://devgan.in/ipc/section/505/ →](https://devgan.in/ipc/section/505/)

**[28]** NLTK Documentation. Available from:

[→https://www.nltk.org/ →](https://www.nltk.org/)

**[29]** M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning, 2<sup>nd</sup> ed., Cambridge, MA: The MIT Press, 2018. [→](#)

**[30]** H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline Over-sampling for Imbalanced Data Classification," International Journal of Knowledge Engineering and Soft Data Paradigms, vol. 3, no. 1, pp. 4–21, 2009.

[→https://doi.org/10.1504/IJKESDP.2011.039875 →](https://doi.org/10.1504/IJKESDP.2011.039875)

# Comprehensive study of cybersecurity issues and challenges

**Akash Dogra**

**Shiv Ashish Dhondiyal**

**Sushil Chandra Dimri**

## **Abstract**

Technology is rapidly advancing, and with this growth comes an ever-increasing need for cybersecurity. Due to the size of the global internet, it is evident that national actors with diverse legal, cultural, and strategic aims would have overlapping zones of influence, leading to cyberspace threats. This interconnectedness also brings about significant cybersecurity challenges and creates a complex cybersecurity landscape that requires comprehensive understanding and effective solutions. In light of these challenges, this chapter addresses the pressing need for cybersecurity measures in our interconnected digital world. It presents a detailed study that delves into the various issues and challenges faced in the field of cybersecurity. This chapter presents a comprehensive study of various issues and challenges faced in cybersecurity. The chapter explores the evolving landscape of cyberthreats and provides insights into the techniques and strategies employed by malicious actors to

compromise digital systems. This chapter also provides a comprehensive overview of cybersecurity challenges, issues, and limitations, as well as emerging trends in the latest technology.

**Keywords:** Cybersecurity, cyberthreats, cybercrime, cyberattacks,

## 1 Introduction

Technology is rapidly advancing, and with this growth comes an ever-increasing need for cybersecurity. Cybersecurity is the practice of protecting networks, systems, and programs from malicious digital attacks [[→1](#)]. These assaults can be carried out with the intent to gain access to, alter, or delete sensitive information, demand money, or interfere with regular corporate activities. In 2021, cybercrime has cost the world's economy over \$787,671 per hour. This equates to \$6,899,997,960 lost globally to cybercriminals throughout the year. As a result, it's critical to put in place robust security rules and safeguards that are intended to guard against these breaches [[→2](#)].

Organizations and corporations may safeguard their systems and data against harmful digital attacks by taking the required precautions and exercising prudence when it comes to cybersecurity. In this chapter, we will discuss the importance of cybersecurity, the risks of cyberattacks, and how to protect your

business from them. Cybersecurity is a growing concern for businesses of all sizes. In today's digital world, organizations must have a plan in place to protect their data and infrastructure from malicious cyberattacks. Cybersecurity is not just about preventing cyberattacks, it is also about protecting the confidentiality, integrity, and availability of company data.

Today, businesses must actively monitor for and respond to threats. This means having the right people, processes, and technology in place to protect against cyberthreats. Companies must be prepared to respond to any potential attack as quickly as possible [[→3](#)]. Cybercrime is an increasing problem that has been made worse by the emergence of sophisticated hacking techniques. The UK, followed by the USA, has the largest density of cybercrime victims per million internet users, according to the most recent figures. With over 3.5 million users now compromised, Russia currently leads the world, followed by the USA with little under 2.5 million.

Cybersecurity is the process of protecting digital networks, systems, and programs from malicious attacks that aim to access, change, or destroy sensitive information, extort money, or interrupt normal business processes. It is a combination of strategies, technologies, and practices that can be implemented to defend against cyberthreats and reduce the risk of data loss,

disruption, or financial loss. Many countries have implemented measures to prevent cybercrime, such as strengthening laws and enforcement, improving network security, and creating public awareness campaigns. These measures have had some success, but cybercrime continues to pose a significant threat. To effectively tackle the issue, governments need to invest more resources into cybersecurity, while businesses and individuals must become more aware and take action to protect themselves [→4]. The goal of cybersecurity is to protect the confidentiality, integrity, and availability of company data and resources. This includes protecting against malicious software (malware) and other malicious activities, such as phishing and social engineering [→5]. Organizations must also have processes and procedures in place to protect against cyberattacks. This includes developing policies, training users, and monitoring for suspicious activity. Cybersecurity is a continual process that requires organizations to stay up-to-date on the latest threats and trends. Cybersecurity is important for a number of reasons. The most important reason is to protect the confidentiality, integrity, and availability of company data and resources. Without proper security measures in place, companies are at risk of suffering damage to their reputation and financial losses due to cyberattacks. Cybersecurity is also important for protecting against data breaches [→6]. Data breaches can occur

when confidential information is accessed by an unauthorized individual or a group. This can lead to financial losses, damage to the company's reputation, and loss of customer trust.

Cybersecurity is important for protecting against other malicious activities, such as phishing and social engineering. These activities can be used by cybercriminals to gain access to company networks and systems. Without proper security measures in place, companies are at risk of suffering financial losses and damage to their reputation [[→6](#)].

Organizations of all sizes must have the right security measures in place to protect against cyberattacks. This includes having the right people, processes, and technology in place. Companies must have a plan in place to respond to any potential attack as quickly as possible. Companies must also educate their employees on the importance of cybersecurity and how to recognize and respond to potential threats. Cyberattacks can have a devastating impact on organizations. Financial losses can occur when cybercriminals gain access to company networks and systems. They can steal data, extort money, or disrupt normal business operations. This can lead to a loss of revenue and higher expenses. Damage to the company's reputation can occur when a cyberattack is made public [[→7](#)]. Customers may become wary of the company and its services.

This can lead to a decrease in sales, as customers may choose to do business with a different company. The loss of customer trust can occur when a data breach is made public. Customers may be reluctant to do business with the company and may choose to do business with a different company. There are several cybersecurity best practices that organizations should follow.

There are many different cybersecurity solutions available. These solutions can help protect against cyberattacks, such as data breaches, malicious software, and other malicious activities. Some of the most popular solutions include firewalls, intrusion prevention systems, antivirus software, encryption, multifactor authentication, access control, and data loss prevention. By investing in the right cybersecurity solutions, organizations can help protect against cyberattacks.

By following the steps outlined in this chapter, organizations can help protect their business from cyberattacks.

Cybersecurity is a continual process that requires organizations to stay up-to-date on the latest threats and trends. By following best practices and investing in the right cybersecurity solutions, organizations can help protect their business from cyberattacks. This chapter covers the current security models and focuses on the problems with cybersecurity threats. The

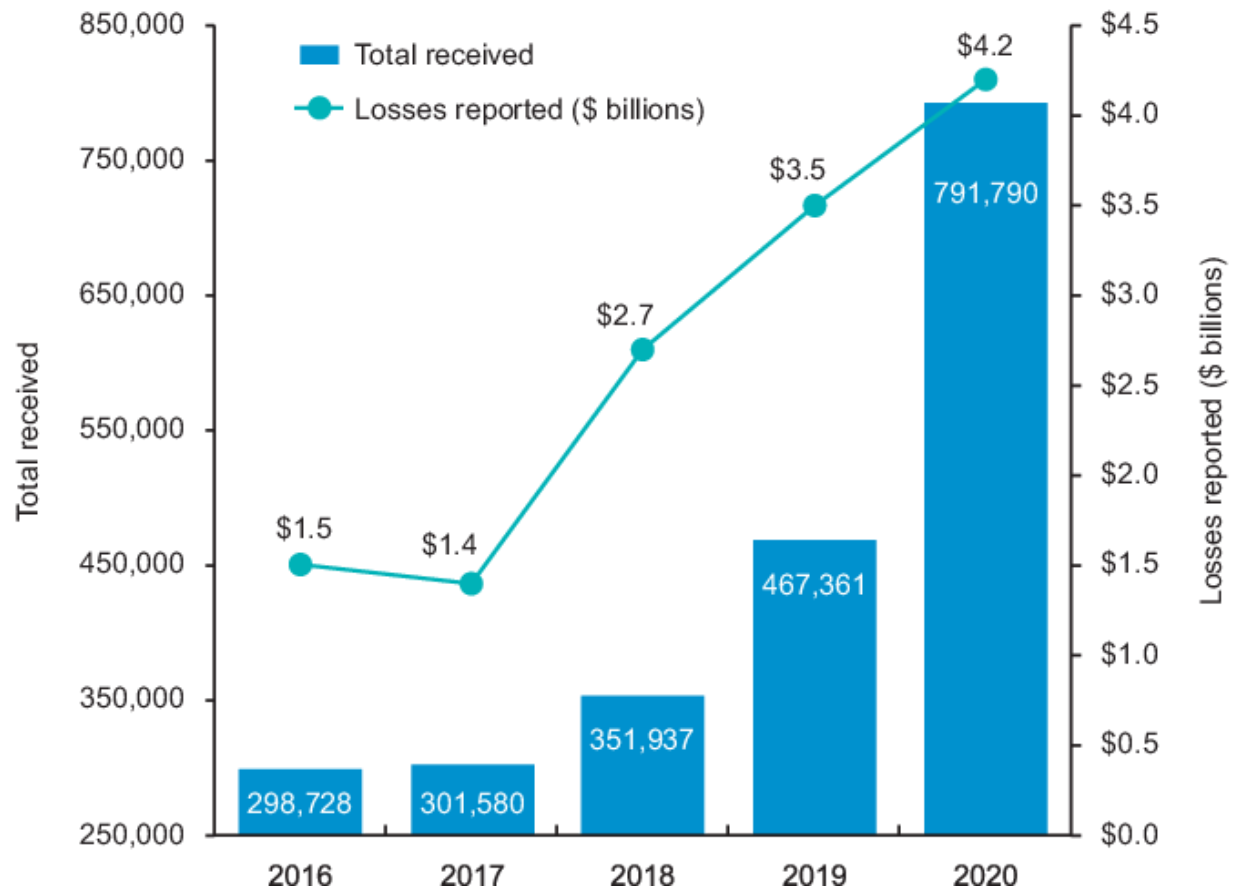
importance of this study is that it will help academics and professionals alike have a comprehensive understanding of the modern cybersecurity industry.

## 2 Cybercrime

Cybercrime is a global phenomenon, with criminals operating across multiple countries and jurisdictions. The effects of cybercrime can be felt in many areas, including the economy, national security, and the personal safety of citizens. The cost of cybercrime is estimated to be in the billions of dollars, with the financial losses caused by cybercriminals increasing each year. One of the biggest challenges in combating cybercrime is the lack of international cooperation. The threat of cybercrime has grown significantly in recent years, with 2020 seeing a 358% increase in malware attacks and 2021 seeing an additional 125% increase in global cyberattacks refer to [→Figure 1](#).

Cybercrime has increased recently, with an annual increase in the number of internet users who fall prey to phishing scams. Victims of investment fraud lost, on average, \$70,811 per incident in 2022 [[→8](#)]. The UK announced the “Ukraine Cyber Programmed” in 2022 to counter these dangers, offering a £6.35 million package to safeguard Ukrainian critical infrastructure. The risk of cybercrime has recently increased because of the

COVID-19 pandemic, with expenses projected to soar to \$10.5 trillion yearly by 2025 [[→9](#)]. Businesses of all sizes need to invest in strong cybersecurity to safeguard against the most recent attacks.



**Figure 1:** Cybercrime cases reported and taken into concern.

For many nations, including India, cybercrime is a growing source of concern. About 208,456 cyber-related crimes were registered in 2018, but this number has increased to 212,485 in the first two months of 2022. This frightening surge

demonstrates how swiftly this kind of criminal activity is spreading [[→9](#), [→10](#)].

Organizations all across the world are at serious risk from cyberattacks, with ransomware and data theft being two of the most frequent attack types. In Europe (26%), North America (30%), and Latin America (29%), ransomware attacks predominated in 2021. In Asia (20%), the Middle East and Africa (18%), and Latin America (18%), server access attacks were the most common attack type [[→10](#)]. In the US, 24,299 victims reported damages of more than \$956 million to the IC3 agency in 2021. The most frequent scams in the US were romance and confidence fraud, with 18,000 complaints and losses of more than \$13.6 million. Sextortion crimes were also widespread. To defend against such assaults, organizations must secure the security of their systems and data and implement a well-defined incident response strategy. Along with a thorough plan for reporting cybercrime occurrences, this should include methods to stop, detect, and respond to cyberthreats. Businesses should spend money on employee training to make sure that their staff members are knowledgeable of potential hazards and how to handle emergencies.

It is predicted that by 2025, 60% of enterprises will consider cybersecurity risk as a major consideration when deciding on

transactions and business engagements with third parties due to the growing awareness of third-party risk. In a poll of 900 businesses, 60% of the respondents said that supply chain attacks – rather than distributed denial of service (DDoS), cyberespionage, advanced persistent threats (APT), ransomware, or data theft – were their top concern.

The pandemic has seen a significant rise in reported cybercrime, with 1,158,208 cases reported in 2020 and 1,402,809 in 2021. In addition, from Q1 to Q2 of 2022, India saw a 15.3% increase in cybercrime. The Atlassian incident of June 2022 is a prime example of the risks that can come from the supply chain, with 180,000 customers in more than 190 countries [[→11](#)]. It was found that nearly 200,000 companies were depending on organizations that could have been affected by the vulnerability. Indian websites have also been targeted, with 17,560 sites hacked in 2018, 26,121 in 2020, and 78% of Indian organizations experiencing a ransomware attack in 2021. Of these, 80% led to data encryption, compared to the average encryption rate of 65%. These figures demonstrate the importance of increasing awareness of third-party risk and the need for organizations to be vigilant in order to protect themselves from data breaches and other cyberattacks. Organizations should take all necessary steps to ensure that

their data is secure and that they are aware of any potential risks in the supply chain.

Cybercriminals often take advantage of the lack of coordination between law enforcement authorities in different countries. This lack of coordination has made it difficult to apprehend cybercriminals and track down their activities [[→12](#)]. To effectively combat cybercrime, governments should work together to develop international standards for cybersecurity. This would allow for greater collaboration between law enforcement agencies and would make it easier to apprehend cybercriminals. Governments should focus on educating their citizens about the risks of cybercrime and how to protect themselves against it. It is also important for businesses to take cybersecurity seriously [[→13](#)]. Companies should ensure that they have adequate security measures in place to protect their networks and data from cybercriminals. This includes having strong passwords, regularly updating software, and using secure communications protocols. Cybercrime is a serious problem that needs to be addressed. Governments and businesses need to come together to develop effective strategies for combating cybercrime. Through collaboration and education, it is possible to protect citizens and businesses from cybercriminals.

### 3 Cyberspace threats

Because of the size of the global internet, it is obvious that national actors with different legal, cultural, and strategic aims would have overlapping zones of influence. It is currently challenging to detach from cyberspace because all nations rely so heavily on it for communication and the management of the real world [[→14](#)]. Thus, the impact of cyberspace on each nation's security activities and responsibilities is growing. It is not possible to make guarantees for the entire product supply chain process because hardware and software are produced on a worldwide scale. The cyberdomain is essentially different due to its size. A bomb's physical range is fairly limited even in the direst situations, but cyberthreats have a far wider impact spectrum; therefore, we have a system that can control real-world activity. Similar to many other specialized industries, cyberspace activities are controlled by a very small number of individuals [[→15](#)]. The hardware and software that users use cannot be changed or managed by the users. It is commonly recognized that a small number of people can effectively manage or direct cyberwarfare. Despite the necessary attention and specialized knowledge, a single person or an organization is unable to completely govern the cyberworld due to its dispersed nature. Based on the continual development of

communication and computer technology, cyberspace is changing quickly. Cybercasting speeds up this acceleration. Every change usher in a fresh era of receptivity and response. Almost much of cyberspace is dynamic and far from static [[→16](#)].

All different types of organizations, from closed governmental systems to those owned and run by the private sector of society, have a vast variety of cyberassets. Each of these enterprises has a unique collection of assets, infrastructure, capabilities, and issues. Technically speaking, it is now impossible to securely trace activities to particular individuals, groups, or organizations due to the nature of cyberspace. In addition to the threats brought on by local forces' insufficient operational skills, the four primary categories of hazards in the supply chain for goods and services are all present in cyberspace [[→17](#)]. Foreign intelligence services utilize cybercapabilities in some of their information gathering and espionage activities. Information infrastructures, such as computer systems, internet information networks, and processors and controllers embedded into crucial enterprises, have frequently been reported to have been exploited or destroyed.

Internally unhappy workers can be a significant source of cybercrime since they frequently have practically unfettered

access to networks. Organizations that fall victim to this kind of assault may not be aware of the workers' hostile intentions or may not have the skills necessary to recognize and stop them [[→18](#)]. Therefore, it is crucial to make sure that every employee is informed of the dangers of cybercrime and to take precautions against it, such as conducting regular security audits. A disciplinary procedure must be in place to deal with any workers who are discovered participating in illegal behavior. By implementing these actions, companies may better defend themselves against the threat of internal cybercrime. Terrorists are another danger since they want to undermine public trust and attitude while also endangering national security by damaging, disabling, or purposefully manipulating vital infrastructure. The cyberattack methods used by them are denial of service assaults, logical bombs, abuse tools, snoopers, Trojan horses, viruses, worms, transmit spam, and botnets [[→19](#)].

When a distributed denial of service attack occurs, authorized users' access to the system and vice versa are lost. At some point, the attacker starts to overwhelm the target systems with messages, preventing the normal flow of data. Because of this, no system can use the internet or communicate with other systems. Another tactic, referred to as "broad denial of service," is attacking simultaneously from several scattered systems

rather than a single source. Worms, which grow on several systems, are usually employed to attack the target in this manner. Tools that can discover and take advantage of network vulnerabilities are available to the general public at different skill levels.

Assault on computer systems can take various forms, ranging from the use of malicious code to gain access to data or systems to the installation of harmful software. Logic bombs, or malicious code inserted into programs, are triggered when certain predetermined conditions are met and can cause damage or disruption to systems or data. Sniffer software is another type of attack that is used to intercept data transmission and scan each packet for specific data, such as passwords [[→18](#)]. Trojan horses are deceptive programs that appear to be beneficial but contain hidden malicious code. Viruses are malicious programs that replicate themselves by inserting copies of themselves into existing system files, enabling them to spread more easily. Unlike viruses, worms are independent software programs that copy themselves from one computer to another on a network and are not reliant upon human interaction to spread.

Electronic soldiers, also known as botnets, are networks of compromised remote-control devices that are used to carry out

malicious activities such as spamming, coordinating attacks, and spreading malware. Botnets are usually installed covertly on target computers, allowing the unauthorized user to gain access to the machine and enact their plans. Botnets are used not only for malicious activities but also for beneficial ones such as distributed computing, distributed denial-of-service attacks, and traffic analysis. However, they can be used to cause significant damage if they are misused, and it is important to ensure that the user has the right security measures in place to combat them.

## 4 Notable recent cybercrime

1. **JBS (José Batista Sobrinho) ransomware attack** [[→19](#)]: On May 30, 2021, the world's largest meat processing company, JBS, suffered a significant cyberattack. Hackers managed to penetrate the JBS network with ransomware, rendering their plants in the USA, Canada, and Australia temporarily inoperative. This posed not only an economic threat to the company but also highlighted weaknesses in the global meat processing supply chain. In response to the attack, JBS issued a statement revealing they employ over 850 IT professionals and spend more than \$200 million a year on IT security [[→20](#), [→21](#)]. Despite these measures, the company had no

choice but to pay the attackers an \$11 million ransom to prevent further disruption and potential data leakage.

2. **Robinhood hack [→21]**: On November 3, 2021, Robinhood, a popular stock trading app, experienced a major data breach. Cybercriminals gained access to the network by social engineering, acquiring employee login credentials and bypassing security measures. As a result, 7 million users had their data accessed and held for ransom. This included email addresses for 5 million users, and full names for 2 million. Three-hundred and ten people were affected more severely, with their dates of birth and zip codes being exposed as well. When the hackers demanded a ransom in exchange for the data, Robinhood refused, instead hiring a cybersecurity firm to investigate the breach [→22, →23]. The company has since implemented increased security measures to protect user data in the future.

3. **Uber hack [→24]**: On September 16, 2022, Uber experienced a data breach when a hacker managed to gain access to their corporate Slack account and cloud account. It is speculated that the hacker had purchased credentials from a contractor that had been previously exposed due to their device being infected with malware [→26]. To gain access, the hacker repeatedly logged into the contractor's account, which caused an "MFA fatigue," where the contractor

eventually accepted the request and the hacker gained entry. In response, Uber took precautionary measures such as blocking or resetting passwords of potentially compromised accounts and resetting access to internal tools [[→25](#)]. The company also locked down its codebase to prevent any changes.

**4. National Health Service (NHS) of UK cybersecurity breach [[→27](#)]:** On August 4, the National Health Service (NHS) of the UK experienced a ransomware attack from a yet unidentified hacking organization. The attack caused several software used for patient check-ins, records, and NHS 111 to be taken offline. GP practices were heavily impacted as access to important patient information was blocked, and notifications could not be sent. To cope with the situation, in-person visits had to be manually recorded, which only increased wait times and put extra strain on the NHS's already struggling workforce [[→26](#)]. Advanced, a security firm, worked on NHS's vulnerabilities and eventually, by August 22, services began to return to normal. The firm is now restoring affected services within a new and secure environment, ensuring that such an attack will not be easily repeated.

**5. Nvidia cyberattack [[→28](#)]:** On February 23, the technology giant Nvidia experienced a data breach that saw

approximately one terabyte of data stolen by the cybercrime group Lapsus\$. Employee information, including account passwords and the source code for graphics card drivers, was also included in this material. In response, Nvidia changed all staff members' passwords to render any leaked information useless. The cybercriminals then allegedly demanded that Nvidia make their drivers open-source as a ransom, although Nvidia has yet to confirm or deny this allegation. In light of the incident, Nvidia has implemented additional security measures to ensure that their data is better protected against similar attacks in the future [[→29](#)]. These precautions will likely include the use of sophisticated firewalls, two-factor authentication, and the regular monitoring of their network for any suspicious activity. Nvidia is also likely to implement more secure protocols for its staff, such as mandating the use of strong passwords and introducing measures to detect any suspicious logins or data transfers [[→29](#)].

**6. Costa Rica ransomware attack 2022 [[→30](#)]:** When Costa Rica saw two waves of ransomware attacks in 2022, it declared a national emergency. The first attack, which took place between the middle of April and the beginning of May, hit 27 government agencies, crippled the customs control IT system, and forced the finance ministry to encrypt 800

servers and many terabytes of data. Trade was hampered as a result, with losses to import and export companies ranging from \$38 to \$125 million a day [[→31](#)]. On May 31, a second attack that targeted the nation's Social Security Fund affected 10,400 machines and more than half of the servers. In the first week after the assault, as a result of this, crucial healthcare systems went offline, forcing doctors to cancel and reschedule 34,677 appointments, or almost 7% of all appointments in the nation that week. A \$10 million ransom was demanded in the initial wave of attacks by the ransomware gang "Conti" in order to stop the release of the data that had been stolen. The HIVE ransomware organization, which has been associated with Conti [[→32](#)], claimed responsibility for the second attack.

**7. WannaCry cyberattack 2017 [[→33](#)]:** One of the worst cyberattacks ever, WannaCry, affected hundreds of thousands of machines all around the world. It took use of an "Eternal Blue" vulnerability in unpatched versions of the Windows operating system, which was purportedly created by the US National Security Agency and discovered by hacker collective The Shadow Brokers. Microsoft responded by issuing a security patch to fix the hole [[→34](#)]. However, a large number of organizations and people neglected to upgrade their systems, making them open to assault. The malicious virus

encrypts data on the afflicted PCs and demands a \$300 Bitcoin ransom from the victims in order to decrypt it. According to estimates, WannaCry cost the world over \$4 billion in losses [[→35](#)]. The assault brought attention to the value of cybersecurity and the necessity for organizations and people to be attentive in upgrading their systems to counter emerging threats.

**8. Marquard & Bahls supply chain attack 2022 [[→36](#)]:** Two German gasoline trading firms Marquard & Bahls, Oiltanking and Mabanaft, were victims of extensive cyberattacks on January 29, 2022. Due to the assaults' disruption of IT systems and supply lines, Shell and other businesses ultimately had to reroute their shipments. The issue has been described as “serious, but not grave” [[→38](#)] by officials from the Federal Office for Information Security. In a joint statement, Oiltanking and Mabanaft confirmed that they are cooperating to find a solution to the problem. Fortunately, the intrusions have not done much harm and are not expected to be as serious as first thought. The businesses are, however, adopting additional security measures to guarantee that their systems are kept safe and that similar assaults won't occur again in the future [[→37](#)].

**9. AIIMS Delhi cyberattack [[→41](#)]:** On November 23, AIIMS (All India Institute of Medical Sciences), Delhi, encountered a

cyberattack, leading to the paralysis of its servers. The Intelligence Fusion and Strategic Operations (IFSO) unit of the Delhi Police registered a case of extortion and cyberterrorism on November 25. According to a technical analysis conducted by the Indian Computer Emergency Response Team (Cert-In), the recent cyberattack on AIIMS was found to have resulted from the compromise of the government-run hospital's IT networks by "unknown threat actors" due to improper network segmentation [[→39](#)]. The investigation conducted by Cert-In in response to AIIMS reporting the cyberincident unveiled that the attack, which occurred during November and December, caused operational disruption as critical applications ceased to function. The Minister of State for Electronics and Information Technology, Rajeev Chandrasekhar, had previously labeled the attack as a "conspiracy and planned by forces" of significant influence, categorizing it as a sophisticated ransomware attack. The Defence Research and Development Organization (DRDO) now has the responsibility for the All India Institute of Medical Sciences' (AIIMS) cyberdivision. Major structural adjustments have been made for the servers' cybersecurity, which stopped the attack, according to an official source at AIIMS [[→40](#)]. The official said that after the cyberassault occurred in

November, the firewall's rules underwent a significant update that stopped the server hack.

## 5 Cybersecurity

Any business or organization must prioritize cybersecurity because it protects sensitive information about customers and clients from external dangers like rivals or hostile actors. It includes a variety of steps required to maintain the security of computer systems, servers, intranets, and networks. These safeguards must be put in place by cybersecurity specialists to guarantee that only authorized users may access the data. Understanding the many forms of cybersecurity available is crucial for offering the greatest protection. There are several sorts, including intrusion detection systems, firewalls, access control, authentication, and encryption [[→41](#)]. To put the best security measures into place, it is crucial to comprehend how these measures differ from one another because each one has advantages and disadvantages of its own. Security is an important aspect of any organization's operations and a key component of any IT strategy. There are a variety of different types of cybersecurity measures that can be taken to ensure the safety of a company's network, applications, information, and operations.

1) **Network security:** Network security is a combination of technologies, processes, and practices designed to protect an organization's computer networks from malicious attacks, misuse, and disruption. This includes the implementation of antivirus programs, the use of encryption to protect data in transit, firewalls to block access from unauthorized users, and security policies to ensure that all users and devices adhere to the same set of acceptable practices [[→9](#)].

Organizations may employ monitoring tools to detect and investigate any suspicious activities. Network security is essential for any business as it helps to protect sensitive data, maintain customer trust, and prevent costly downtime.

2) **Information security:** Information security is the practice of protecting data from unauthorized access, disclosure, misuse, modification, or destruction. It is a multilayered approach that includes the implementation of strong access control measures, the use of encryption technologies and secure networks, and the adoption of robust policies and procedures for data access and usage [[→42](#)]. It also requires the active participation of all staff members, who must adhere to prescribed security protocols and use technology as appropriate. Information security requires all personnel to be aware of their responsibility in maintaining the security of the organization's data and to take the necessary

steps to ensure that data is kept secure on all devices, networks, and systems. Businesses must have a data protection strategy in place to ensure the protection of their data and the privacy of their customers.

3) **Operational security:** This involves the implementation of technical, administrative, and physical controls to safeguard data from unauthorized access, disclosure, modification, or destruction. Data security measures include encryption, access control, authentication, authorization, backup, and disaster recovery [[→43](#)]. Data security should consider ways to mitigate risk and maximize data privacy, such as data minimization, data governance, and data classification.

4) **Application security:** Application security is a critical component of any successful development project. Software solutions, such as antivirus programs, encryption, and firewalls, should be employed to protect the system against external threats. Hardware solutions, such as network segmentation, can be used to increase security and reduce the chances of unauthorized access to sensitive data or applications. To ensure that risk levels remain low, these hardware and software solutions should be regularly monitored and updated with the latest security protocols [[→44](#)]. It is important to ensure that application developers are trained on best practices regarding security and that they

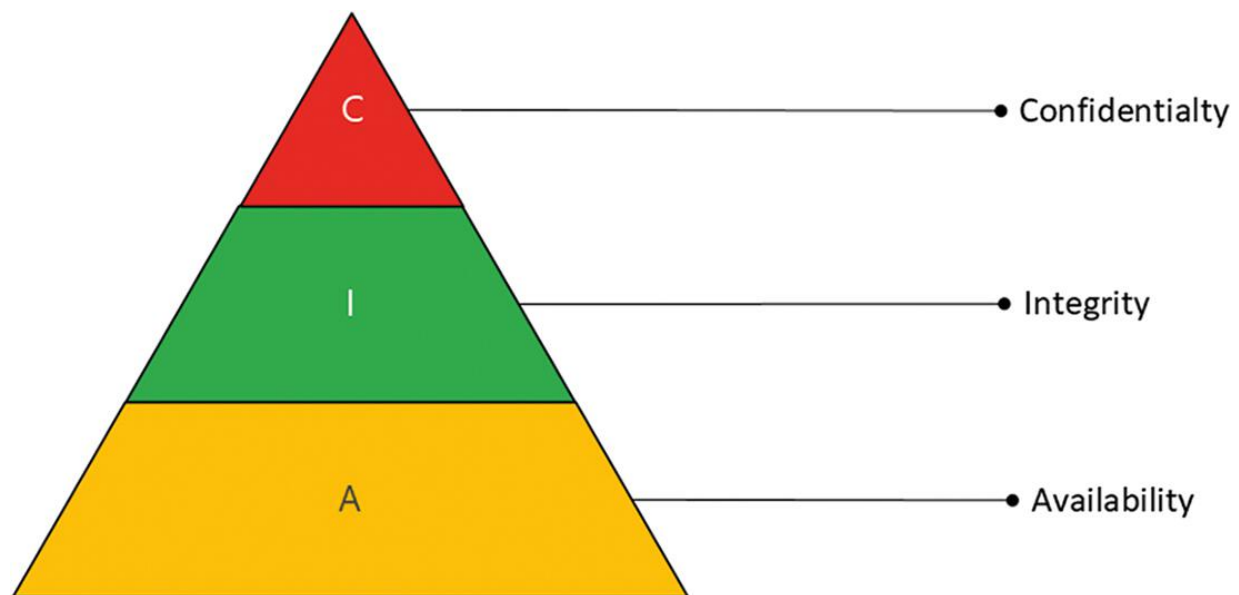
are aware of the ramifications of their actions in the event of a breach.

5) **Cloud security:** Cloud security is an essential component of any business's cybersecurity strategy. By implementing cloud security measures, organizations can protect their data, intellectual property, and other sensitive information from unauthorized access, malicious attacks, and other cyberthreats. Cloud security involves a range of techniques, from encryption and authentication to firewalls and multi-factor authentication [[→45](#)]. It also requires a comprehensive approach that incorporates both technical and nontechnical measures, such as user education, policies, and procedures. With the right cloud security measures in place, organizations can ensure the integrity, availability, and confidentiality of their cloud-based data and applications.

6) **User training:** User training is an essential element of any corporate security plan. It helps to educate employees on how to protect their workplace from potential threats. Training should focus on how to identify and remove suspicious attachments in emails, how to safely interact with external storage devices, and how to ensure that passwords are used correctly and securely [[→46](#)]. User training should also provide an understanding of the consequences of not adhering to these security protocols. The purpose of this

training is to ensure that users are aware of the risks to their system and their data, allowing them to make informed decisions when interacting with their work environment.

Organizations need to take steps to protect their data and systems from cyberattacks. This can be done by implementing policies and procedures that focus on the CIA principles of confidentiality, integrity, and availability ([→Figure 2](#)). Security measures such as encryption, authentication, and access control can help ensure that only authorized personnel can access sensitive data [[→47](#)].



**Figure 2:** CIA triad.

Advanced hackers can bypass basic cybersecurity protection, making it increasingly difficult to protect larger companies. The

increasing number of participants in the virtual world of data exchange presents another challenge to cybersecurity. There is a lack of qualified professionals with specific cybersecurity skills and knowledge, making cybersecurity processes more difficult to implement [[→48](#)]. Given the increasing interconnectedness of the world's major infrastructure, a combined cyberphysical security approach needs to be adopted in order to achieve a comprehensive level of cybersecurity. This approach involves the integration of physical, technical, and social security measures to protect people, resources, and information.

Organizations must ensure that their online systems are secure and free from cyberattacks. This requires custom-made strategies tailored to each business, which should involve strong encryption and a “security perspective.” Companies must also stay ahead of hackers with high-level security systems and services, such as those offered by McAfee, Cisco, and Trend Micro. Decision-makers should consider how cyberattacks could affect their performance and take appropriate preventive measures [[→49](#)]. Security must be constantly monitored and updated to stay ahead of evolving security threats. By investing in the right cybersecurity solutions, organizations can guard their systems, networks, and data against malicious malware and criminals.

## 6 Cybersecurity policy

Cyber has revolutionized the way communities share and access data. It has increased the efficiency of many systems and applications, allowing for faster and more efficient data transfers. This increase in speed and convenience has also led to an increase in cybersecurity risks. Cybersecurity policies are used to mitigate these risks and protect sensitive data.

Cybersecurity policies are rules and regulations implemented to protect data from unauthorized access and malicious attacks.

These policies are tailored to the specific needs of the organization or industry and can include anything from data retention policies to system operation strategies. At a basic level, these policies typically outline access control measures, user authentication requirements, and data encryption techniques [[→50](#)]. The implementation of a comprehensive cybersecurity policy is a key factor in ensuring that digital systems remain secure and protected. Without a policy in place, organizations open themselves up to a variety of risks and vulnerabilities, making it important that all organizations have an established cybersecurity policy in place [[→51](#)].

All domestic residents, foreign business people, and workers in the industry with valid contracts are covered by the national cybersecurity policy. It describes the mode of interpretation and

registration of the policy and lays forth the goals of the pertinent regulatory agency. The national and corporate security goals may not be the same, and there may be differences in the methods used to turn goals into policies and laws [[→52](#)]. Many businesses have a centralized security department that is in charge of developing cybersecurity guidelines, standards, and solutions. When security is of the utmost significance, the components of the common components wing may declare their cyberpolicy. Such policies serve as the foundation for regulations. When two or more policies are put into effect at once, there may occasionally be contradictions. The country's national security policy now fully incorporates the national cyberpolicy [[→53](#)]. Cybersecurity policy is an important aspect of any state's legal framework. It outlines the guidelines and regulations that organizations and individuals must adhere to when using technology and the internet. Although policies are not legally binding, they still provide an important framework for the proper use of technology, as well as for protecting against cyberthreats. Organizations must take the necessary steps to ensure that their cybersecurity policies are properly implemented. This includes having an effective system for monitoring policy compliance, as well as developing the necessary processes and protocols to ensure that any violations are dealt with quickly and

appropriately [[→54](#)]. It is also important for organizations to ensure that their policies are regularly updated and adapted to reflect the changing nature of the digital world.

Organizations should ensure that their policies are communicated to all staff members and stakeholders and that they are aware of their responsibility to comply with the policy. This will help to ensure that all parties are informed of their role in protecting the organization against cyberthreats [[→55](#)]. Organizations must ensure that they have the necessary tools and resources to enforce their cybersecurity policies. Having the right tools and resources in place can help to ensure that the organization is properly protected from cyberthreats.

Organizations are increasingly recognizing the need for senior data security managers to oversee the organization's security situation. Clear divisions of tasks and responsibilities between security experts and nonexperts are essential in order to reduce the risk of data leakage. Organizations must instill the right culture within the organization to ensure that everyone understands their role in ensuring data security. This includes ensuring that people are aware of the data security policies, procedures, and regulations, as well as making sure that everyone understands their role in preventing data theft or potential data breaches. Regular assessments and reviews

should be conducted to ensure that the security measures are up-to-date and that all employees are trained and knowledgeable about the policies and procedures surrounding data security [[→56](#)]. This will help to ensure that all stakeholders have the necessary information to make informed decisions and take the right steps to ensure data security.

## 7 Future direction

As we look toward the future of cybersecurity, it is evident that this field will continue to evolve rapidly, driven by advancements in technology and the ever-increasing sophistication of cyberthreats. One key area that holds great promise is the integration of artificial intelligence (AI) and machine learning (ML) into security systems [[→57](#)]. These technologies have the potential to analyze vast amounts of data in real time, enabling faster threat detection and response. By leveraging AI and ML, organizations can proactively defend against emerging threats and stay one step ahead of cyberattackers. Another crucial aspect of the future of cybersecurity is quantum-resistant cryptography [[→58](#)]. With the rise of quantum computing, traditional cryptographic methods may become vulnerable to attacks. As a result, researchers and industry experts must work together to

develop and implement cryptographic solutions that can withstand the computing power of quantum machines, ensuring the security of digital communication and data [[→59](#)].

The increasing reliance on the Internet of things (IoT) and interconnected devices opens up new avenues for cyberattacks. Future research should focus on securing IoT ecosystems and developing robust protocols to safeguard critical infrastructure and personal data [[→60](#)]. The interplay between cybersecurity and privacy in the context of IoT devices also requires careful examination to strike the right balance between convenience and data protection. The proliferation of cyberphysical systems, such as smart cities and autonomous vehicles, demands comprehensive research on their security implications [[→61](#)]. Understanding the potential consequences of cyberphysical attacks and devising resilient security architectures for these complex systems are critical for ensuring the safety and well-being of society. The study of human factors in cybersecurity is another important direction for future research [[→62](#)]. Understanding human behavior and decision-making in the context of cyberthreats can provide valuable insights into the vulnerabilities that cyberattackers exploit. Research in this area can lead to the development of targeted training programs and awareness campaigns to empower individuals and

organizations to better protect themselves against social engineering and phishing attacks [[→63](#)].

## 8 Conclusion

Computer security is an ever-growing area of concern as the world becomes increasingly interconnected. Networks are used to carry out essential transactions, making cybercrime a real and ongoing threat. In the twenty-first century, cyberspace and related technology constitute a major source of power and influence in international relations, alongside existing governments. This means other actors, such as private companies, terrorist and criminal organizations, and individuals, must now also be considered. When considering the implications of these changes to national security, three distinct elements must be analyzed: security, quality of life, and the lack of geographical boundaries [[→64](#)]. Security is the primary consideration when discussing national security, as opposed to geopolitical or military concerns. The threat posed by cybercrime extends far beyond traditional geographic constraints, making it a global issue. The quality of life of citizens can be negatively impacted by cybercrime, making it a matter of national security [[→65](#)].

Cyberthreats can be difficult to identify and devastating in their effects. They are unpredictable and multifaceted and require cooperation between governments and the private sector in order to be effectively combated. Conventional means of dealing with them, such as the use of military and police force, are not enough. It must also be remembered that individuals and businesses are just as susceptible to cyber risks as governments. As such, international relations must consider the security of the digital age, which is no longer solely a matter of concern for governments. There is a need for a wide range of theoretical approaches that go beyond the traditional state-centric view of international relations if we are to successfully confront the threat of cyber risks.

## References

[1] T. Balon and I. Baggili, "Cybercompetitions: A Survey of Competitions, Tools, and Systems to Support Cybersecurity Education," *Education and Information Technologies*, 2023.

[→https://doi.org/10.1007/s10639-022-11451-4](https://doi.org/10.1007/s10639-022-11451-4) →

[2] B. E. Gavett, R. Zhao, S. E. John, C. A. Bussell, J. R. Roberts, et al., "Phishing Suspiciousness in Older and Younger Adults: The Role of Executive Functioning," *PLOS ONE*, vol. 12, no. 2, pp.

e0171620, 2017. →<https://doi.org/10.1371/journal.pone.0171620>  
→

**[3]** V. Chang, L. Golightly, Q. A. Xu, T. Boonmee, and B. S. Liu, "Cybersecurity for Children: An Investigation into the Application of Social Media," Enterprise Information Systems, 2023. doi:: 10.1080/17517575.2023.2188122. →

**[4]** O. Abiodun, O. Omolara, M. Alawida, R. Alkhawaldeh, and H. Arshad, "A Review on the Security of the Internet of Things: Challenges and Solutions," Wireless Personal Communications, vol. 119, pp. 1–35, 2021. doi:: 10.1007/s11277-021-08348-9. →

**[5]** M. Alawida, A. E. Omolara, O. I. Abiodun, and M. Al-Rajab, "A Deeper Look into Cybersecurity Issues in the Wake of Covid-19: A Survey," Journal of King Saud University – Computer and Information Sciences, vol. 34, no. 10, pp. 8176–8206, 2022 Nov. doi:: 10.1016/j.jksuci.2022.08.003. Epub 2022 Aug 11. PMCID: PMC9367180. →

**[6]** H. Arshad, E. Omlara, I. Abiodun, and A. Aminu, "A Semi-Automated Forensic Investigation Model for Online Social Networks," Computers and Security, vol. 97, pp. 101946, 2020. doi:: 10.1016/j.cose.2020.101946. [a](#), [b](#)

**[7]** H. Arshad, S. Abdullah, M. Alawida, A. Alabdulatif, O. I. Abiodun, and O. Riaz, "A Multi-Layer Semantic Approach for Digital Forensics Automation for Online Social Networks," *Sensors*, vol. 22, pp. 1115, 2022.

→<https://doi.org/10.3390/s22031115> →

**[8]** "Internet Crime Complaint Center Releases 2022 Statistics," Federal Bureau of Investigation, 22 Mar. 2023.

→<https://www.fbi.gov/contact-us/field-offices/springfield/news/internet-crime-complaint-center-releases-2022-statistics> →

**[9]** "Cybersecurity in Ukraine: National Strategy and International Cooperation," Thegfce.org.

→<https://thegfce.org/cybersecurity-in-ukraine-national-strategy-and-international-cooperation/>. Accessed 4 Aug. 2023. [a](#), [b](#), [c](#)

**[10]** S. Z. Sajal, I. Jahan, and K. E. Nygard, "A Survey on Cyber Security Threats and Challenges in Modern Society," in 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, 2019, pp. 525–528. doi: 10.1109/EIT.2019.8833829. [a](#), [b](#)

**[11]** D. Yadav, D. Gupta, D. Singh, D. Kumar, and U. Sharma, "Vulnerabilities and Security of Web Applications," in 2018 4th

International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 2018, pp. 1–5. doi: 10.1109/CCAA.2018.8777558. →

**[12]** T. M. Mbelli and B. Dwolatzky, “Cyber Security, a Threat to Cyber Banking in South Africa: An Approach to Network and Application Security,” in 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), Beijing, China, 2016, pp. 1–6. doi: 10.1109/CSCloud.2016.18. →

**[13]** M. Antunes, M. Maximiano, R. Gomes, and D. Pinto, “Information Security and Cybersecurity Management: A Case Study with SMEs in Portugal,” Journal of Cybersecurity and Privacy, vol. 1, pp. 219–238, 2021.

→<https://doi.org/10.3390/jcp1020012> →

**[14]** E. Babulak, J. Hyatt, K. K. Seok, and J. S. Ju, “COVID-19 & Cyber Security Challenges US, Canada & Korea (October 27, 2020),” Transactions on Machine Learning and Data Mining, vol. 13, no. 1, pp. 43–59, 2020. © 2020, ibai-publishing, P-ISSN: 1865-6781, E-ISSN 2509-9337 ISBN 978–3-942952-78–1, Available at SSRN: →<https://ssrn.com/abstract=3904325> →

**[15]** V. Balas, R. Kumar, and R. Srivastavs, Recent Trends and Advances in Artificial Intelligence and Internet of Things, 2020.

doi:: 10.1007/978-3-030-32644-9. ➔

**[16]** A. Bossler, "Neutralizing Cyber Attacks: Techniques of Neutralization and Willingness to Commit Cyber Attacks," American Journal of Criminal Justice, 2021. doi: 10.1007/s12103-021-09654-5. ➔

**[17]** A. J. Burns, M. Johnson, and D. Caputo, "Spear Phishing in a Barrel: Insights from a Targeted Phishing Campaign," Journal of Organizational Computing and Electronic Commerce, vol. 29, pp. 24–39, 2019. doi: 10.1080/10919392.2019.1552745. ➔

**[18]** Cbsnews, 2021. ➔<https://www.cbsnews.com/news/us-covid-relief-hacking-hackers-arrested-indonesia-aid-program-scam/>. Cressey, D. R. (1953). Other people's money; a study of the social psychology of embezzlement. [a](#), [b](#)

**[19]** "JBS: Cyber-Attack Hits World's Largest Meat Supplier," BBC News, 2 June 2021. [Online]. Available: ➔[www.bbc.com/news/world-us-canada-57318965](http://www.bbc.com/news/world-us-canada-57318965). Accessed 29 June 2023. [a](#), [b](#)

**[20]** Person, "Cyber-Attack Hits Jbs World's Largest Meat Supplier," Food Digital, 2 June 2021. ➔[www.fooddigital.com/food/cyber-attack-hits-jbs-worlds-largest-meat-supplier](http://www.fooddigital.com/food/cyber-attack-hits-jbs-worlds-largest-meat-supplier) ➔

[21] K. Lyons, "Robinhood Says a Hacker Who Tried to Extort the Company Got Access to Data for 7 Million Customers," The Verge, 8 Nov. 2021.

→<https://www.theverge.com/2021/11/8/22770861/robinhood-7-million-customers-hacker-breach-extortion-security> a, b

[22] J. Nelson, "Hackers Take over Robinhood Twitter Account to Promote Scam," Decrypt, 25 Jan. 2023.

→<https://decrypt.co/119985/hackers-take-over-robinhood-twitter-account-to-promote-scam> →

[23] Z. Whittaker, "Robinhood Says Millions of Customer Names and Email Addresses Taken in Data Breach," TechCrunch, Nov. 2021. →<https://techcrunch.com/2021/11/09/robinhood-data-breach/> →

[24] L. H. Newman, "The Uber Hack's Devastation Is Just Starting to Reveal Itself," Wired, Sept. 2022.

→<https://www.wired.com/story/uber-hack-mfa-phishing/> →

[25] D. Strom, "How Uber Was Hacked – Again,"

Avast.com. →<https://blog.avast.com/uber-hack>. Accessed 28 June 2023. →

[26] "What the Uber Hack Can Teach Us about Navigating IT Security," BleepingComputer.

→<https://www.bleepingcomputer.com/news/security/what-the-uber-hack-can-teach-us-about-navigating-it-security/> a, b

**[27]** R. Collier, "NHS Ransomware Attack Spreads Worldwide," Journal de l'Association Medicale Canadienne [Canadian Medical Association Journal], vol. 189, no. 22, pp. E786–E787, 2017.

→<https://doi.org/10.1503/cmaj.1095434> →

**[28]** P. Arntz, "Nvidia, the Ransomware Breach with Some Plot Twists," in Malwarebytes, 3 Mar. 2022.

→<https://www.malwarebytes.com/blog/news/2022/03/nvidia-the-ransomware-breach-with-some-plot-twists> →

**[29]** I. Ilascu, "NVIDIA Confirms Data Was Stolen in Recent Cyberattack," BleepingComputer, 1 Mar. 2022.

→<https://www.bleepingcomputer.com/news/security/nvidia-confirms-data-was-stolen-in-recent-cyberattack/> a, b

**[30]** J. Allen, "Conti Costa Rica Ransomware Attack Explained," PurpleSec, 10 July 2022. →[https://purplesec.us/security-](https://purplesec.us/security-insights/conti-ransomware-attack)

[insights/conti-ransomware-attack](https://purplesec.us/security-insights/conti-ransomware-attack) →

**[31]** "Wannacry Ransomware Attack," Wikipedia, The Free Encyclopedia, 23 June 2023.

→[https://en.wikipedia.org/w/index.php?title=WannaCry\\_ransomware\\_attack&oldid=1161572147](https://en.wikipedia.org/w/index.php?title=WannaCry_ransomware_attack&oldid=1161572147) →

**[32]** E. Gately, "Marquard&bahls Attack," in Channel Futures.  
→<https://www.channelfutures.com/slides/gar7-jpg>. Accessed 1  
July 2023. →

**[33]** J. Karabus, "Cyberattacker Hits German Service Station  
Petrol Terminal Provider," The Register, 1 Feb. 2022.  
→<https://www.theregister.com/2022/02/01/oiltrading/> →

**[34]** "UPDATE 4-Shell Re-Routes Oil Supplies after Cyberattack on  
German Firm," Reuters, 1 Feb. 2022.  
→[https://www.reuters.com/article/germany-cyber-shell-](https://www.reuters.com/article/germany-cyber-shell-idCNL1N2UC0QD)  
[idCNL1N2UC0QD](https://www.reuters.com/article/germany-cyber-shell-idCNL1N2UC0QD) →

**[35]** E. T. Ciso, "AIIMS Ransomware Attack: What It Means for  
Health Data Privacy," ETCISO, 27 Dec. 2022.  
→[https://ciso.economictimes.indiatimes.com/news/aiims-](https://ciso.economictimes.indiatimes.com/news/aiims-ransomware-attack-what-it-means-for-health-data-privacy/96538957)  
[ransomware-attack-what-it-means-for-health-data-](https://ciso.economictimes.indiatimes.com/news/aiims-ransomware-attack-what-it-means-for-health-data-privacy/96538957)  
[privacy/96538957](https://ciso.economictimes.indiatimes.com/news/aiims-ransomware-attack-what-it-means-for-health-data-privacy/96538957) →

**[36]** K. P. Singh, "Ransomware Attack: Report Flags Host of  
Security Lapses at AIIMS," The Hindustan Times, 7 Dec. 2022.  
→[https://www.hindustantimes.com/cities/delhi-news/report-](https://www.hindustantimes.com/cities/delhi-news/report-flags-host-of-security-lapses-at-aiims-101670359251255.html)  
[flags-host-of-security-lapses-at-aiims-101670359251255.html](https://www.hindustantimes.com/cities/delhi-news/report-flags-host-of-security-lapses-at-aiims-101670359251255.html) →

**[37]** The Hindu Bureau, "1.3 TB Data Encrypted and Five Servers  
Affected in AIIMS Ransomware Attack: Centre," Thehindu.com,

16 Dec. 2022. →<https://www.thehindu.com/news/national/13-tb-data-encrypted-in-ransomware-attack-on-aiims-by-unknown-threat-actors-centre/article66271226.ece>

[38] S. Kumar, "Cybercrime: Rising Concern to Cyber World," Security Boulevard, 17 Jan. 2022. →<https://securityboulevard.com/2022/01/cybercrime-rising-concern-to-cyber-world/> →

[39] C. Cross and M. Kelly, "The Problem of 'White Noise': Examining Current Prevention Approaches to Online Fraud," Journal of Financial Crime, vol. 23, pp. 806–818, 2016. doi: 10.1108/JFC-12-2015-0069. →

[40] S. Das, "The Cyber Security Ecosystem: Post-global Financial Crisis," in Proceedings of the International Conference on Cybersecurity, Cyber Crime and Cyber Forensics, 2015, pp. 343–354. doi: 10.1007/978-81-322-1979-8\_36. →

[41] "IBM Quantum Computing," IBM Quantum Experience, 1 Oct. 2015. →[https://www.ibm.com/quantum/quantum-safe?utm\\_content=SRCWW&p1=Search&p4=43700076610856185&p5=p&gclid=Cj0KCQjw2eilBhCCARIsAG0Pf8vHIjQ- kk8Oq1p4K3JXQ0yeF\\_dWKaDtNmbzIq2hWFUzzgRspXSR8n8aAnBxEALw\\_wcB&gclsrc=aw.ds\\_a,b,c](https://www.ibm.com/quantum/quantum-safe?utm_content=SRCWW&p1=Search&p4=43700076610856185&p5=p&gclid=Cj0KCQjw2eilBhCCARIsAG0Pf8vHIjQ- kk8Oq1p4K3JXQ0yeF_dWKaDtNmbzIq2hWFUzzgRspXSR8n8aAnBxEALw_wcB&gclsrc=aw.ds_a,b,c)

**[42]** A. Esther Omolara, et al., "Honey Details: A Prototype for Ensuring Patient's Information Privacy and Thwarting Electronic Health Record Threats based on Decoys," Health Informatics Journal, vol. 26, no. 3, pp. 2083–2104, 2020. doi: 10.1177/1460458219894479. ➔

**[43]** "Cybersecurity Insiders," IBM Security, 2021.  
➔<https://info.greathorn.com/hubfs/Reports/2021-Business-Email-Compromise-Report-GreatHorn.pdf> ➔

**[44]** S. Hasham, et al., "Financial Crime and Fraud in the Age of Cybersecurity," McKinsey & Company, 1 Oct. 2019.  
➔<https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/financial-crime-and-fraud-in-the-age-of-cybersecurity> ➔

**[45]** M. Hill, "HMRC Shuts Down Almost 300 COVID19 Phishing Scam Sites," Infosecurity Magazine, 2020.  
➔<https://www.infosecuritymagazine.com/news/hmrc-covid19-phishing-scams/> ➔

**[46]** M. S. Hossain, G. Muhammad, and N. Guizani, "Explainable AI and Mass Surveillance System-based Healthcare Framework to Combat COVID-19 like Pandemics," IEEE Network, vol. 34, pp. 126–132, 2020. ➔

**[47]** “INTERPOL Report Shows Alarming Rate of Cyberattacks during COVID-19,” INTERPOL.

→<https://www.interpol.int/en/News-and-Events/News/2020/INTERPOL-report-shows-alarming-rate-of-cyberattacks-during-COVID-19>. Accessed 2 July 2023. →

**[48]** M. S. Jalali, A. Landman, and W. Gordon, “Telemedicine, Privacy, and Information Security in the Age of COVID-19,” Journal of the American Medical Informatics Association, vol. 28, no. 1, pp. 10–12, 2020. doi:: 10.1093/jamia/ocaa310. →

**[49]** M. Vergelis, “Coronavirus Phishing,” Kaspersky, 7 Feb. 2020. →<https://www.kaspersky.com/blog/coronavirus-phishing/32395/> →

**[50]** N. Khan, S. Brohi, and N. Jhanjhi, “Ten Deadly Cyber Security Threats Amid COVID-19 Pandemic,” 2020. →<https://doi.org/10.36227/techrxiv.12278792.v1> →

**[51]** A. Khurshid, “The Crisis of Trust in COVID-19 Pandemic: Can Blockchain Technology Help?,” JMIR Medical Informatics, vol. 8, no. 1, 2020. →<https://doi.org/10.2196/20477> →

**[52]** H. Lallie, L. Shepherd, J. Nurse, A. Erola, G. Epiphaniou, C. Maple, and X. Bellekens, “Cyber Security in the Age of COVID-19: A Timeline and Analysis of Cyber-Crime and Cyber-Attacks

during the Pandemic,” *Computers and Security*, vol. 105, pp. 102248, 2021. doi:: 10.1016/j.cose.2021.102248. ➡

**[53]** S. Mansfield-Devine, “The Growth and Evolution of DDoS,” *Network Security*, pp. 13–20, 2015. doi:: 10.1016/S1353-4858(15)30092-1. ➡

**[54]** A. Manzalini, “Quantum Communications in Future Networks and Services,” *Quantum Reports*, vol. 2, pp. 221–232, 2020. doi: 10.3390/quantum2010014. ➡

**[55]** R. König, A. Seifert, and M. Doh, “Internet Use among Older Europeans: An Analysis based on SHARE Data,” *University Access to Information Society*, vol. 17, pp. 621–633, 2018. doi: 10.1007/s10209-018-0609-5. ➡

**[56]** Z. Kotulski, et al., “Towards Constructive Approach to End-to-end Slice Isolation in 5G Networks,” *EURASIP Journal on Information Security*, vol. 2018, no. 2, 2018. doi: 10.1186/s13635-018-0072-0. ➡

**[57]** P. Parrend, et al., “Foundations and Applications of Artificial Intelligence for Zero-day and Multi-step Attack Detection,” *EURASIP Journal on Information Security*, vol. 2018, no. 4, 2018. doi: 10.1186/s13635-018-0074-y. ➡

**[58]** Journal on Information Security, vol. 2018, no. 5, 2018. doi: 10.1186/s13635-018-0076-9. ➡

**[59]** J. Navarro, et al., "OMMA: Open Architecture for Operator-guided Monitoring of Multi-step Attacks," EURASIP Journal on Information Security, vol. 2018, no. 6, 2018. doi: 10.1186/s13635-018-0075-x. ➡

**[60]** G. Jaideep and B. P. Battula, "Detection of Spoofed and Non-spoofed DDoS Attacks and Discriminating Them from Flash Crowds," EURASIP Journal on Information Security, vol. 2018, no. 9, 2018. doi: 10.1186/s13635-018-0079-6. ➡

**[61]** A. Gurung and M. Raja, "Online Privacy and Security Concerns of Consumers," Information and Computer Security, vol. 24, pp. 348–371, 2016. doi: 10.1108/ICS-05-2015-0020. ➡

**[62]** M. Saqib, "The Challenges and Security Issues Faced by E-Commerce in India: A Review," International Journal of Computer Sciences and Engineering, vol. 6, pp. 447–449, 2018. doi: 10.26438/ijcse/v6i3.447449. ➡

**[63]** J. Ricci, F. Breitingner, and I. Baggili, "Survey Results on Adults and Cybersecurity Education," Education and Information Technologies, vol. 24, 2019. doi: 10.1007/s10639-018-9765-8. ➡

**[64]** K. K. Raymond Choo, "The Cyber Threat Landscape: Challenges and Future Research Directions," *Computers and Security*, vol. 30, pp. 719–731, 2011. doi: 10.1016/j.cose.2011.08.004. ➡

**[65]** S. Mir and S. Quadri, "Information Availability: An Insight into the Most Important Attribute of Information Security," *Journal of Information Security*, vol. 7, pp. 185–194, 2016. doi: 10.4236/jis.2016.73014. ➡

# An energy-efficient FPGA-based implementation of AES algorithm using HSTL IO standards for new digital age technologies

**Chandrashekhar Patel**

**Bhanu Priya Yadav**

**Aditi Saxena**

## **Abstract**

Our daily activities in the new digital age are heavily reliant on the internet. Online communication is used for many purposes, including leisure, commerce, and work-related activities. It suggests that a significant amount of data being exchanged online. Therefore, the researcher is concentrating on security to stop threats and online vulnerabilities. To make digital communication quick and secure, the authors used an energy-efficient FPGA-based implementation of the AES algorithm. The XPA (XPower Analyzer) tool is used to analyze total power usage throughout the experimental activity, and the Vivado IDE tool is utilized for simulation using the Verilog language. To calculate the overall amount of power consumed with voltages ranging from 0.95 to 1.20 V, voltage scaling is applied. The

average total power consumption of the IO standard family HSTL\_I, according to the researcher's analysis, is 0.0938 W that is less than that of the other IO standard families, making it the most appropriate for our digital component.

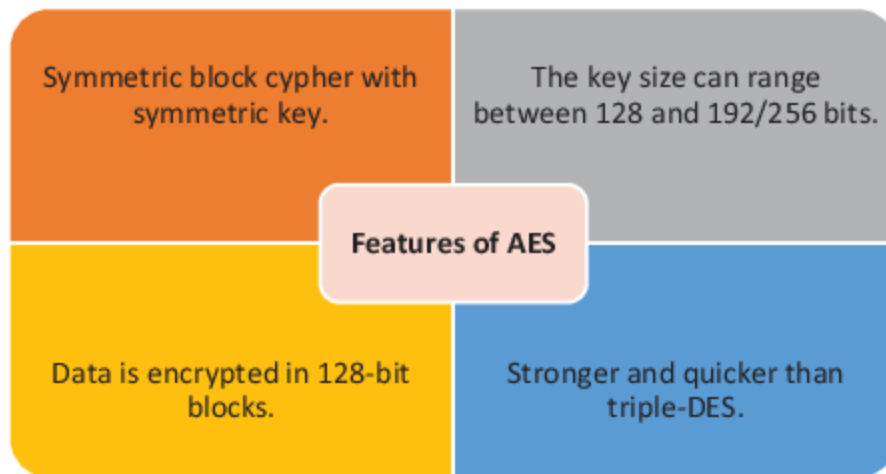
**Keywords:** AES, FPGA, HSTL, Verilog, Vivado, DES,

# 1 Introduction

## 1.1 The Advanced Encryption Standard (AES)

AES was developed by the United States National Institute of Standards and Technology in 2001. It is a standard for electronic data encryption. We have more standards for data encryption available like DES and triple DES, but AES has gained popularity over them in spite of being difficult to implement because it is much stronger. When compared to triple DES, it was found that AES is six times faster than triple DES. With the tiny key size of DES, AES is the best replacement for it. The feature of AES is shown in [→Figure 1](#).

### 1. *Features of AES*



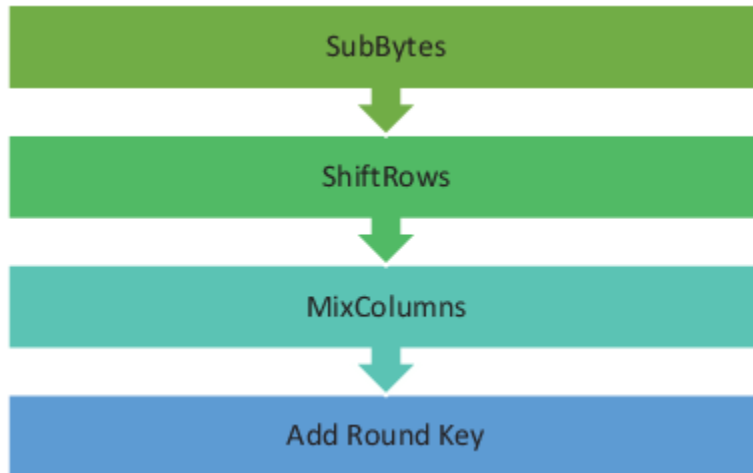
**Figure 1:** Features of AES.

### 1. *Creation of round keys*

A key schedule method is used to compute all of the round keys from the key. As a result, the initial key is utilized to generate several other round keys that will be used in the matching round of encryption.

### 1. *Encryption*

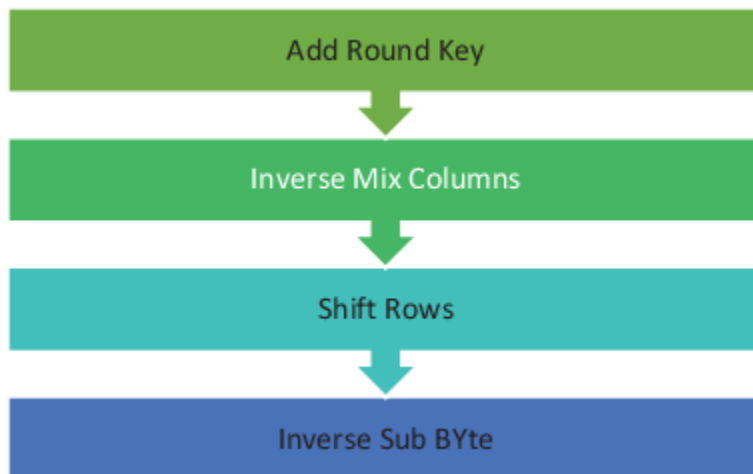
Each block is treated by AES as a 16 byte ( $4 \text{ byte} \times 4 \text{ byte} = 128$ ) grid in a column-major configuration. Each round comprises four steps. The stages of each round of encryption are shown in →Figure 2.



**Figure 2:** Stages of each round in encryption.

### 1. *Decryption*

The technique of AES cipher text decryption is identical to the encryption process but just in the reverse order. Four procedures are performed in each cycle. The stages of each round of decryption are shown in [→Figure 3](#).

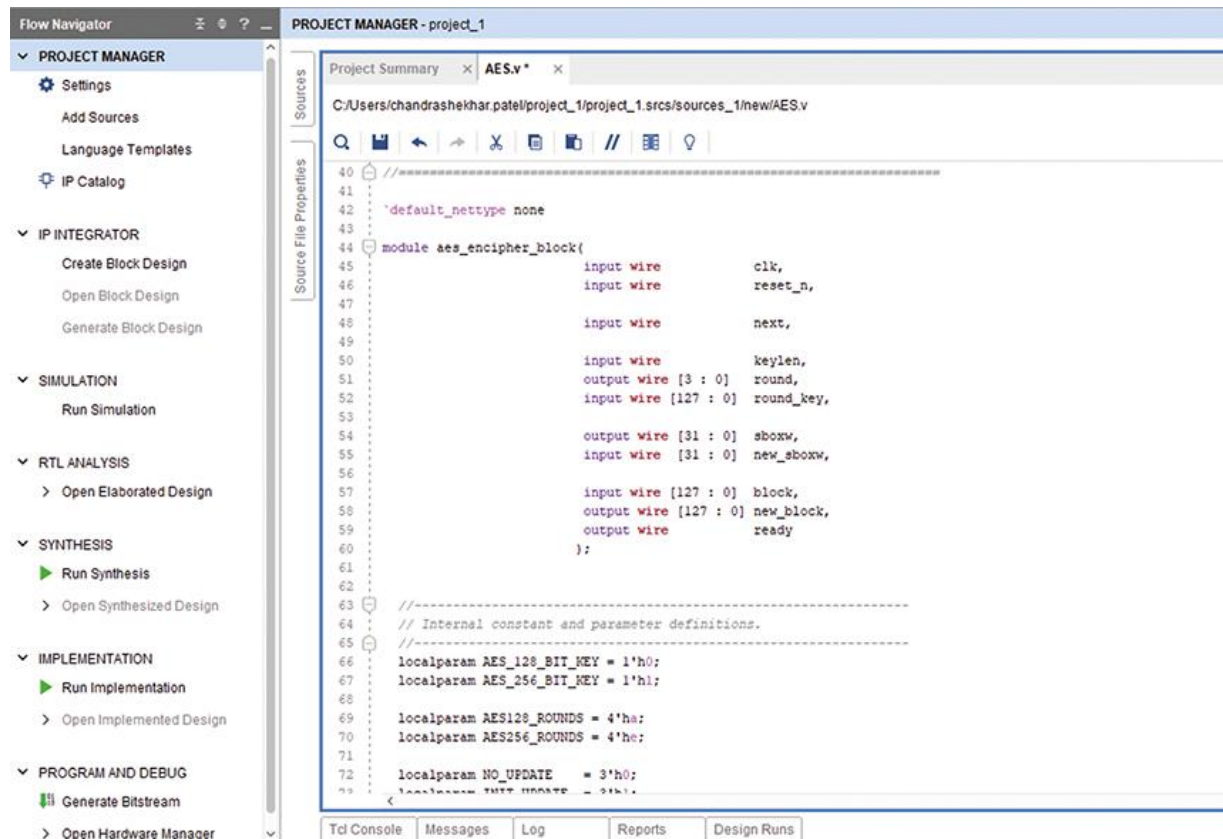


**Figure 3:** Stages of each round in decryption.

## 1.2 FPGA

FPGA stands for field programmable gate array. They are semiconductor devices. The property that makes FPGA different from ASICs (Application Certain Integrated Circuits), which are designed for specific purposes, is that we can reprogram FPGA to meet specific applications and needs. We have simple technologies like programmable read-only memory (PROM) and programmable logic devices (PLDs) and FPGA is an evolved version of them.

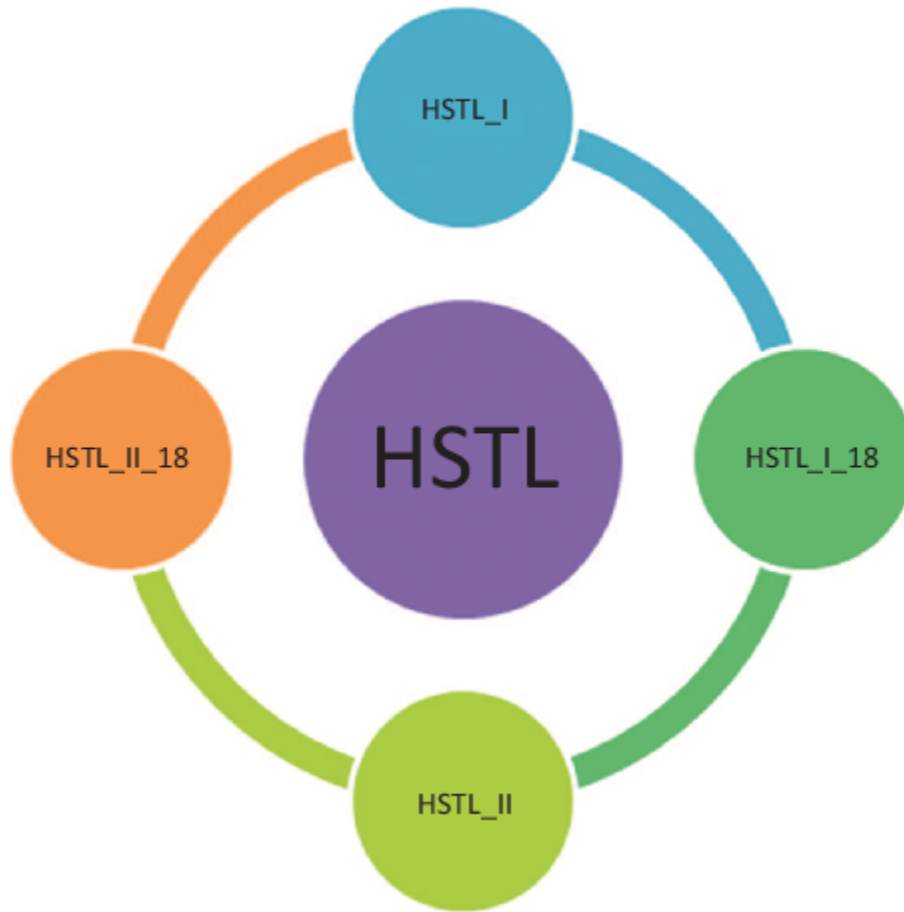
For the experimental work, Vivado IDE tool used for designing purpose using Verilog language and XPA (XPower Analyzer) tool was used for analyzing total power consumption. →[Figure 4](#) shows the AES\_encipher\_block Code using the Vivado IDE tool.



**Figure 4:** AES\_encipher\_blockCode.

### 1.3 HSTL (high-speed transceiver logic) IO standard

JEDEC (Joint Electron Device Engineering Council, associated with the Electronic Industry Association EIA) established high-speed transceiver logic or HSTL IO in 1995. They established four classes: Class I, Class II, Class III, and Class IV. Seven series FPGA I/O supports Class I and Class II. This version of HSTL supports Class I for 1.2 version, as well as Class II for the 1.5 V and 1.8 V versions. →[Figure 5](#) shows different IO families of HSTL.



**Figure 5:** HSTL IO standard family.

## 2 Literature review

The rapid growth of digital technologies in recent years has necessitated the need for energy-efficient cryptographic algorithms to secure sensitive data [[→1](#)]. The Advanced Encryption Standard (AES) has emerged as one of the most widely used symmetric encryption algorithms due to its robust security and widespread adoption [[→2](#)]. However, the demand for faster and more energy-efficient implementations of AES

has led to research focusing on hardware acceleration using FPGAs and the integration of modern input/output (IO) standards, such as HSTL, to cater to the requirements of the new digital age technologies [[→3](#), [→4](#)].

Research efforts have been directed toward optimizing the hardware implementation of AES [[→5](#), [→6](#)]. Various techniques, such as parallel processing, pipelining, and resource sharing, have been explored to accelerate the encryption and decryption processes [[→7](#), [→8](#)]. FPGAs offer a customizable hardware platform for implementing AES efficiently, exploiting parallelism and tailored optimizations to achieve high throughput while managing energy consumption [[→9](#), [→10](#)].

FPGAs provide an excellent platform for hardware acceleration due to their reconfigurable nature. Several studies have proposed FPGA-based AES implementations that aim to strike a balance between speed and energy efficiency [[→11](#)].

Techniques, such as loop unrolling, key expansion optimization, and efficient data path design, have been explored to enhance the performance of AES on FPGAs [[→12](#)].

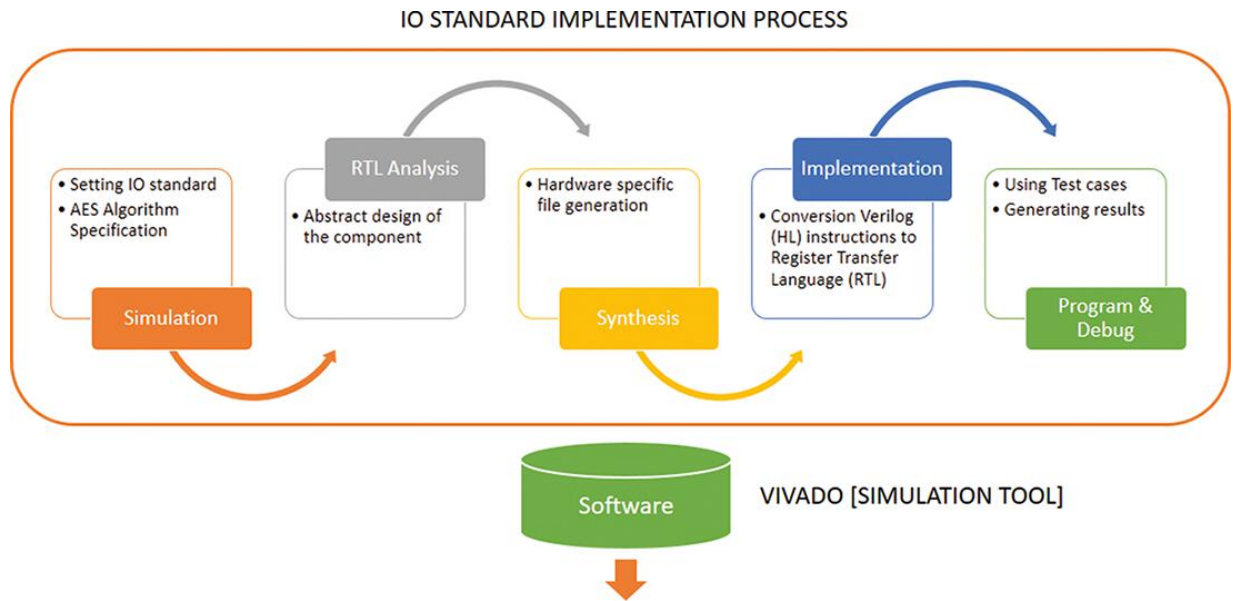
Energy efficiency is a critical concern in modern digital systems. Researchers have explored techniques like dynamic voltage and frequency scaling, clock gating, and power gating to

reduce the energy consumption of FPGA-based AES implementations [[→13](#), [→14](#)]. These techniques ensure that the FPGA operates at the minimum required power levels without compromising performance [[→15](#), [→16](#)].

HSTL IO standards provide fast signal transmission and reduced signal noise, making them suitable for high-speed data interfaces in modern technologies [[→17](#), [→18](#)]. Integrating HSTL IO standards into FPGA-based AES implementations can enhance data throughput while minimizing signal integrity issues [[→19](#), [→20](#)]. This integration requires careful consideration of the FPGA's IO capabilities and the design of an interface that can efficiently manage the data transfer between the AES core and external memory or communication interfaces [[→21](#), [→22](#)].

### **3 Design methodology**

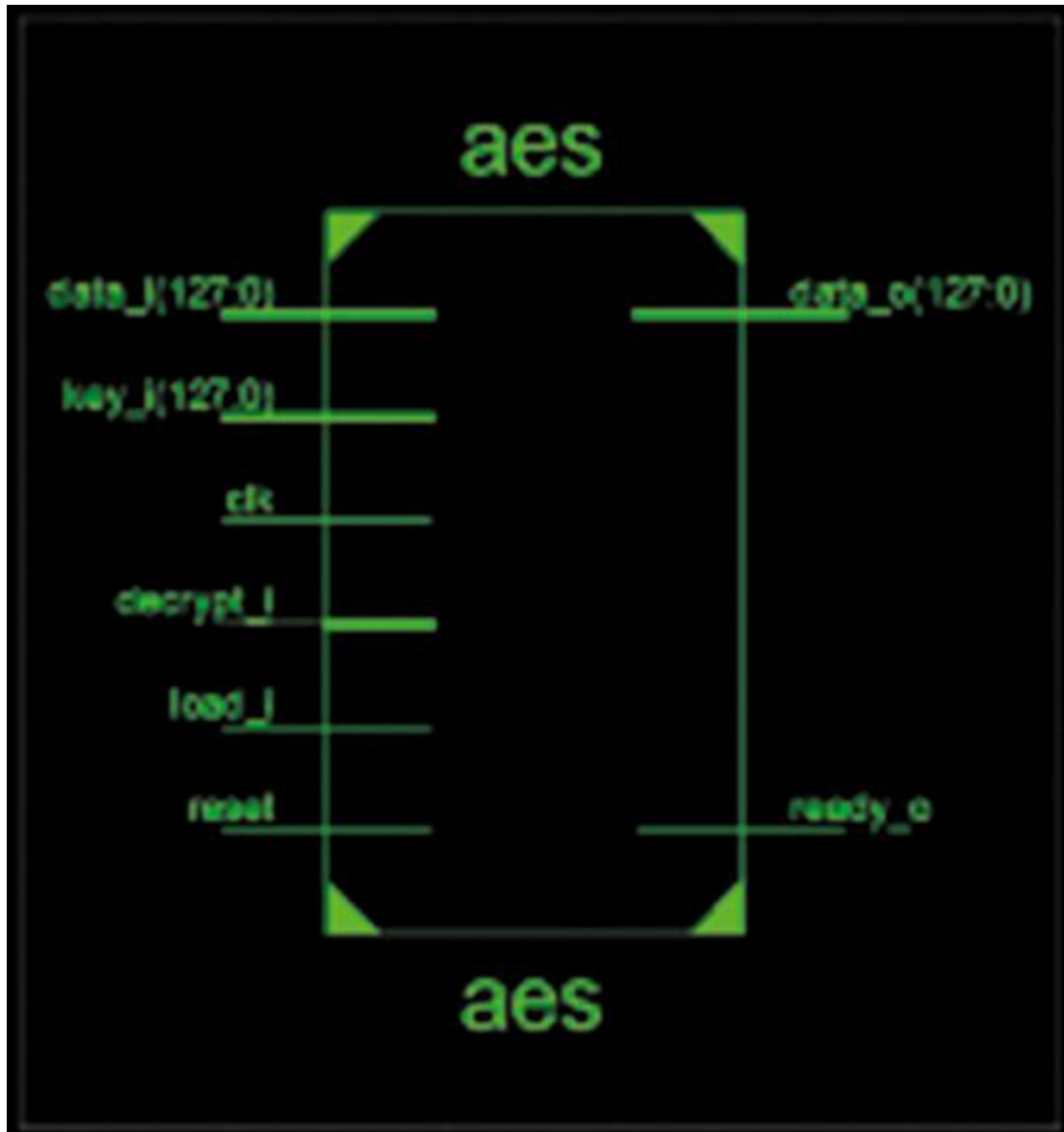
[→Figure 6](#) shows the design methodology of FPGA-based AES algorithm [[→12](#)]. The researcher briefly explains the implementation steps of FPGA-based AES algorithm [[→13](#)].



**Figure 6:** Design methodology of FPGA-based AES algorithm.

### 3.1 Simulation

FPGA simulation is a technique used to model and test the behavior and performance of FPGA designs before they are synthesized and implemented on actual FPGA hardware [→14]. It allows designers to verify the functionality of their designs, analyze their performance, and debug any issues prior to the physical implementation [→15]. →Figure 7 shows the top level of schematic of FPGA-based AES algorithm.

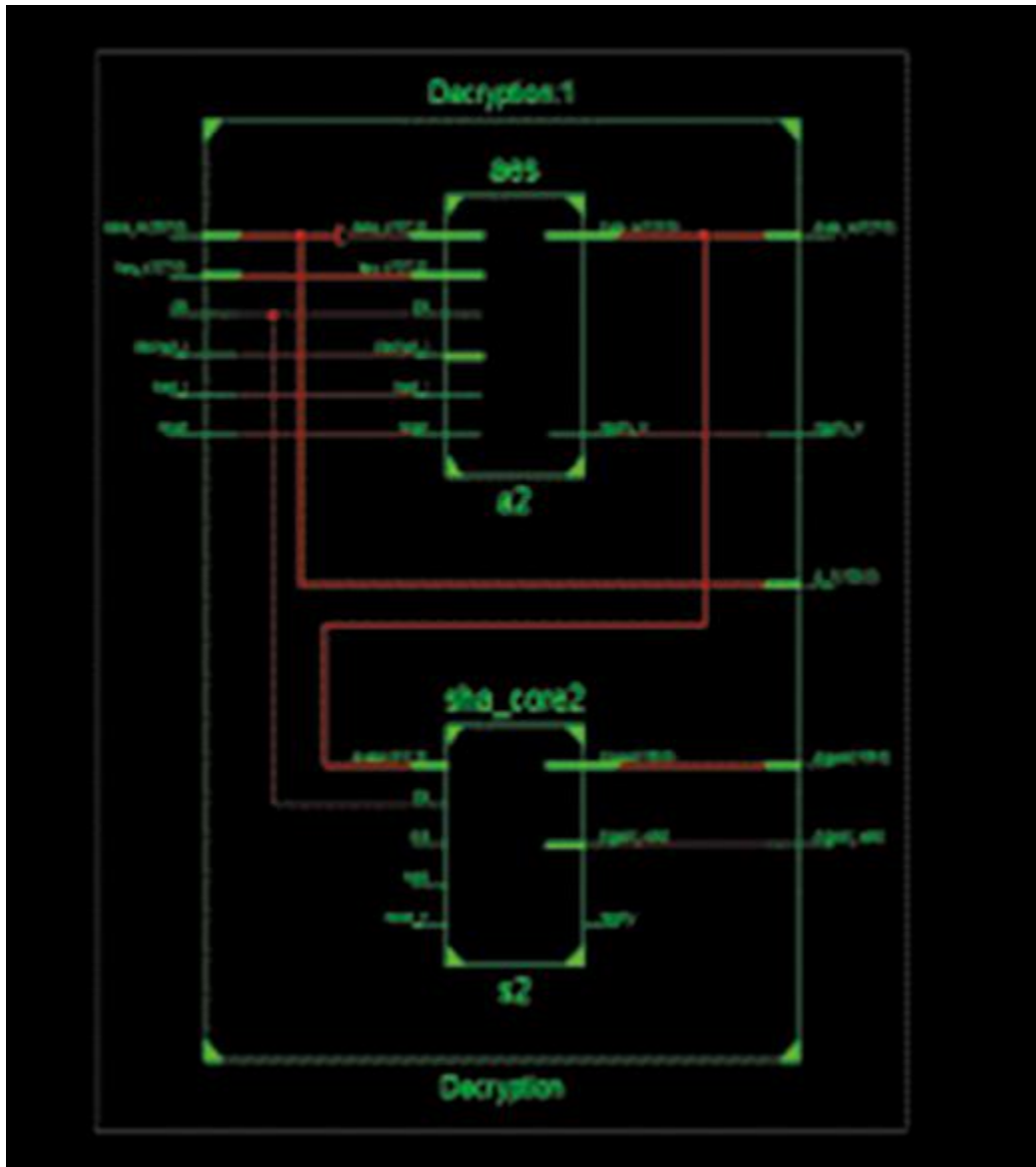


**Figure 7:** Top level of schematic of FPGA-based AES algorithm.

### 3.2 RTL analysis

RTL analysis, also known as “register transfer level analysis,” is a process used in digital design to verify the correctness and functionality of a design at the register transfer level [[→16](#)]. It involves analyzing the behavior and interactions of the various

registers, data paths [[→17](#)], and control logic within a digital system. RTL schematic of FPGA-based AES algorithm is shown in [→Figure 8](#).



**Figure 8:** RTL schematic of FPGA-based AES algorithm.

### 3.3 Synthesis

Synthesis refers to the process of transforming a high-level hardware description language (HDL) description, such as VHDL or Verilog [[→18](#)], into a lower-level gate-level representation that can be implemented on the FPGA hardware [[→19](#), [→20](#)].

### **3.4 Implementation**

The implementation phase refers to the process of translating the synthesized design into a physical implementation on the target FPGA device [[→19](#)]. This phase involves several steps to configure the FPGA with the desired functionality [[→21](#), [→22](#), [→23](#)].

### **3.5 Program and debug**

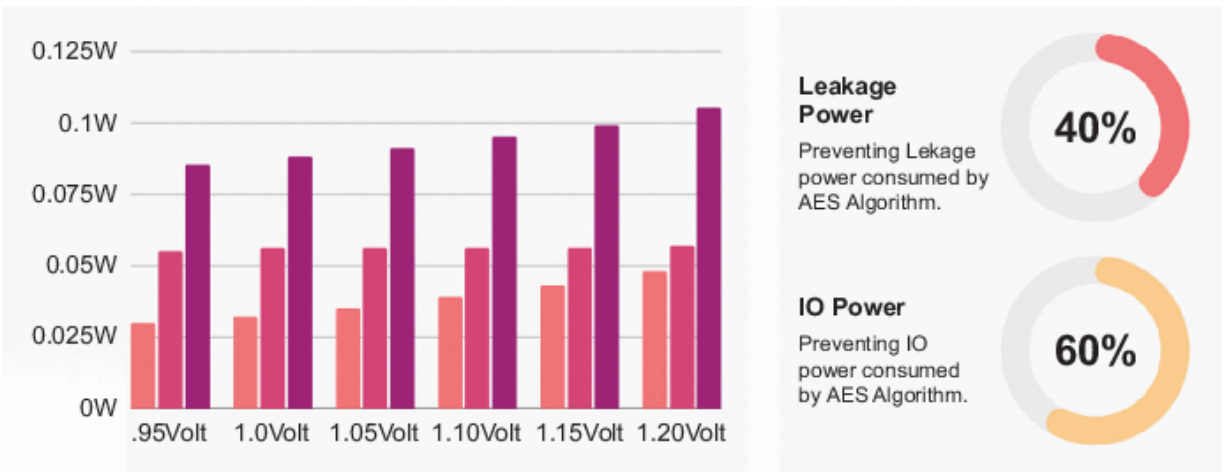
It involves the processes of loading the synthesized design onto the FPGA device and verifying its functionality, as well as identifying and fixing any issues that may arise.

## **4 Results and analysis**

**Table 1:** Power consumption using HSTL\_I.

Voltage	Leakage power	IO power	Total power
0.95	0.030 W	0.055 W	0.085 W
1.0	0.032 W	0.056 W	0.088 W
1.05	0.035 W	0.056 W	0.091 W
1.10	0.039 W	0.056 W	0.095 W
1.15	0.043 W	0.056 W	0.099 W
1.20	0.048 W	0.057 W	0.105 W

In [→Table 1](#), we are applying IO standard HSTL\_I. In this analysis process, voltages vary from 0.95 to 1.20 V range. Total power utilization is considered as the sum of the device's static power and dynamic power. During the experiment dynamic power (leakage power) consumes 40% of the total power. The average value of total power consumption is 0.093 W by varying the voltage from 0.095 to 1.20 V. Analysis of HSTL\_I is also represented by using a bar graph in [→Figure 9](#).



**Figure 9:** Power consumption using HSTL\_I.

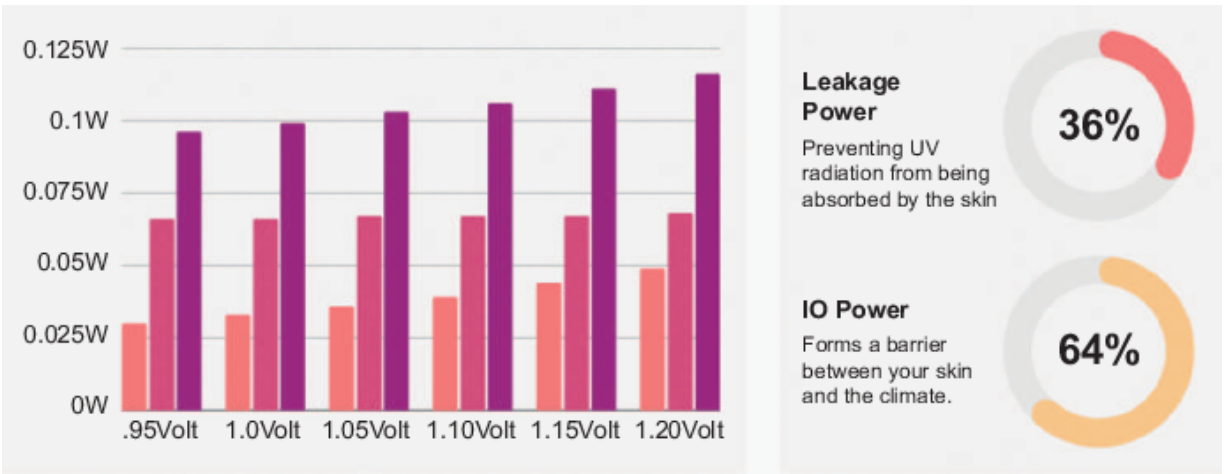
## 4.1 Computing total power consumption of HSTL\_I\_18

**Table 2:** Power consumption using HSTL\_I\_18.

Voltage	Leakage power	IO power	Total power
0.95	0.030 W	0.066 W	0.096 W
1.0	0.033 W	0.066 W	0.099 W
1.05	0.036 W	0.067 W	0.103 W
1.10	0.039 W	0.067 W	0.106 W
1.15	0.044 W	0.067 W	0.111 W
1.20	0.049 W	0.068 W	0.116 W

In [→Table 2](#), the authors have applied HSTL\_I\_18 I/O standard for the experiment. We can observe that in total power consumption 64% is contributed by IO power, while the remaining 36% is contributed by leakage power.

The average value of total power consumption of HSTL\_I\_18 is 0.105 W, which varies the voltage from 0.095 to 1.20 V. Same analysis is shown in the bar graph in [→Figure 10](#).



**Figure 10:** Power consumption using HSTL\_I\_18.

## 4.2 Computing total power consumption of HSTL\_II

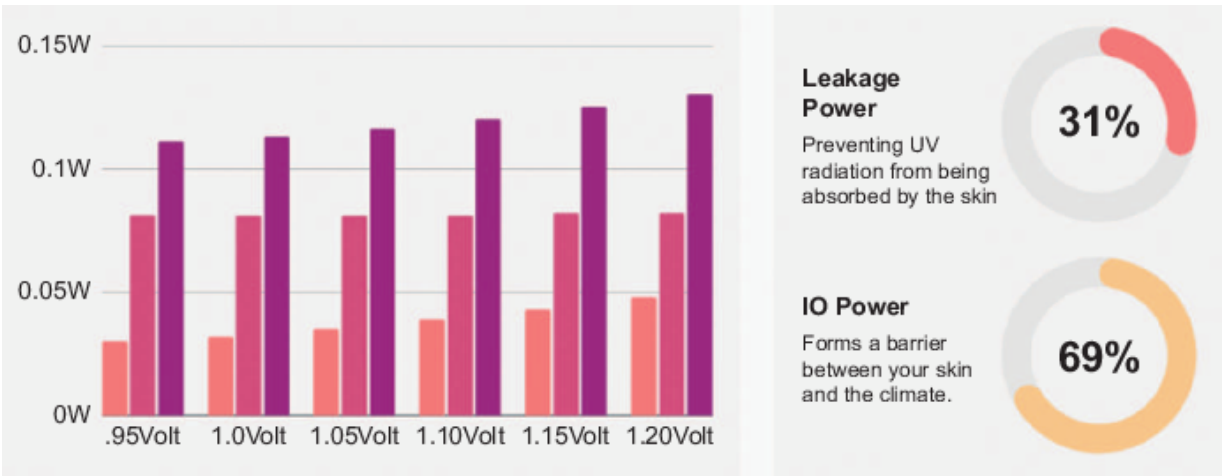
**Table 3:** Power consumption using HSTL\_II.

Voltage	Leakage power	IO power	Total power
.95	0.030 W	0.081 W	0.111 W
1.0	0.032 W	0.081 W	0.113 W
1.05	0.035 W	0.081 W	0.116 W
1.10	0.039 W	0.081 W	0.120 W
1.15	0.043 W	0.082 W	0.125 W
1.20	0.048 W	0.082 W	0.130 W

In [→Table 3](#), we apply IO standard HSTL\_II. During the experiment, when the scholars apply HSTL\_II I/O standard, it is observed that IO power remains almost static with voltage variation and the leakage power changes its value in the range from 0.030 to 0.048.

The average value of total power consumption is 0.119 W, which varies the voltage from 0.095 to 1.20 V. [→Figure 11](#) shows

the same analysis by using the bar graph.



**Figure 11:** Power consumption using HSTL\_II.

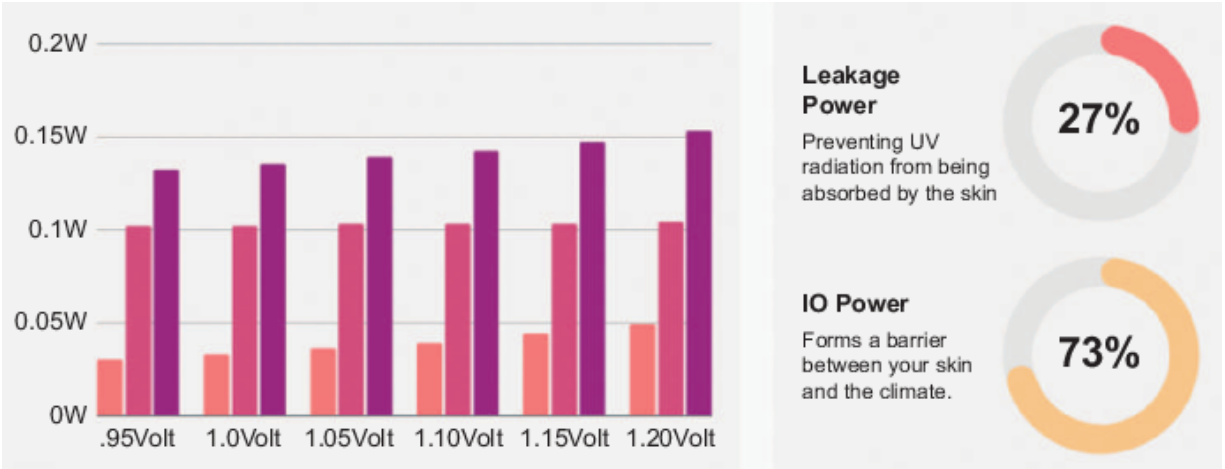
### 4.3 Computing total power consumption of HSTL\_II\_18

**Table 4:** Power consumption using HSTL\_II\_18.

Voltage	Leakage power	IO power	Total power
0.95	0.030 W	0.102 W	0.132 W
1.0	0.033 W	0.102 W	0.135 W
1.05	0.036 W	0.103 W	0.139 W
1.10	0.039 W	0.103 W	0.142 W
1.15	0.044 W	0.103 W	0.147 W
1.20	0.049 W	0.104 W	0.153 W

In [→Table 4](#), the analysis is done with HSTL\_II standards. The total power consumption is the sum of leakage power and IO power of the device, which are 0.231 W and 0.617 W, respectively. The average value of total power consumption is 0.141 W, which varies the voltage from 0.095 to 1.20 V.

The total power analysis for HSTL\_II\_18 I/O standard is shown in [→Figure 12](#) using a bar graph.



**Figure 12:** Power consumption using HSTL\_II\_18.

## 5 Conclusion

During the analysis, the researchers considered two parameters, leakage power and IO power, for computing the total power consumption by implementing voltage scaling technique. During the analysis, it was observed that leakage power is constant among all four families of the HSTL; it varies from 0.030 to 0.049 W. For computing the total power consumption, IO power is the main factor in calculating total power consumption; it determines that 64% of total power in HSTL\_I and 73% of total power in HSTL\_II\_18 are the minimum and maximum values, respectively. In our whole analysis, we found that the HSTL\_I IO standard is most suitable for our AES algorithm because its average value of total power consumption is 0.0938 W, less than all other IO standard families.

## 6 Future scope

Here researchers used Spartan 7 FPGA board for computing the total power consumption of an energy-efficient FPGA-based AES algorithm. However, in the future, research can be done on the latest 28 nm Virtex 7 and Airtex-7 board for computing the total power consumption. In this work, the authors have implemented only voltage scaling technique, but in the upcoming research the frequency scaling can also be taken into consideration.

## References

- [1] S. Pandey, G. Verma, B. Das, T. Kumar, and M. Dhankar, "Energy Efficient Solar Charge Sensor Design Using Spartan-6 FPGA," *Gyancity Journal of Electronics and Computer Science*, vol. 1, no. 1, pp. 18–24, 2016. ISSN: 2446–2918. doi: 10.21058/gjecsc.2016.11004. ➡
- [2] A. Saxena, A. Bhatt, P. Gautam, P. Verma, and C. Patel, "High Performance FIFO Design for Processor through Voltage Scaling Technique," *Indian Journal of Science and Technology*, vol. 9, no. 45, pp. 1–5, 2016. doi: 10.17485/ijst/2016/v9i45/106916. ➡

**[3]** W. Swiegers and J. H. R. Enslin, "An Integrated Maximum Power Point Tracker for Photovoltaic Panels," Available: IEEE Xplore database. Accessed 20 Jul 2006. ➡

**[4]** K. H. Hussein, I. Muta, T. Hoshino, and M. Osakada, "Maximum Photovoltaic Power Tracking: An Algorithm for Rapidly Changing Atmospheric Conditions," in IEEE Proceedings of Generation, Transmission and Distribution, 2006, p. 142. ➡

**[5]** A. Saxena, S. Gaidhani, A. Pant, and C. Patel, "Capacitance Scaling Based Low Power Comparator Design on 28nm FPGA," International Journal of Computer Trends and Technology (IJCTT), 2015. ➡

**[6]** A. Saxena, C. Patel, and M. Khan, "Energy Efficient CRC Design for Processor of Workstation and Server Using LVCMOS," Indian Journal of Science and Technology, vol. 10, no. 4, pp. 1-5, 2017. doi: 10.17485/ijst/2017/v10i4/110890. ➡

**[7]** A. Singla, A. Kaur, and B. Pandey, "LVCMOS Based Energy Efficient Solar Charge Sensor Design on FPGA," in Power Electronics (IICPE), 2014 IEEE 6th India International Conference, 2014, pp. 1-5. doi: 10.1109/IICPE.2014.7115800. ➡

**[8]** M. Renovell, J. Figueras, and Y. Zorian, "Test of RAM-Based FPGA: Methodology and Application to the Interconnect," in 15th

IEEE VLSI Test Symposium, Monterey, CA, 1997, pp. 230–237. ➡

**[9]** R. Roux, G. Schoor, and P. Vuuren, “Block RAM-based Architecture for Real-time Reconfiguration Using Xilinx RFPGAs,” Research Article – SACJ, vol. 56, 2015. ➡

**[10]** C. Patel, P. Verma, P. Agarwal, A. Omer, B. Gururani, and S. Verma, “Designing Green ECG Machine Based on Artix-7 28nm FPGA,” Gyancity Journal of Engineering and Technology, vol. 3, no. 1, pp. 36–41, 2017. doi: 10.21058/gjet.2017.31006. ➡

**[11]** W. K. Huang and F. Lombardi, “An Approach for Testing Programmable/Configurable Field Programmable Gate Arrays,” in 14th IEEE VLSI Test Symposium, Princeton, NJ, USA, 1996, pp. 450–455. ➡

**[12]** A. Saxena, S. Sharma, P. Agarwal, and C. Patel, “SSTL Based Energy Efficient FIFO Design for High Performance Processor of Portable Devices,” International Journal of Engineering and Technology (IJET), vol. 9, no. 2, pp. 113–117, 2017. doi: 10.21817/ijet/2017/v9i2/170902113. [a](#), [b](#)

**[13]** M. Stan, M. Hall, M. Ibrahim, and V. Betz, “HPIPE NX: Boosting CNN Inference Acceleration Performance with AI-Optimized FPGAs,” in 2022 International Conference on Field-Programmable Technology (ICFPT), 2022, pp. 1–9. [a](#), [b](#)

**[14]** A. Boutros, E. Nurvitadhi, and V. Betz, "Architecture and Application Co-Design for Beyond-FPGA Reconfigurable Acceleration Devices," IEEE Access, vol. 10, pp. 95067–95082, 2022. [a](#), [b](#)

**[15]** G. Mao, A. Yakovlev, F. Xia, S. Yu, and R. Shafik, "Automated Mapping of Asynchronous Circuits on FPGA under Timing Constraints," in Proceedings of the 2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2022, pp. 104–109. [a](#), [b](#)

**[16]** R. Ma, E. Georganas, A. Heinecke, S. Gribok, A. Boutros, and E. Nurvitadhi, "FPGA-Based AI Smart NICs for Scalable Distributed AI Training Systems," IEEE Computer Architecture Letters, vol. 21, no. 2, pp. 49–52, 2022. [a](#), [b](#)

**[17]** A. Boutros, E. Nurvitadhi, and V. Betz, "Specializing for Efficiency: Customizing AI Inference Processors on FPGAs," in 2021 International Conference on Microelectronics (ICM), 2021, pp. 62–65. [a](#), [b](#)

**[18]** A. Rafii, W. Sun, and P. Chow, "Pharos: A Multi-FPGA Performance Monitor," 2021 31st International Conference on Field-Programmable Logic and Applications (FPL), 2021, pp. 257–262. [a](#), [b](#)

**[19]** M. Crepaldi, A. Merello, and M. Di Salvo, "A Multi-One Instruction Set Computer for Microcontroller Applications," IEEE Access, vol. 9, pp. 113454–113474, 2021. [a](#), [b](#), [c](#)

**[20]** H. Öztekin, A. Lazzem, and İ. Pehlivan, "Using FPGA-based Content-addressable Memory for Mnemonics Instruction Searching in Assembler Design," The Journal of Supercomputing, 2023. [a](#), [b](#)

**[21]** K. Gavaskar, D. Malathi, G. Ravivarma, P. S. Priyatharshan, S. Rajeshwari, and B. Sanjay, "Design of Low Power Multiplier with Less Area Using Quaternary Carry Increment Adder for New-Fangled Processors," Wireless Personal Communications, 2022. [a](#), [b](#)

**[22]** S. Azimi, C. De Sio, A. Portaluri, D. Rizzieri, and L. Sterpone, "A Comparative Radiation Analysis of Reconfigurable Memory Technologies: FinFET versus Bulk CMOS," Microelectronics Reliability, vol. 138, pp. 114733, 2022. [a](#), [b](#)

**[23]** M. Carminati and G. Scandurra, "Impact and Trends in Embedding Field Programmable Gate Arrays and Microcontrollers in Scientific Instrumentation," Review of Scientific Instruments, vol. 92, no. 9, pp. 091501, 2021. [=>](#)

# A comparative study on security issues and clustering of wireless sensor networks

**Bhawmesh Kumar**

**Ashwani Kumar**

**Harendra Singh Negi**

**Ishwari Singh Rajput**

## **Abstract**

While working with the energy efficiency of the sensor network, clustering focused on dividing the whole area into a small network known as clusters. In each cluster, there is a criterion for the selection of cluster heads to play the role of receiving the sensed data from their cluster members. These cluster heads perform data aggregation to send the complete data to base station of the sensor network. To design the energy-efficient clustering approach, some security challenges, such as internal and external attacks, are faced during transmission between cluster head and cluster member. In this study, the clustering approaches and security issues are addressed. While having cluster-based sensor networks, how the nodes make their clusters and increase the network lifetime these security issues are considered. A critical and related

survey was done in this study and it was found that there are various areas where researchers have scope to work on the security issues faced during clustering of the sensor network. Critical factors, such as cluster count and cluster stability, are also considered for securing clustering wireless sensor networks against attacks.

**Keywords:** Sensor network, clustering, security issues, attacks, energy consumption,

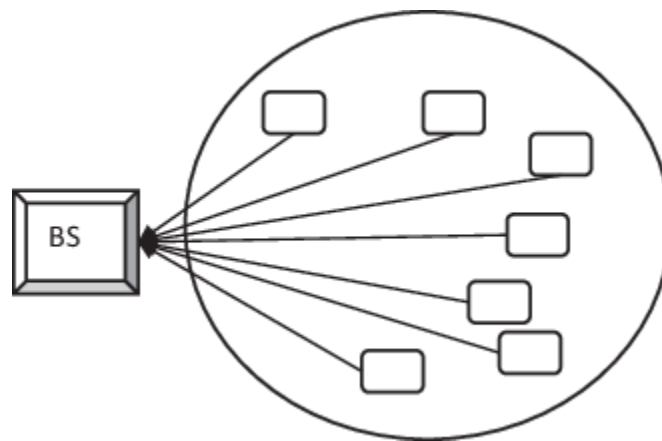
## 1 Introduction

A homogeneous or heterogeneous sensor node with distinguish properties composition of sensor network which is wireless sensor network (WSN). To collect the sensed data from the network node, a center controller node is also available, called a “base station” (BS) and sink node, which is connected through network communication. Security, protection, calculation and energy imperatives, and unwavering quality issues are a few significant difficulties confronting WSNs, particularly during routing [[→1](#)]. To tackle these difficulties, WSN routing should guarantee secrecy, uprightness, protection safeguarding, and dependability in the network area [[→2](#)]. To have an energy-efficient approach for sensor nodes, clustering would help reduce the usage of energy of each node [[→3](#)]. In addition, the security issues during the clustering of sensor networks need to

be considered. Some physical or environmental conditions are connected to a sensor node while sensing the data [[→4](#)], because sensed data should be more secure to reach at the BS in multihop communication. In between the data traveled through various intermediate nodes. So, there is a possibility of attacks on sensed data. Some modifications can also be done by the intruder nodes.

Nodes are connected with the help of distinguishing network topologies [[→5](#)]. The enhancement of the network lifetime needs some efficient approach to form the nodes in a way that remains for the lifetime. The deployment of nodes can be in a hostile or hilly area where the battery backup is not easily available [[→6](#)]. The clusters form with the aid of some clustering approaches and later some selection mechanism is applied to elect the cluster head (CH). When the CH is selected, then all the cluster members send their sensed data to their respective CH. Later, the CH performs the data aggregation to send complete data to the BS. So, in this case, the whole network is divided into clusters; management of all nodes becomes easy and energy of the network also increases. But in some situations, there may also be some security threats and attacks to clustering. In clustering, there can be some challenges at the time of clustering.

→[Figure 1](#) shows how nodes are connected to the BS and also represents all nodes that come under a single cluster. The properties, applications, and routing protocols are explained in WSNs [[→7](#), [→8](#), [→9](#), [→10](#)]. This study also focused on the security challenges while designing the cluster-based sensor network [[→11](#), [→12](#), [→13](#)].



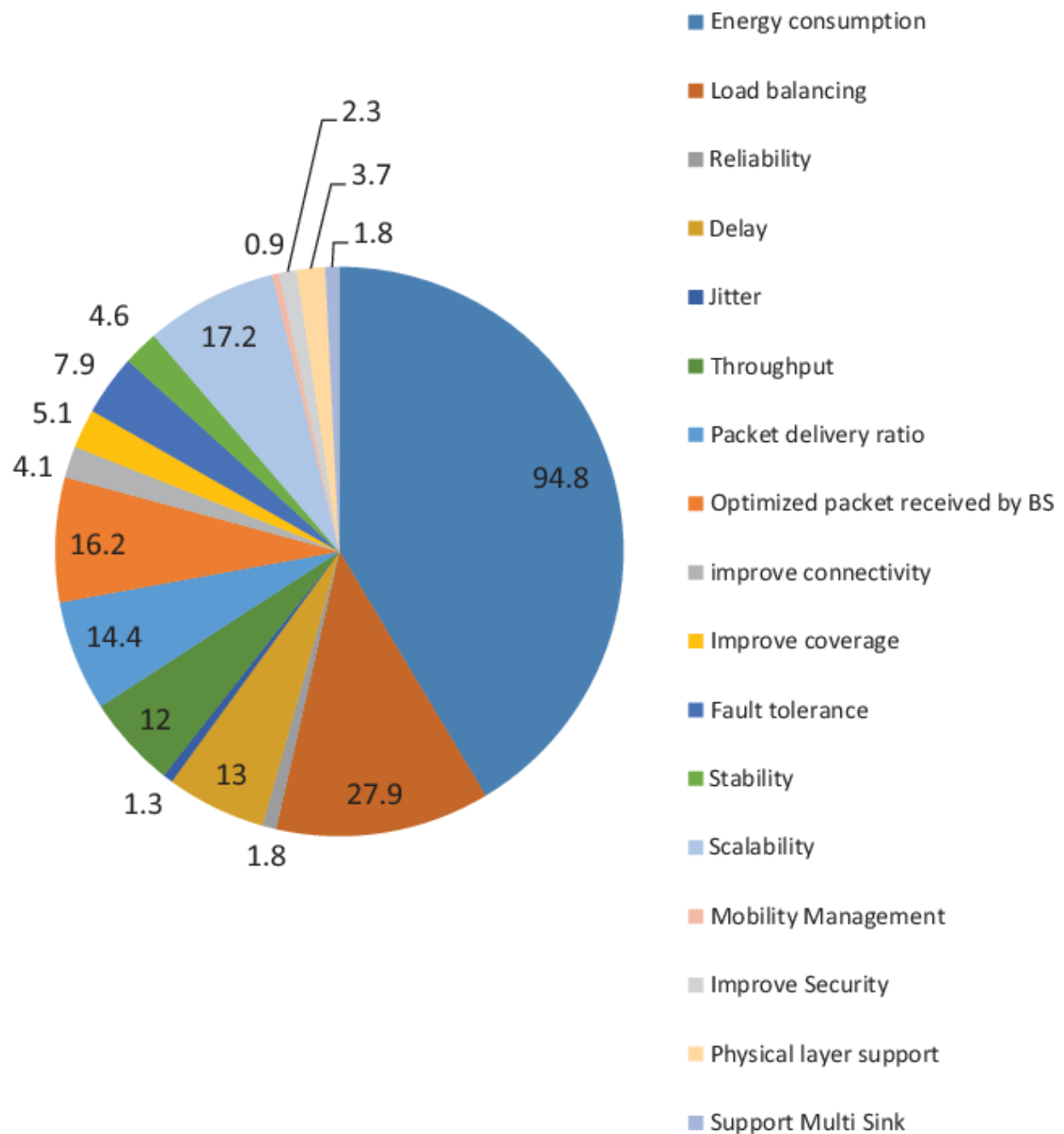
**Figure 1:** Wireless sensor network.

## 1.1 Popular research areas of WSNs

- Data transmission scheduling techniques
- Data aggregation
- Routing protocols
- Energy-efficient clustering
- Deployment strategies
- Security challenges during clustering

## **1.2 Comparison of clustering objectives and energy consumption of node**

For comparison, various clustering objectives are considered such as energy consumption, load balancing, improved security, fault tolerance, and others. The work is approximately done 95% on energy consumption whereas to improve the security 2.3% only as per discussed [[→14](#)]. The objectives research is shown in [→Figure 2](#).



**Figure 2:** Analysis percentage of clustering objectives.

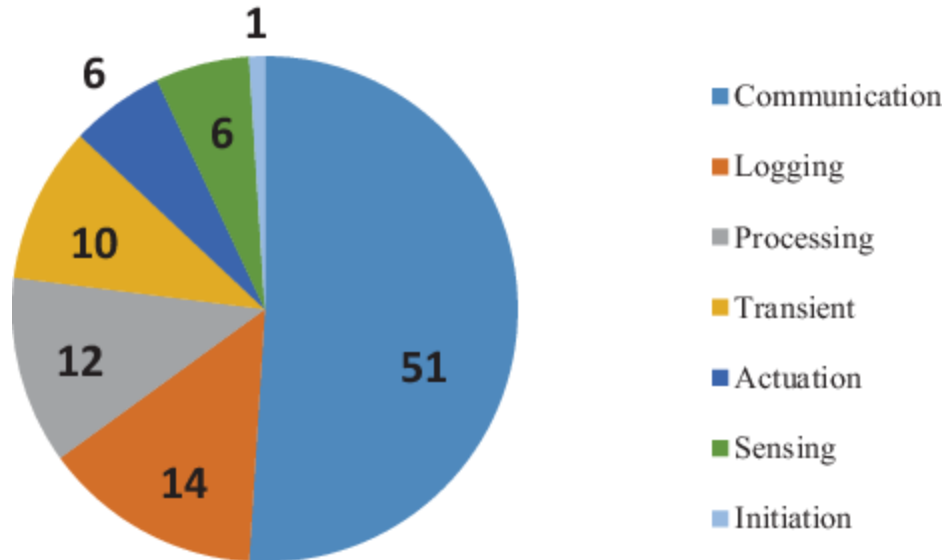
Energy consumption [[→15](#)] by a sensor node is shown in [→Figure 3](#). For communication parameter, the usage of energy

consumption of sensor node is 51%, 14% for logging and 12% for processing and remaining parameters.

### **1.3 Security and privacy issues in WSN**

Following security and privacy issues are considered in the sensor networks [[→16](#)].

1. *Security and privacy issues:* For a resource requirement, sensor network with source node sending information to the objective through a few go-between nodes, there is plausible of interruption, character followed by a foe, gathering, and change of source information by the middle nodes.
2. *Manipulating routing information:* This focuses on the routing data between the two sensors. It may be sent off by mocking or replaying the directing data.



**Figure 3:** Energy consumption by the node (in percentage).

1. *Sybil attack*: In this attack, the foe compromises of the WSN by making counterfeit personalities to disturb the network conventions.
2. *Sinkhole attack*: In this attack, there are chances of the sink node getting the total and right information from the sensors, hence causing a danger to higher-layer applications.
3. *Clone attack*: In this attack, catches the real nodes of the sensor network, collects the data from location of nodes, makes replicas as clone nodes, and lastly conveys them to the network.
4. *Selective forwarding attack*: A malicious node in the sensor network specifically advances a few information parcels to the BS while dropping others, with intent on compromising the uprightness and accessibility of the sensor network.

5. *HELLO flood attack*: An unlawful node in the sensor network floods hello requests to the real node and breaks the security.
6. *Denial of service attack*: It disables the main functionality of the sensor network.

## 2 Critical literature review

To enhance the energy level of the sensor network, there is a need for an efficient approach for the betterment of the network. Through clustering, the whole network is split into clusters, where CHs and CMs perform their tasks in the sensor network [[→17](#)]. Some routing protocols are also available to perform the transformations. The homogeneous clustering protocols are LEACH [[→18](#)], HEED [[→19](#)], PEGASIS [[→20](#)], and so on, whereas heterogeneous clustering protocols are SEP [[→21](#)], LEACH-E [[→22](#)], DEEC [[→23](#)], and so on. There are a few security and protection issues related to multihop directing. Snooping, spoofing, sinkholes, tampering with Sybil, cloning, and so on are just a few examples of these problems that influence the WSNs' data confidentiality, availability, and integrity. We have performed a critical research survey on the existing state-of-the-art clustering algorithms along with security and protection issues, which are discussed below.

Chelli [[→24](#)] explained issues and attacks on the security of sensor networks and also gave the countermeasures. Mahboub et al. [[→25](#)] proposed a hybrid clustering focused on energy efficiency with the help of K-means algorithm. Ouafaa et al. [[→26](#)] proposed a LEACH protocol to enhance the clustering approach in the sensor network. In WSN, internal and external attacks are vulnerable. In the future, this approach can also be implemented using MICA 2 mote on TinyOS.

Zhao et al. [[→27](#)] introduced the optimized agglomerative nesting algorithm that focuses on the distance, dual CHs, and node dormancy mechanism. In this approach, CH depends on the residual energy and node location. This proposed method reduces the energy consumption decay rate, increases the network lifetime, and improves the throughput as well. Jain et al. [[→28](#)] examined the energy-based K-means clustering approach that considered the distance between CHs to be minimal. Kanchan et al. [[→29](#)] presented a quantum-based PSO (particle swarm optimization) approach for energy-efficient clustering (QPSOEEC). This proposed algorithm works when the position vector parameter and BS position vary.

Liang et al. [[→30](#)] suggested the routing protocol, it is calculated the optimal cluster count which based on complete energy consumption. In this proposed work, CH is used to depict the

Voronoi diagram and ant colony to optimize the multihop routing protocol.

Azizi et al. [[→31](#)] proposed a new efficient clustering protocol for energy consumption for both types of networks, homogenous and heterogeneous. This work is based on two parameters: residual energy and distance. On the basis of these two parameters, the selection of CH process is implemented.

Minani [[→32](#)] presented an energy-efficient clustering algorithm to prolong the lifetime of the sensor network. The proposed system chooses the farthest node as the CH based on the following parameters: density of node, residual energy, and intercluster distance. This method improves the lifetime of sensor network and throughput and degrades the packet delay and sensor power consumption.

Zhu and Wei [[→33](#)] invented an unequal clustering protocol, which is energy efficient for WSN, to form a double CH technique for improving the energy efficiency of the whole network. To improve the overhead of CH, this protocol allows to have two CHs: main head and vice head. To balance the energy consumption, a rotational-based CH is considered that takes into account energy and time.

Mood et al. [[→34](#)] introduced a power distance sums scaling gravitational search algorithm (PDSS-GSA) to find the optimal count, manage the clusters, and select the best CH for each round. This approach is based on high-energy level CH and is an extension of the gravitational search algorithm.

Bongale et al. [[→35](#)] support data aggregation based on an intracluster strategy for the sensor network. The data coming from different nodes aggregated at the CH node is further forwarded to the BS. The proposed technique improved the network lifetime after implementing the data aggregation.

Yao et al. [[→36](#)] suggested a novel energy-balanced clustering routing (EBCR), maximizing the lifespan of network. This work gives a complete solution to the clustering process such as CH selection and intercluster routing. It supports distributed clustering that introduces dynamic cluster radius and division schemes at the intersection region node with the support of new principles. This method balances the competition between CHs.

Khediri et al. [[→37](#)] invented a distance energy-evaluated (DEE) approach, which follows energy-based CH selection along with the consideration of degree and distance. This approach improves the reliability, lifetime, and energy consumption.

Sinde et al. [[→38](#)] focus on reducing the network delay and enhancing the lifetime of network. So, they proposed a deep reinforcement learning (DRL) based energy-efficient scheduling concept. It is centered on a zonal-based clustering scheme, which works along with particle swarm optimization and affinity propagation. This proposed algorithm is implemented on the NS3 simulator.

Wang et al. [[→39](#)] proposed a location and residual energy factors-based rotational CH selection approach for improving the energy consumption. This protocol gives a remarkable output for network energy saving and is capable of working against energy constraints. Ren et al. [[→40](#)] proposed EECHS, which is an energy-efficient CH selection scheme. The whole sensor network divides the nodes as CH, cluster member, and scheduling node. This approach follows an efficient CH selection criterion and results in better performance compared to other protocols.

Gheisari et al. [[→41](#)] suggested a literature review on clustering algorithms in WSNs with challenges, research, and trends. The comparison of different clustering protocols on the basis of complexity, described in this review. In summary, the described techniques, related methodologies, outcomes, limitations, and further work are discussed.

Periyasamy et al. [[→42](#)] presented a K-means approach with modification that improves the sensor network's time and also minimizes the clustering process than K-means clustering for sensor network. It does not calculate the optimum number of CHs and does not support multihop communication. Future work can focus on improving multihop communication and also on finding the optimal number of CHs.

Ray et al. [[→43](#)] produced balanced clusters, which help to balance the load of CHs, find the optimal number of CH, and also prolong the network lifetime. This approach ignores data loss, and energy consumption is balanced for all sensor nodes. In the future, this approach can be used to minimize the data loss and also balance the energy usage of sensor nodes.

Zhao et al. [[→44](#)] modified the CH selection algorithm-based LEACH protocol, which focused on energy burden and on increasing the size of the data received at the BS. This algorithm improves the network lifetime, energy usage, and amount of data. Future work focuses on improving energy saving, taking into consideration the distance between CHs. This approach can be explored by considering the distance of CH from BS and the distance of cluster members from CH till the last round of energy is used.

Kumar et al. [[→45](#)] proposed a statistical analysis of the survey of distinguishing machine learning algorithms. Multiple challenges in WSN addressed by machine learning techniques, such as data aggregation, energy consumption, mobile sink path, and synchronization, are highlighted. The implemented deployment of nodes is mostly two-dimensional. It observed that further studies are required to deploy the sensor nodes in three-dimensional spaces. In the future, there is scope to work on deploying the clustering algorithm for static and mobile sensor network three-dimensional space. In addition, the energy efficiency of the mobile sink can also be improved.

Ishaq et al. [[→46](#)] proposed a new clustering approach along with security involvement. It improves the cluster security by using an inspector node to monitor the CH. In future, involvement of IoT with some data mining algorithms to have other nodes behavior and attacks analysis also need attention.

Behera et al. [[→47](#)] assumed that three parameters are sufficient to elect the CH; these parameters are initial energy level, residual energy, and optimal value of clusters. The limitation of this proposed algorithm is that more parameters should also be considered to select the CH. In the future, more parameters must be included for the CH selection. This algorithm can be explored as an alternative for selecting the CH by considering some more parameters.

In Elsadig et al.'s [[→48](#)] work, WSN constraints, vulnerabilities, and security attacks are considered. Investigated the countermeasures, used to counter WSN attacks and their limitations. A balanced outcome of WSN security should be the future goal.

A qualitative literature survey of clustering objectives with homogeneous and heterogeneous networks was done by Shahraki et al. [[→14](#)]. Here, the authors used the associated methods to achieve each clustering objective. They showed that the clustering process can solve multiple network challenges. In the future, we can consider QoS, load balancing, and mobility management. In a heterogeneous network, selected CH for long time will drain the sensor node energy rapidly. This also showed that CH should be rotated so the CH responsibility is balanced. In future work, CH duty rotation should be incorporated. To overcome huge energy consumption, there are some methods: CH duty rotation, hierarchical clustering, unbalanced hierarchical clusters, and balancing between clusters. In the future, we need to find the solution to these challenges.

Khan et al. [[→49](#)] focused on WSN security attacks and their consequences. They evaluated the solution of these attacks and

derived secured cluster-based solutions. Critical factors should also be considered in future to secure WSN clustering.

Xia et al. [[→50](#)] studied the internal and external attacks of WSN. The considered parameters are routing path, identification of malicious nodes, and so on. Some other security concerns can also be taken into account to improve the clustering performance of WSNs.

As per the above discussion, it is found that there are many areas of secured WSNs clustering; some of these points are listed in [→Table 1](#).

**Table 1:** Future scope area.

### **Future scope areas**

To emphasize multihop communication and also to find the optimal number of CHs [[→51](#)].

To minimize the data loss and also balance the energy usage of sensor nodes [[→43](#)].

More work is required to support of multipath and heterogeneous nodes in the future of this approach [[→31](#)].

Three-dimension-based routing protocol for clustering.

CH formation to reduce the energy consumption for both homogeneous and heterogeneous networks [[→28](#)].

There is a huge difference between two- and three-dimension-based routing protocols. The future work suggests a three-dimensional space routing protocol. In the future, there is a need to improve parameters like delay and first node death [[→45](#)].

## **Future scope areas**

Datamining algorithms to have other nodes' behavior and attacks as well [[→46](#)].

To update the proposed methods to exceed the number of clusters and optimize multiple paths and mobile BS [[→30](#)].

More work is needed further to extend subclustering with sub-CH and also a space for homogeneous and heterogeneous sensor network types [[→52](#)].

Balanced outcome of WSN security [[→48](#)].

Secure WSN clustering by attacks and vulnerability [[→49](#)].

To secure and improve the performance of WSN clustering [[→50](#)].

## Future scope areas

To incorporate the CH duty rotation. To overcome the huge energy consumption, some methods are CH duty rotation, hierarchical clustering, unbalanced hierarchical clusters, and balancing between clusters. In the future, to find the solution to these challenges [[→14](#)].

## 3 Research gaps identified

On the basis of the exhaustive and critical literature survey, we found active research gaps in the field of security issues and clustering in WSN, which are listed as follows:

1. Initial energy level, residual energy, and the number of clusters are the parameters used to select the CH. Some more parameters, such as distance between nodes and distance from BS, must also be taken into consideration for selecting the CH to maintain the energy efficiency of the sensor network. This is a research gap between different proposed protocols.
2. Not every node can be used as a CH for a long time; the CH duty should be rotated to balance the responsibility of being the CH, which is identified as an open research issue.

3. Development and assessment of advanced clustering approaches to improve the performance using mobility of nodes and heterogeneity of sensor network is also a research gap.
4. Minimizing the energy consumption mobility of the BS is identified as a research gap, which, in turn, can increase the network's lifetime.
5. Clustering can also be extended as subclustering; clusters can have their subclusters to improve the energy of the sensor network. This indicates that there is still scope for research in the direction of improving energy usage.
6. There are various frameworks for enhancing energy efficiency clustering using unsupervised algorithms such as k-means. However, most models do not minimize the data loss and also do not balance the energy usage of sensor nodes. These gaps are identified as open research issues to enhance energy usage.
7. In future, involvement of IoT with some data mining algorithms to have other nodes behavior and attacks analysis also need attention.
8. Critical factors, such as cluster count, cluster stability, and topology of intraccluster, should also be considered against the attacks and issues in WSNs.
9. A balanced outcome of WSN security should be a future goal.

10. Secured LEACH approach can also be implemented on other types of motes and operating systems to test the performance of real hardware.

## **4 Conclusion**

While working with the energy effectiveness of sensor organization, clustering is used to separate the entire region into little organizations known as groups. In each cluster, there are models of CHs to assume the part of getting the detected information from their cluster nodes individually. To send all the data to the BS of the sensor network, these CHs perform data aggregation. During transmission between the CH and member, the energy-efficient clustering approach faces security issues, such as attacks (within and outside). In this chapter, the clustering approaches and security concerns are tended to. This research takes into account several issues related to cluster-based sensor networks, including how nodes construct clusters and how to extend the network's lifespan. The critical and related survey conducted for this study revealed several opportunities for researchers to investigate the challenges associated with sensor network clustering. Critical aspects are also taken into account when protecting clustering in a WSN.

Research gaps are identified and explained during this study of security issues and clustering-based WSNs.

In the future, applications and hardware-oriented systematic literature surveys can also be done. More research points are also discussed in terms of routing protocols of sensor networks on the basis of security attacks.

## References

[1] L. Han, M. Zhou, W. Jia, Z. Dalil, and X. Xu, "Intrusion Detection Model of Wireless Sensor Networks based on Game Theory and an Autoregressive Model," *Information Sciences (N Y)*, vol. 476, pp. 491–504, 2019. ➡

[2] N. Shabbir and S. R. Hassan, "Routing Protocols for Wireless Sensor Networks (WSNs)," in *Wireless Sensor Networks – Insights and Innovations*, InTech, 2017. doi: 10.5772/intechopen.70208. ➡

[3] A. More and V. Raisinghani, "A Survey on Energy Efficient Coverage Protocols in Wireless Sensor Networks," *Journal of King Saud University – Computer and Information Sciences*, vol. 29, no. 4, pp. 428–448, 2017. doi: 10.1016/j.jksuci.2016.08.001. ➡

**[4]** A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler, "Wireless Sensor Network for Habitat Monitoring.pdf," IEEE Communications Magazine, pp. 102–114, 2002. ➡

**[5]** S. Santha Meena and J. Manikandan, "Study and Evaluation of Different Topologies in Wireless Sensor Network," in Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017, vol. 2018-Jan, pp. 107–111, 2018. doi: 10.1109/WiSPNET.2017.8299729. ➡

**[6]** V. Sharma, R. B. Patel, H. S. Bhadauria, and D. Prasad, "Deployment Schemes in Wireless Sensor Network to Achieve Blanket Coverage in Large-scale Open Area: A Review," Egyptian Informatics Journal, vol. 17, no. 1, pp. 45–56, 2016. doi: 10.1016/j.eij.2015.08.003. ➡

**[7]** S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A Taxonomy of Wireless Micro-sensor Network Models," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 6, no. 2, pp. 28–36, 2002. doi: 10.1145/565702.565708. ➡

**[8]** J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," IEEE Wireless

Communications, vol. 11, no. 6, pp. 6–27, 2004. doi: 10.1109/MWC.2004.1368893. ➡

**[9]** L. K. Ketshabetswe, A. M. Zungeru, M. Mangwala, J. M. Chuma, and B. Sigweni, “Communication Protocols for Wireless Sensor Networks: A Survey and Comparison,” *Heliyon*, vol. 5, no. 5, p. e01591, 2019. doi: 10.1016/j.heliyon.2019.e01591. ➡

**[10]** I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: A Survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002. doi: 10.1016/S1389-1286(01)00302-4. ➡

**[11]** Y.-J. Han, M.-W. Park, and T.-M. Chung, “SecDEACH: Secure and Resilient Dynamic Clustering Protocol Preserving Data Privacy in WSNs,” in *ICCSA 2010, Part III, LNCS 601*, Springer, pp. 142–157, 2010. ➡

**[12]** B. R. Swathi and G. A. Kumar, “Secure and Efficient Method of Overcoming Various Attacks in CWSN’s,” *International Journal of Engineering Research & Technology*, vol. 3, no. 4, pp. 1188–1190, 2014. [Online]. Available: ➡[www.ijert.org](http://www.ijert.org) ➡

**[13]** A. Choudhary, S. Kumar, and K. P. Sharma, “RFDCS: A Reactive Fault Detection and Classification Scheme for Clustered

WSNs,” Peer-to-Peer Networking and Applications, vol. 15, no. 3, pp. 1705–1732, May 2022. doi: 10.1007/s12083-022-01308-5. ➡

**[14]** A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, “Clustering Objectives in Wireless Sensor Networks: A Survey and Research Direction Analysis,” Computer Networks, vol. 180, no. June, 2020. doi: 10.1016/j.comnet.2020.107376. [a](#), [b](#), [c](#)

**[15]** S. Yadav and R. S. Yadav, “A Review on Energy Efficient Protocols in Wireless Sensor Networks,” The Journal of Mobile Communication, Computation and Information, vol. 22, no. 1, pp. 335–350, 2016. doi: 10.1007/s11276-015-1025-x. ➡

**[16]** O. Olufemi Olakanmi and A. Dada, “Wireless Sensor Networks (WSNs): Security and Privacy Issues and Solutions,” in Wireless Mesh Networks – Security, Architectures and Protocols, IntechOpen, 2020. doi: 10.5772/intechopen.84989. ➡

**[17]** A. A. Abbasi and M. Younis, “A Survey on Clustering Algorithms for Wireless Sensor Networks,” Computer Communications, vol. 30, no. 14–15, pp. 2826–2841, 2007. doi: 10.1016/j.comcom.2007.05.024. ➡

**[18]** W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” Proceedings of the 33rd Annual Hawaii

International Conference on System Sciences, Maui, HI, USA, vol. 2-, pp. 10 pp, 2000. doi: 10.1109/HICSS.2000.926982. ➡

**[19]** O. Younis and S. Fahmy, "HEED : A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks," in IEEE Transactions on Mobile Computing, vol. 3, no. 4, pp. 366–379, Oct.–Dec. 2004, doi: 10.1109/TMC.2004.41. ➡

**[20]** S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," Proceedings, IEEE Aerospace Conference, Big Sky, MT, USA, 2002, pp. 3–3, doi: 10.1109/AERO.2002.1035242. ➡

**[21]** G. Smaragdakis, I. Matta, and A. Bestavros. SEP: A stable election protocol for clustered heterogeneous wireless sensor networks. Boston University Computer Science Department, 2004. ➡

**[22]** W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-specific Protocol Architecture for Wireless Microsensor Networks," IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660–670, 2002. doi: 10.1109/TWC.2002.804190. ➡

**[23]** L. Qing, Q. Zhu, and M. Wang, "Design of a Distributed Energy-efficient Clustering Algorithm for Heterogeneous

Wireless Sensor Networks," Computer Communications, vol. 29, no. 12, pp. 2230–2237, 2006. doi: 10.1016/j.comcom.2006.02.017.

⇒

**[24]** K. Chelli, "Security Issues in Wireless Sensor Networks: Attacks and Countermeasures," in Proceedings of the World Congress on Engineering 2015, 2015. ⇒

**[25]** A. Mahboub, M. Arioua, and E. M. En-Naimi, "Energy-efficient Hybrid K-means Algorithm for Clustered Wireless Sensor Networks," International Journal of Electrical and Computer Engineering, vol. 7, no. 4, pp. 2054–2060, 2017. doi: 10.11591/ijece.v7i4.pp2054-2060. ⇒

**[26]** O. Bayat, S. Aljawarneh, and H. F. Carlak, International Association of Researchers, Institute of Electrical and Electronics Engineers, and Akdeniz niversitesi, "A New Secure Solution for Clustering in Wireless Sensors Networks Based on LEACH," in Proceedings of 2017 International Conference on Engineering & Technology (ICET'2017), 2017. ⇒

**[27]** Z. Zhao, K. Xu, G. Hui, and L. Hu, "An Energy-efficient Clustering Routing Protocol for Wireless Sensor Networks based on AGNES with Balanced Energy Consumption Optimization,"

Sensors (Switzerland), vol. 18, no. 11, 2018. doi: 10.3390/s18113938. [→](#)

**[28]** B. Jain, G. Brar, and J. Malhotra, "EKMT-k-Means Clustering Algorithmic Solution for Low Energy Consumption for Wireless Sensor Networks Based on Minimum Mean Distance from Base Station," pp. 113–123, 2018. doi: 10.1007/978-981-10-4585-1\_10. [a](#), [b](#)

**[29]** P. Kanchan and S. D. Pushparaj, "A Quantum Inspired PSO Algorithm for Energy Efficient Clustering in Wireless Sensor Networks," Cogent Engineering, vol. 5, no. 1, pp. 1–16, 2018. doi: 10.1080/23311916.2018.1522086. [→](#)

**[30]** H. Liang, S. Yang, L. Li, and J. Gao, "Research on Routing Optimization of WSNs Based on Improved LEACH Protocol," EURASIP Journal on Wireless Communications and Networking, vol. 2019, no. 1, 2019. doi: 10.1186/s13638-019-1509-y. [a](#), [b](#)

**[31]** M. S. Azizi and M. L. Hasnaoui, "An Energy Efficient Clustering Protocol for Homogeneous and Heterogeneous Wireless Sensor Network," ACM International Conference Proceeding Series, vol. Part F1481, 2019. doi: 10.1145/3320326.3320396. [a](#), [b](#)

**[32]** F. Minani, "Maximization of Lifetime for Wireless Sensor Networks based on Energy Efficient Clustering Algorithm," International Journal of Electronics and Communication Engineering, vol. 13, no. 6, pp. 389–395, 2019. ➡

**[33]** F. Zhu and J. Wei, "An Energy-efficient Unequal Clustering Routing Protocol for Wireless Sensor Networks," vol. 15, no. 9, 2019. doi: 10.1177/1550147719879384. ➡

**[34]** S. Ebrahimi Mood and M. M. Javidi, "Energy-efficient Clustering Method for Wireless Sensor Networks Using Modified Gravitational Search Algorithm," Evolving Systems, vol. 11, no. 4, pp. 575–587, 2020. doi: 10.1007/s12530-019-09264-x. ➡

**[35]** A. M. Bongale, C. R. Nirmala, and A. M. Bongale, "Energy Efficient Intra-cluster Data Aggregation Technique for Wireless Sensor Network," International Journal of Information Technology (Singapore), 2020. doi: 10.1007/s41870-020-00419-7. ➡

**[36]** Y. Yao, W. Chen, J. Guo, X. He, and R. Li, "Simplified Clustering and Improved Intercluster Cooperation Approach for Wireless Sensor Network Energy Balanced Routing," EURASIP Journal on Wireless Communications and Networking, vol. 2020, no. 1, 2020. doi: 10.1186/s13638-020-01748-8. ➡

**[37]** S. El Khediri, N. Nasri, R. Ullah, and K. Abdennaceur, "An Improved Energy Efficient Clustering Protocol for Increasing the Life Time of Wireless Sensor Networks," *Wireless Personal Communications*, no. 0123456789, 2020. doi: 10.1007/s11277-020-07727-y. ➡

**[38]** R. Sinde, F. Begum, K. Njau, and S. Kaijage, "Refining Network Lifetime of Wireless Sensor Network Using Energy-efficient Clustering and DRL-based Sleep Scheduling," *Sensors (Switzerland)*, vol. 20, no. 5, 2020. doi: 10.3390/s20051540. ➡

**[39]** J. Wang, Z. Du, Z. He, and X. Wang, "A Cluster-Head Rotating Election Routing Protocol for Energy Consumption Optimization in Wireless Sensor Networks," *Complexity*, vol. 2020, Article ID 6660117, 13 pages, 2020.

➡<https://doi.org/10.1155/2020/6660117>. ➡

**[40]** Q. Ren and G. Yao, "An Energy-efficient Cluster Head Selection Scheme for Energy-harvesting Wireless Sensor Networks," *Sensors (Switzerland)*, vol. 20, no. 1, pp. 1–17, 2020. doi: 10.3390/s20010187. ➡

**[41]** M. Gheisari, et al., "A Survey on Clustering Algorithms in Wireless Sensor Networks: Challenges, Research, and Trends," in *Proceedings – 2020 International Computer Symposium, ICS*

2020, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 294–299. doi: 10.1109/ICS51289.2020.00065. ➡

**[42]** S. Periyasamy, S. Khara, and S. Thangavelu, “Balanced Cluster Head Selection Based on Modified K-means in a Distributed Wireless Sensor Network,” vol. 2016, 2016. doi: 10.1155/2016/5040475. ➡

**[43]** A. Ray and D. De, “Energy Efficient Clustering Protocol Based on for Enhanced Network Lifetime in Wireless Sensor Network,” pp. 1–11, 2016. doi: 10.1049/iet-wss.2015.0087. [a](#), [b](#)

**[44]** L. Zhao, S. Qu, and Y. Yi, “A Modified Cluster-head Selection Algorithm in Wireless Sensor Networks Based on LEACH,” EURASIP Journal on Wireless Communications and Networking, vol. 2018, no. 1, 2018. doi: 10.1186/s13638-018-1299-7. ➡

**[45]** D. P. Kumar, T. Amgoth, C. Sekhara, and R. Annavarapu, “Machine Learning Algorithms for Wireless Sensor Networks : A Survey,” Information Fusion, vol. 49, no. April 2018, pp. 1–25, 2019. doi: 10.1016/j.inffus.2018.09.013. [a](#), [b](#)

**[46]** Z. Ishaq, S. Park, and Y. Yoo, “A Security Framework for Cluster-Based Wireless Sensor Networks against the Selfishness Problem,” Wireless Communications and Mobile Computing, vol. 2018, 2018. doi: 10.1155/2018/8961239. [a](#), [b](#)

**[47]** T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, "Residual Energy Based Cluster-head Selection in WSNs for IoT Application," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5132-5139, June 2019, doi: 10.1109/JIOT.2019.2897119. ➔

**[48]** M. A. Elsadig, A. Altigani, and M. A. A. Baraka, "Security Issues and Challenges on Wireless Sensor Networks," International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 4, pp. 1551–1559, Jul. 2019. doi: 10.30534/ijatcse/2019/78842019. [a](#), [b](#)

**[49]** J. Khan, A. Munir Shah, B. Nawaz, P. Penh Cambodia Khalid Mahmood, M. Kashif Saeed, and M. Ul Hassan, "A Cluster-Based Mitigation Strategy against Security Attacks in Wireless Sensor Networks," 2020. [Online]. Available: ➔[www.ijacsa.thesai.org](http://www.ijacsa.thesai.org) [a](#), [b](#)

**[50]** Z. Xia, Z. Wei, and H. Zhang, "Review on Security Issues and Applications of Trust Mechanism in Wireless Sensor Networks," Computational Intelligence and Neuroscience, vol. 2022, Hindawi Limited, 2022. doi: 10.1155/2022/3449428. [a](#), [b](#)

**[51]** P. Sasikumar and S. Khara, "K-Means Clustering in Wireless Sensor Networks," Proceedings – 4th International Conference

on Computational Intelligence and Communication Networks, CICN 2012, pp. 140–144, 2012. doi: 10.1109/CICN.2012.136. ➡

**[52]** F.Zhu and J.Wei, “An Energy-efficient Unequal Clustering Routing Protocol for Wireless Sensor Networks,” International Journal of Distributed Sensor Networks, vol. 15, no. 9, 2019. doi:10.1177/1550147719879384. ➡

# Heuristic approach and its application to solve NP-complete traveling salesman problem

**Deonarain Brijlall**

**Tauqeer Ahmed Usmani**

**Richa Indu**

## **Abstract**

Since long ago, a suitable solution to the traveling salesman problem in different scenarios has always been a popular problem for research. Various heuristic and evolutionary approaches have been designed for it. We developed a simple heuristic approach to identify  $n$  optimal routes from  ${}^nC_2$  routes abiding a degree constraint, where only those routes are selected in the set of feasible routes ( $Hx$ ), which have a degree less than or equal to 2. We implemented the present tactic on the milk delivery problem, that is, to determine the best route for a milk van supplying milk to (i) 10 houses and (ii) 20 houses, in an area including dairy. For (i), out of 45 possible routes, 10 optimal routes of Hamiltonian cycle length 188 have been selected in 0.1344 s. Similarly, for (ii), 20 optimal routes of Hamiltonian cycle length 328 have been selected in 0.2488 s from 190 total routes. In this way, the time complexity of the

proposed heuristic approach is  $O(n^2 \log_2 n)$ , where its solution always lies in the right neighborhood of 0.

**Keywords:** Traveling salesman problem, heuristic solution, nearest neighbor, optimal, sorting,

## 1 Introduction

If one is on a trip to a city and wants to visit each of its popular places at most once, then using the traveling salesman problem (TSP) is worth the effort. The main objective of a TSP is the minimization of the time or distance consumed, whatever the constraint in visiting each location. However, this is not the lone area where TSP is used. In 1932, Karl Menger proposed a problem, namely, the messenger problem based on the postal messengers [[→1](#), [→2](#)], which begins from the starting location to the next closest location. However, it does not result in the shortest path. Then in the 1940s Mahalanobis studied this problem in connection with agricultural fields of Bengal for the transportation of men and equipment [[→3](#)]. Later on, Flood in 1956 [[→4](#)] addressed TSP as an NP-hard problem, and many other authors too analyzed TSP with their solutions [[→1](#)].

From long ago, TSP was used in overhauling gas turbine engines [[→5](#)], drilling the PCBs to minimize the travel time of machine heads [[→6](#)], in X-ray crystallography [[→7](#)], routing postal

vehicles, school buses [[→8](#), [→9](#)], scheduling printing press, cranes, and crew [[→10](#), [→11](#), [→12](#)], picking orders from warehouses [[→13](#), [→14](#), [→15](#)], in global navigation satellite systems [[→16](#)], network optimization [[→1](#), [→17](#)], and in steel and iron industries [[→18](#)]. Therefore, as per the general TSP, the salesman compulsorily visits all cities at least once depicted as vertices (or nodes) in a graph by selecting the shortest possible route illustrated as edges connecting the cities in various possible ways. In this way, TSP also has several special cases, such as multiple TSPs (mTSP), k-TSP, family TSP (FTSP), clusters TSP (CTSP), and precedence constraint-generalized TSP (PCGTSP), which are used in different scenarios [[→1](#), [→17](#), [→19](#), [→20](#), [→21](#), [→22](#), [→23](#)]. Even today, the use of TSP cannot be overlooked in our routine. Flight and parcel delivery route scheduling, medicine and food delivery, and so forth are the very recent applications of TSP. Therefore, the sustainable, cost-effective, and fast ways of route scheduling with minimal human involvement always remain an open mathematical problem.

But this graph computational problem has certain limitations like the initial worst-case complexity  $O(n!)$ , which was reduced to  $O(n^2 2^n)$  or  $O(n 2^n)$  by dynamic programming methods, to  $O(n^2 \log n)$  by greedy approach,  $O(n^2)$  or  $O(2^n)$  by branch and bound methods [[→24](#)]. Thus, the objective of every designed solution

for TSP and its variants is to continually reduce its computational complexity along with minimizing time and cost, by suggesting new and advanced solutions. Some of the other solutions are nearest neighbor methods, Brute force algorithms, multiagent systems, zero suffix method, Monte Carlo optimization, local search, meta-heuristic and hybrid approaches, local neighborhood optimization, genetic algorithms, and evolutionary strategies [[→1](#)]. Among these, the heuristic approaches for dealing with various kinds of TSP are widely used and praised. Generally, heuristic methods utilize reliable and convenient mental shortcuts learned from past experiences to narrow down the options in a scenario having diverse choices to easily and cognitively solve problems. However, they often result in irrational or inaccurate conclusions due to cognitive bias, incomplete knowledge of the problem, and mishandling of the problem.

## **2 Related work**

Numerous strategies have been put forth recently for solving various TSP variations. Some of them are discussed in this section. Erol and Bulut presented a real-world calculation of the TSP's route based on current traffic intensity data from Google Maps [[→25](#)]. Greedy, exhaustive, and heuristic A-star searches along with branch-and-bound algorithm and BitMask dynamic

programming were used in the development of the user interface for the identification of the shortest route via Google Maps, either in distance or duration, whichever opted by the user. Ismail [[→26](#)] demonstrated a Domino algorithm to solve TSP datasets of less than or equal to 100 cities, to reduce the calculation complexity of tour lengths. In contrast to random permutation in the global search of meta-heuristics that generates  $n!$  of prospective solutions, it only generates  $n^2.C(n, n^2)$  constructed tours.

A heuristic solution for the parallel drone scheduling TSP was suggested by Murray and Chu [[→27](#)] in 2015, which was then improved by Saleu and colleagues [[→28](#)] in 2018. To minimize the delivery time (or route completion time) for both drones and vehicles, where vehicles represented the classical delivery paths, and the drones were on the back-forth trips. The problem was treated with an iterative two-step heuristic comprising bilevel, that is, partitioning and route optimization. Coding segregated a solution into a customer sequence based on the availability of vehicles or drones, which was decomposed into an excursion for the vehicles and short trips for the drones in the deciphering step performed via dynamic programming using the bicriteria shortest path problem. Similarly, the randomized variable neighborhood descent (RVND) heuristic was demonstrated in [[→29](#)] as a solution for the collaborative

work of trucks and drones for delivering parcels. In contrast to the drone-less route, their hybrid heuristic approach reduced the total time taken in the delivery of goods by 19.28%. Besides this, they also minimized the transit period considering UAV battery life, cargo weight, and several visits in just three steps. These are the generation of optimal solutions by Concorde TSP Solver, the transformation of truck clients into UAV customers, and the improvement of the delivery route using the RVND algorithm. Again, Kitjacharoenchai et al. [[→30](#)] revisited the problem of the truck-drone combination of delivery problems, also known as the “multiple traveling salesman problem with drones” (mTSPD). Their idea was to let autonomous drones take off from delivery vehicles, conduct deliveries, and then land on any close by delivery truck. It will not only reduce the delivery times but also cut off the costs. For this, a mixed integer programming (MIP) adaptive insertion heuristic (ADI) approach was implemented. The mTSPD tours (or initial solutions) were constructed using a genetic algorithm (GA), combined K-means/nearest neighbor, and random cluster. Then gradually, a few truck-customer were replaced with drone-customer nodes, which were then evaluated and optimized by the ADI algorithm. Moreover, their method also keeps a record of customers served by drone. They compared the experimental results with [[→28](#)] and demonstrated a potential operational gain.

Mestria addressed the clustered traveling salesman problem (CTSP) signifying a warehouse problem [[→18](#)] with the iterated metaheuristics approach, namely, the VNRDGILS algorithm. It comprised variable neighborhood random descent, greedy randomized adaptive search, and deterministic rule (for local search). As compared to other such hybrid heuristics methods, their method had a low computational time. Moraes and Freitas addressed the moving target variant of TSP (MT-TSP), in which the location of cities changes with time, that is, dynamic [[→19](#)]. For such a crowd-monitoring application, they used a numeric movement prediction module for computing the length of tours by traversing them and simultaneously evaluating the new target as well. Three evolutionary approaches, viz. ant colony optimization (ACO), GA, and simulated annealing (SA) were used for generating solutions and optimizing the route. In terms of restricted time and processing power, the ideal results were achieved with GA, secondly with SA, and then with ACO. Identification of a viable solution for  $k$ -TSP requires selecting the origin together with a subset of  $k$  cities from  $n$  cities ( $k \leq n$ ), and a tour of these  $k$  cities via circular permutation. A similar problem was addressed by Pandiri and Singh [[→20](#)] by implementing two multistart heuristic approaches comprising general variable neighborhood search (GVNS) and hyper-heuristic (HH) algorithms. GVNS utilizes two neighborhood

structures with exchange-swap and variable neighborhood descent operations, whereas HH includes two low-level heuristics, that is, HH-random and HH-greedy selection mechanisms. The experimental analysis of the instances of TSPLIB revealed that the HH method was better than the GVNS tactic. Another version of TSP is the family traveling salesman problem (FTSP) in which the set of cities is separated into several families for computing the minimum-cost route for traversing a certain number of cities in each family.

Bernardino and Paias [[→21](#)] suggested two metaheuristics methods for this. One was a population-based local search method with GA, and the other was a hybridized technique including branch-and-cut with a local search technique. The latter approach comprises both constructive and improvement phases within a method, which took an average of 243 s on 1,002 nodes. However, on the same number of nodes, the population-based local search method with GA took an average computational time of 378 s. Similarly, another distinguished adaptation of the renowned generalized traveling salesman problem (GTSP) was termed precedence constrained generalized traveling salesman problem (PCGTSP), which was first mentioned by Dantzig and Ramsten [[→31](#)] for solving the supply problem for gas stations. Khachay, Kudriavtsev, and Petunin [[→22](#)] extended the generalized large neighborhood

search (GLNS) heuristic GTSP solver [[→32](#)] using a novel MILP model and the branch-and-bound method of Gurobi optimizer [[→33](#)], and evaluated its performance using instances in the PCGTSP LIB library.

Considering the requirement of additional methods to improve solutions and increase speed for local search, Costa et al. [[→34](#)] applied deep learning to solve the TSP. They suggested a policy gradient neural network to develop a local search heuristic relying on two-opt operators, which may be expanded to k-opt operators as well. Their work not only improved the generation of arbitrary initial solutions but also reached optimal solutions faster than other similar approaches. Another deep learning method combined with the traditional heuristic Lin-Kernighan-Helsgaun (LKH) to solve TSP was demonstrated by Xin et al. [[→35](#)]. Since the original LKH [[→36](#)] required hand-crafted rules for edge candidate selection, the problems with a large number of instances caused more time consumption. However, the NeuroLKH algorithm used the sparse graph network (SGN), which simultaneously yielded the edge scores, node penalties, and edge candidate set. To save time, rather than executing instance-wise iterative optimization, NeuroLKH used those node penalties as edge distances, which were ascertained during training. Experimental outcomes further revealed that NeuroLKH generalized well to much larger sizes, and could also

be applied to capacitated vehicle routing problems (CVRP), pickup and delivery problems (PDPs), and CVRP with time windows (CVRPTW).

The growth in the count of cities to visit turns TSP toward the NP-hard problem, which makes it difficult to optimize. Thus, for solving TSP comprising 29 to 7,397 cities, Alipour et al. [[→37](#)] hybridized the GA and multiagent reinforcement learning (MARL) heuristic. The initial population was generated by smart multipoint crossover, comprising GA and GA with a novel crossover operator, whereas the route construction and improvement were suggested by MARL. Their approach provided an efficient trade-off across CPU time and the quality of the solution. Dahiya and Sangwan [[→38](#)] reviewed the different approaches used for solving TSP. These included a vast discussion on methods of total enumeration, the efficient algorithm of Clarke and Wright, the branch-and-bound method, computer simulation, and evolutionary approaches like ACO, particle swarm optimization (PSO), and GA. Taillard suggested a linearithmic heuristic solution for TSP and tested it on more than 2 billion cities [[→39](#)]. This method used POPMUSIC (partial optimization metaheuristic under special intensification conditions) approach to generate initial solutions and to improve them too. Their method was designed to improve the 3D printing problem; however, the time consumed by the

printing head's nonproductive moves overshadowed its advantages.

After reviewing several works focusing on solving TSP with biheuristics, Rokbani et al. suggested a hybridized approach involving flower pollination algorithm (FPA), ACO with local search, ant supervised by flower pollination with local search (ASFPA-Ls), and ant supervised by PSO with local search (ASPSO-Ls) [[→40](#)]. To find the optimal solution for 2-opt TSP, their work proposed three variants of each ASFPA-Ls and ASPSO-Ls, which are cognitive, social, and chaotic. However, all three approaches have low error rates, where the social ASPSO-Ls attained an efficient trade-off between performance and time from the rest two on different test instances. Another improved version of ACO for solving TSP was demonstrated by Du et al. [[→41](#)], where an adaptive heuristic factor was utilized. Three improvements highlighted in AHACO were the use of K-means clustering for classifying cities and reducing search time, a 2-opt local optimizer for fine-tuning the solution, and an artificial bee colony optimization approach to escape from the local optimum. Moreover, the overall time complexity of AHACO on 39 instances was  $O(n^2)$ , which was found to be comparatively 83.33% better than other improvements on ACO. Nejad and Fazekas [[→42](#)] introduced a blend of the whale optimization algorithm (WOA) and K-means clustering for dealing with TSP.

This aids in identifying the best path with the lowest possible value of the fitness function, which means traversing through that path would be the least time-consuming. Thus, they divided the entire problem into the smallest possible clusters and implemented WOA for generating and optimizing solutions. And then rejoined those small clusters based on their Euclidean distances.

Another solution for the *m*TSP variant of TSP was suggested by Zheng et al. [[→43](#)] using an iterated two-stage heuristic algorithm (ITSHA). Their approach tried to simultaneously minimize both the overall duration of the expedition and the span of the longest excursion. This two-stage process generated initial solutions using fuzzy C-means (FCM) clustering and random greedy heuristic algorithms, which aid ITSHA in avoiding local optima. During the second phase, variable neighborhood search (VNS) was implemented to enhance the initial solution. Moreover, a record of the nearest cities was also kept in a candidate set for each city, arranged in ascending order, which probably reduced the search scope. The entire process is repeated until a possible best solution is identified.

From these, a scope for possible improvements can be identified in the drawbacks of [[→21](#), [→22](#), [→27](#), [→34](#), [→39](#)], mentioned as follows:

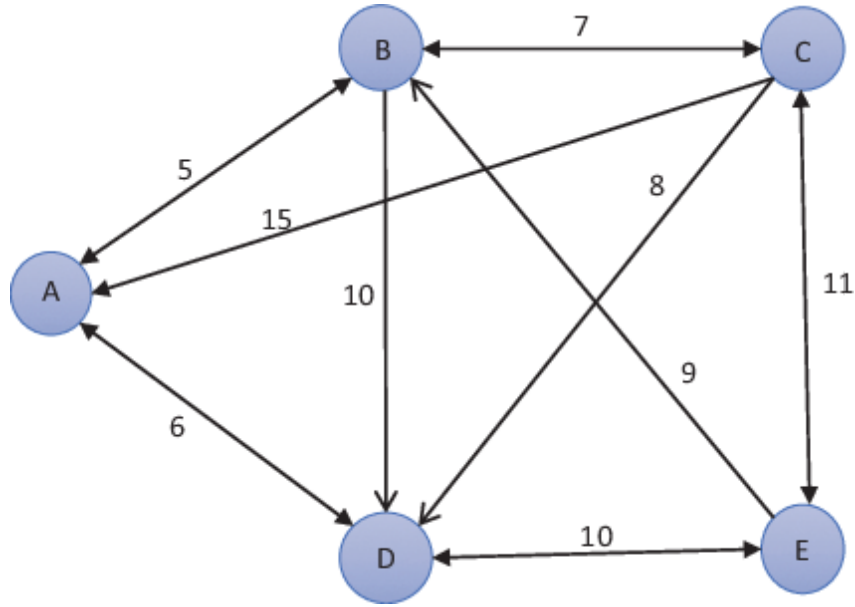
- Dodging iteration in a metaheuristic approach to reduce computation time, and lack of evolutionary approaches that would volunteer for natural candidates during the initial phase [[→27](#)].
- A deep learning approach, that is, policy gradient neural network [[→34](#)] was tested on a small number of samples, which may affect the tuning of parameters, and the optimized results as well.
- The extended version of the GLNS heuristic [[→22](#)] delayed the optimization of its general framework for integrating VNS and GA, which deviated it from parallel implementation in industry-inspired test instances.

Therefore, the goal of this chapter is to suggest an effective heuristic algorithm for scheduling the route of a milk van in an area with (i) 10 houses and (ii) 20 houses. The structure of the work comprises the mathematical model of the proposed work along with a few existing methods in Section 3. Results are discussed in Section 4 accompanied by the conclusion in Section 5.

## **3 Techniques and mathematical models**

### **3.1 The traveling salesman problem**

Generally, a TSP describes a salesperson who must visit each of the  $n$  cities at least once during the tour [[→44](#)]. The order in which he does so is not important as long as he/she visits each city during his/her trip, and finishes visits from where he/she started. In mathematical terms, each city is linked to further nearby cities via the edges. Each edge has a weight (or cost) assigned to it, which demonstrates the difficulty in traversing that path in terms of time or distance. Now, an optimized solution to such a problem is that during the tour, the salesman has to keep both travel distance and time as low as possible. For instance, [→Figure 1](#) portrays a weighted directed graph  $G(V, E)$  containing five vertices denoted as  $V = \{A, B, C, D, E\}$ , and edges  $E = \{AB, AD, BC, BD, CA, CD, CE, DE, EB\}$  with cost  $c_{ij} = \{5, 6, 7, 10, 15, 8, 11, 10, 9\}$  representing the distance from city  $i$  to city  $j$ , such that  $i, j \in V$ . It means traveling between cities  $A$  and  $B$  costs 5,  $A$  to  $D$  costs 6,  $B$  to  $C$  costs 7, and so on. In this way, the whole problem revolves around finding a tour of minimal length.



**Figure 1:** Directed graph representing a general traveling salesman problem.

From [→Figure 1](#), the probable traveling routes from  $A$  to  $A$  would be  $ABCEDA$ ,  $ADECBA$ ,  $ADEBCA$ , and  $ABDECA$  with the total length of the route (or cost of the tour) equivalent to 33, 39, 47, and 51, respectively. As evident, the shortest trip with minimum cost is  $ABCEDA$  (with cost = 33). In this way, a simple TSP with five nodes can be optimized. However, a similar solution cannot be implemented for more nodes as it will be difficult to handle the complexity of the problem then. Now, considering  $A \in V$  is the starting and ending point of the tour,  $c_{ij} \in E(i,j)$  is the cost of traveling from path  $i$  to path  $j$ . Mathematically, the minimum cost of the tour beginning from  $A$  and ending on  $A$ , where other than  $A$ , each city from set  $V$  appears exactly once [[→44](#)] would be given as follows:

$$G(A, V - \{A\}) = \min\{c_{ij} + G(j, V - \{A, j\}), \text{s. t. } i, j \in V\}$$

### 3.2 Nearest neighbor approach

Another popular heuristic to solve the TSP in polynomial time is simply the nearest neighbor approach (NN). In NN, the salesman starts from a particular node selected arbitrarily in graph  $G(V, E)$ , such that  $G(V, E)$  is a complete weighted graph, where  $|V| = n$ . Let the salesman start from city 1 and then select that city as the next city to visit in the set of unvisited cities, which is nearest to the current city, and so on. Now, assuming that he/she traverses a tour with possibly of minimum weight. The approach is very simple, but this may lead to disastrous solutions. Since once he/she visits all the cities none remains unvisited, he/she has to come back to the starting city, which he/she is bound to use, and this turns problematic if the weight of that edge is very high, and no other option is available.

### 3.3 Greedy approach

Another good but a little time costly method is Kruskal's and/or Prim's approach [[→44](#)]. In Kruskal's and/or Prim's algorithm, we determine the minimum spanning tree, which can be modified to find a Hamiltonian cycle with minimum weight in a provided weighted complete graph  $G(V, E)$ .

### 3.4 The proposed heuristic solution

In computer science and operation research, TSP is a popular NP-complete problem known to identify a Hamiltonian cycle of minimum possible weight. In a Hamiltonian cycle, each vertex of the graph is traversed exactly once, and the starting and terminating vertex is the same. Moreover, in the cycle, any vertex can be the starting vertex and the same vertex will be the terminating vertex, thus forming a complete graph. In a complete weighted graph  $G(V, E)$ , such that  $|V| = n$  denotes the total vertices (or nodes), then the total number of edges  $E$  can be evaluated as follows:

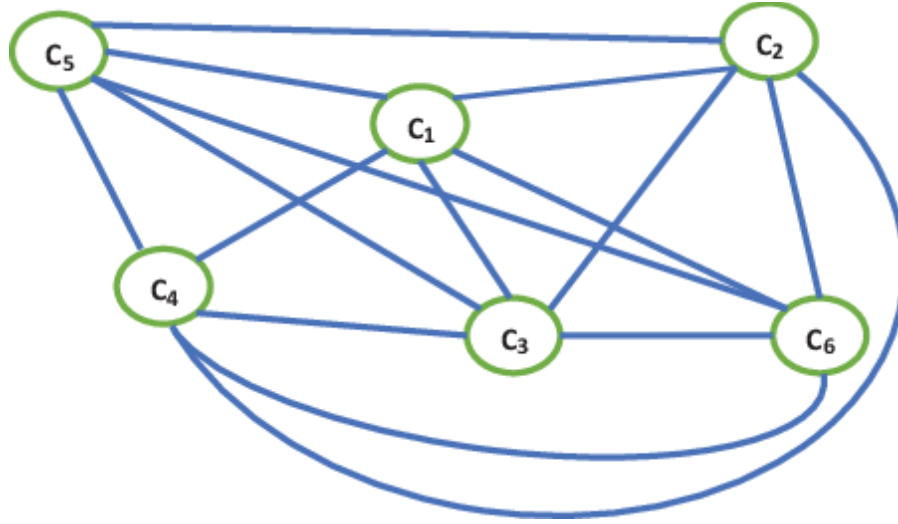
$$|E| = {}^nC_2 \quad (2)$$

Or simply,

$$|E| = \frac{n(n-1)}{2} \quad (3)$$

Now, with the Hamiltonian cycle of size  $n$ , the goal is to identify the Hamiltonian cycle with the lowest weight. Accordingly,

→[Figure 2](#) illustrates a complete weighted graph comprising six vertices  $\{C_1, C_2, C_3, C_4, C_5, C_6\}$  and 15 edges.



**Figure 2:** A complete weighted graph.

Again, for  $n$  vertices, the possible number of edges  $E$  computed using eq. (2) are sorted in ascending order by weights using quick sort technique [→45]. From this sorted set of routes, those routes that have a degree of less than or equal to 2 are assigned to another set named  $Hx$ . To fulfill the degree constraint, an adjacency matrix is created for each node (or vertex), such that if an edge exists between two vertices, then  $Adj_{ij} = 1$ , where  $Adj$  is  $n \times n$  adjacency matrix with  $i$  representing rows, and  $j$  denotes columns. Otherwise,  $Adj_{ij} = 0$ , in case no edge exists between two vertices. These adjacency matrices are created using the edges from set  $E$ , and the sum of elements in each row is calculated with each iteration. Now, if the rowwise sum of the elements in the adjacency matrix is greater than 2, then the vertices associated with those rows are discarded. Otherwise, if the rowwise sum of the elements in the adjacency matrix is less

than or equal to 2, the vertices related to that row are assigned to  $H_x$ . Therefore,  $H_x$  is the required collection of feasible routes, where the size of  $H_x$  is always equal to the number of vertices, that is,  $n$ . This algorithm is induced by Kruskal's algorithm to find a minimal spanning tree. The pseudo-code of the proposed algorithm is as follows:

### **Algorithm**

**Input:** A complete weighted graph  $G(V, E)$ , where  $|V|=n$ , and  $|E|=^nC_2$ .

**Output:** Set  $H_x$ , Edges in  $H_x$  composing the minimum cost tour (Hamiltonian cycle)

**Step 1:** Sort the edge set  $E$  with an efficient sorting method.

$e_{i1} \leq e_{i2} \leq e_{i3} \leq \dots \leq e_{i|E|}$ .

**Step 2:**  $ecounter=0; H_x = \phi; k=0;$

**Step 3:** while ( $ecounter < n$ ) do

$k=k+1;$

Select the Edge  $e_{ik}$

*If Inclusion of  $e_{ik}$  in  $H_x(H_x \cup \{e_{ik}\})$ , either increases the degree of any node in  $H_x$  more than 2 or formed a cycle with total edge length less than  $n$ , then:*

*discard edge  $e_{ik}$ .*

*else*

*append  $e_{ik}$  in set  $H_x$ , i.e.  $H_x \cup \{e_{ik}\}$ ;*

*ecounter= ecounter+1;*

*return  $H_x$ ;*

### **3.4.1 The heuristic function**

The heuristic function for  $H_x$  can be defined as follows:

$$h(x) = e - \sum_{e_i \in H_x} w(e_i) \quad (4)$$

where  $e = e_{i1} + e_{i2} + e_{i3} + \dots + e_{in}$ . The value of  $h(x)$  decreases stepwise with the iteration. Ideally, without constraints, the best solution would be the one for which  $h(x) = 0$ . But due to the imposition of constraint, that is, the degree of the selected node must not go beyond 2 and no subcycle formation with edge length less than  $n$ ; thus, the optimal solution would be the best feasible solution for which  $h(x)$  would be in close right *nhd* (neighborhood) of 0. If more than one Hamiltonian cycle

exists in the graph, then that Hamiltonian cycle was the optimal, which has heuristic  $h(x)$  much close to 0.

### 3.4.2 Analyzing the complexity of the proposed approach

Since we have sorted the set of edges using the quick sort method [[→45](#)], the time complexity of this operation would be  $O(|E| \log_2 |E|)$ , where  $|E| = {}^nC_2$ , or

$$|E| = \frac{n(n-1)}{2} \quad (5)$$

Thus, the time complexity can be calculated as follows:

$$\begin{aligned} O\left(\frac{n(n-1)}{2} \left(\log_2 \frac{n(n-1)}{2}\right)\right) \\ \frac{n(n-1)}{2} \in O(n^2) \\ \log_2 \frac{n(n-1)}{2} \in O(\log_2 n) \\ \log_2 n + \log_2 \frac{n(n-1)}{2} \in O(\log_2 n) \end{aligned}$$

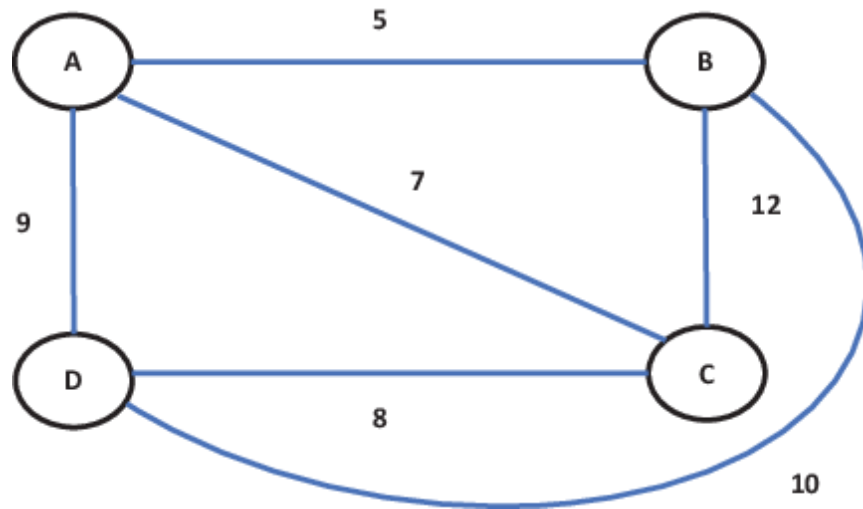
And so

$$\frac{n(n-1)}{2} \log_2 \frac{n(n-1)}{2} \in O(n^2 \log_2 n)$$

### 3.4.3 Assessing the feasibility of the suggested approach

Consider the TSP given in the weighted graph of [→Figure 3](#), with  $n = 4$  vertices, that is,  $\{A, B, C, D\}$ . Using eq. (2), the possible edges in this graph are 6, and the weights randomly assigned to each edge can be represented as  $\{(A, B, 5), (A, C, 7), (A, D, 9), (B, C, 12),$

$(B, D, 10), (C, D, 8)\}$ . At this point,  $(A, B, 5)$  signifies traveling from  $A$  to  $B$  where the weight of the edge is 5,  $(A, C, 7)$  denotes traveling from  $A$  to  $C$  where the weight of the edge is 7, and so forth for the remaining edges.



**Figure 3:** A complete weighted graph representing TSP.

Now, the sorted edge set  $E$  will be  $\{(A, B, 5), (A, C, 7), (C, D, 8), (A, D, 9), (B, D, 10), (B, C, 12)\}$ . Let  $Hx = \{\}$  is an empty set, and the global variable *ecounter* is initialized with 0. Then, with each iteration, we have

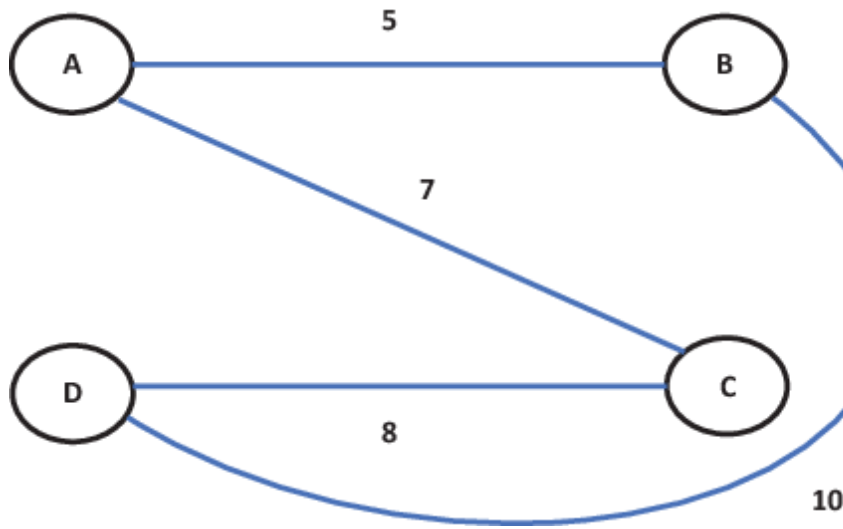
At *ecounter* = 1,  $Hx = \{(A, B, 5)\}$ ;

At *ecounter* = 2,  $Hx = \{(A, B, 5), (A, C, 7)\}$ ;

At *ecounter* = 3,  $Hx = \{(A, B, 5), (A, C, 7), (C, D, 8)\}$ ;

At *ecounter* = 4,  $Hx = \{(A, B, 5), (A, C, 7), (C, D, 8), (B, D, 10)\}$

As per the suggested algorithm, the edge from A to D with weight 9 is not added in  $Hx$  because it will violate the degree constraint. Thus, the final feasible route attained in four iterations is portrayed in [→Figure 4](#), which is  $Hx = \{(A, B, 5), (A, C, 7), (C, D, 8), (B, D, 10)\}$ , and the achieved Hamiltonian cycle is of length  $5 + 7 + 8 + 10 = 30$ .



**Figure 4:** The resultant tour computed by the presented approach.

## 4 Results and discussion

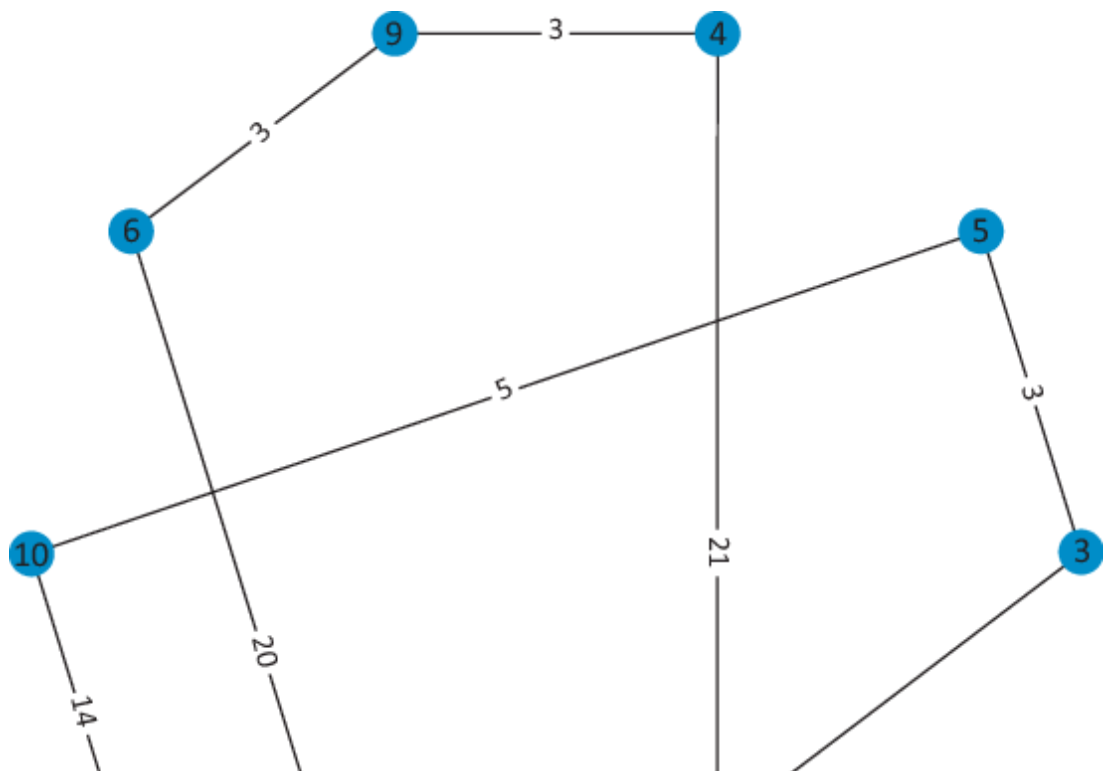
To test the feasibility of our proposed approach, we tried to trace the delivery route for a milk van in an area to supply milk to every house. The platform used to implement this method is Python version 3.11. Assuming that there are  $n$  houses in an area, where milk is to be delivered from a dairy. Then all the possible routes (i.e., edges) to each house can be computed using

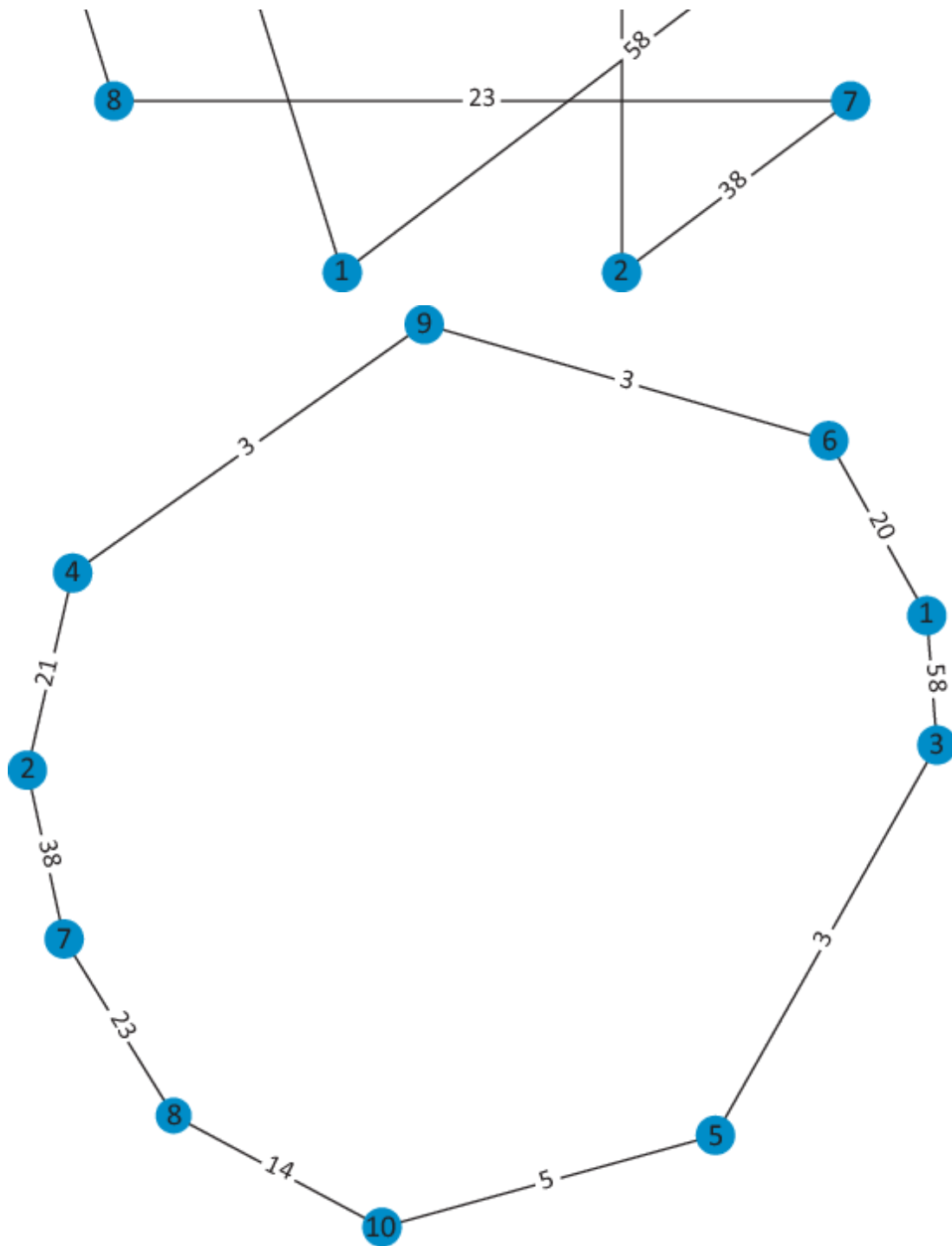
eq. (2). Once a collection of all possible routes are identified, they are sorted in ascending order by weights using quick sort. Apparently, the very first node with the smallest weight is the dairy itself, thus is first assigned to  $Hx$ . From the remaining routes, only those routes that abide imposed degree constraint of the presented technique have been assigned to  $Hx$  correspondingly. For this, as aforesaid, an adjacency matrix is created for each house representing the possible routes to the other neighboring houses. Thereafter, iteratively, if the rowwise sum of the elements in each of the adjacency matrices is greater than 2, then that route is not taken. Otherwise, if the same is less than or equal to 2, then that particular route is selected as one of the optimal routes and assigned to set  $Hx$ . Therefore,  $Hx$  comprises the collection of  $n$  possible routes for the milk van to deliver milk to each of the  $n$  houses, such that none of the houses are visited more than once.

#### **4.1 Identifying the optimal route of a milk van for 10 houses**

The ideal route for a milk van supplying milk to  $n = 10$  houses in an area is portrayed in [→Figure 5\(a\)](#) and [→\(b\)](#) in Fruchterman Reingold and circular layouts, respectively [[→46](#)]. In the circular layout, all the houses are placed in the form of a circle irrespective of distance, whereas the Fruchterman Reingold

layout places closer houses in the same neighborhood, and distant houses at distant locations. As obvious, if  $n = 10$ , then the total possible routes(or edges) will be  $E = 45$ , from which only 10 can be selected as the optimum solution. The randomly generated unsorted weights for each route are {(1, 1, 0), (1, 2, 58), (1, 3, 58), (1, 4, 69), (1, 5, 66), (1, 6, 20), (1, 7, 79), (1, 8, 84), (1, 9, 23), (1, 10, 80), (2, 2, 0), (2, 3, 97), (2, 4, 21), (2, 5, 13), (2, 6, 28), (2, 7, 38), (2, 8, 68), (2, 9, 25), (2, 10, 21), (3, 3, 0), (3, 4, 68), (3, 5, 3), (3, 6, 84), (3, 7, 64), (3, 8, 56), (3, 9, 75), (3, 10, 60), (4, 4, 0), (4, 5, 100), (4, 6, 64), (4, 7, 25), (4, 8, 56), (4, 9, 3), (4, 10, 44), (5, 5, 0), (5, 6, 23), (5, 7, 12), (5, 8, 40), (5, 9, 30), (5, 10, 5), (6, 6, 0), (6, 7, 49), (6, 8, 97), (6, 9, 3), (6, 10, 16), (7, 7, 0), (7, 8, 23), (7, 9, 25), (7, 10, 53), (8, 8, 0), (8, 9, 9), (8, 10, 14), (9, 9, 0), (9, 10, 48), (10, 10, 0)}.





**Figure 5:** TSP solution graph arranged in (a) the Fruchterman Reingold layout for  $n = 10$  houses and (b) the circular layout for  $n = 10$  houses.

After sorting the list of routes using the quick sort method, the first route assigned to  $Hx$  is a dairy node. From the remaining 44 routes, the 10 selected routes are  $Hx = \{(3, 5, 3), (4, 9, 3), (6, 9, 3), (5, 10, 5), (8, 10, 14), (1, 6, 20), (2, 4, 21), (7, 8, 23), (2, 7, 38), (1, 3, 58)\}$ , where the length of the Hamiltonian cycle (or cost of traveling) is 188 obtained in 0.1344 s.

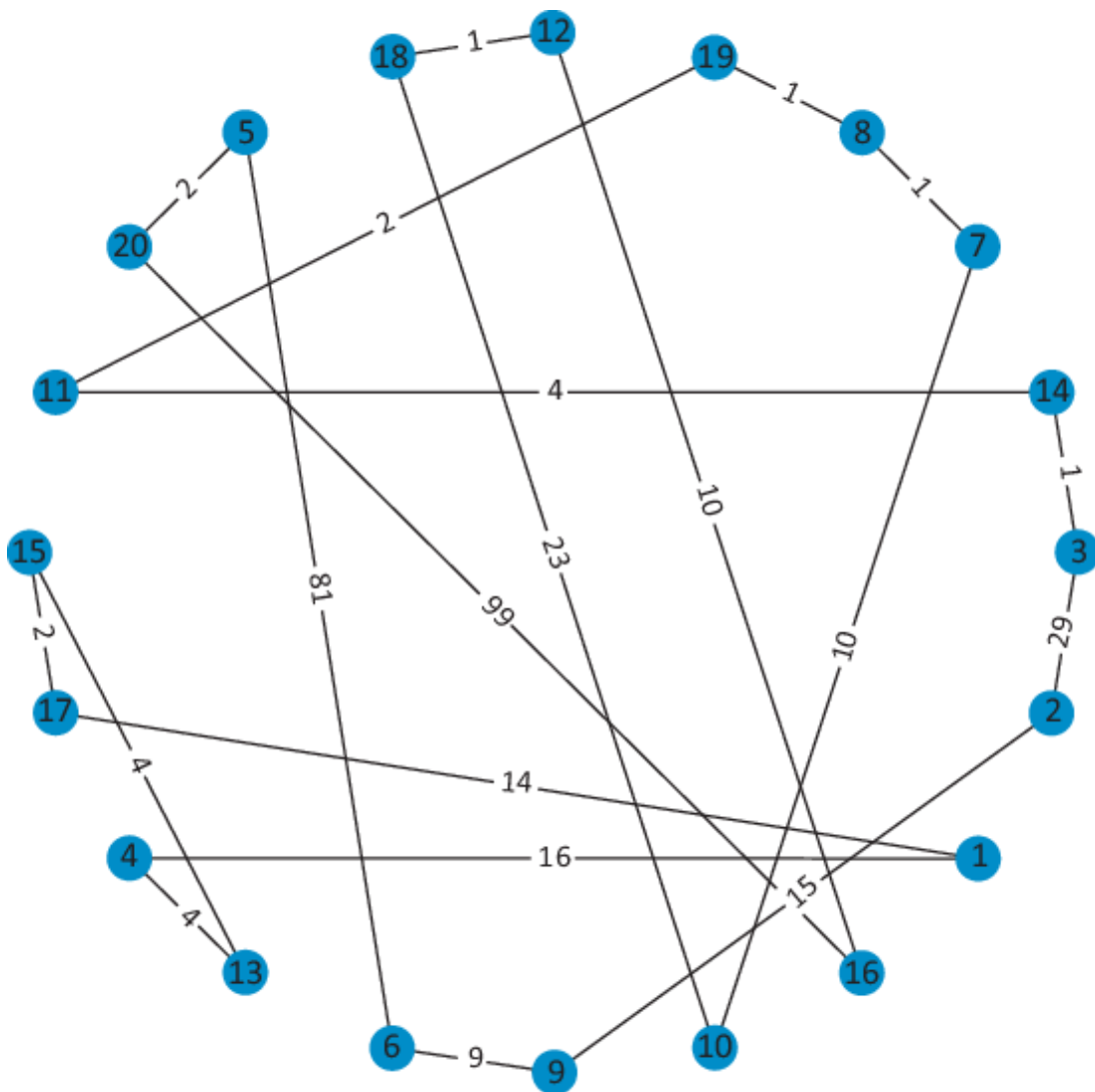
## 4.2 Identifying the optimal route of a milk van for 20 houses

Similarly, for  $n = 20$  houses,  $E = 190$  routes are possible, calculated using eq. (2). These randomly generated unsorted routes and their weights are  $\{(1, 1, 0), (1, 2, 46), (1, 3, 74), (1, 4, 16), (1, 5, 41), (1, 6, 30), (1, 7, 71), (1, 8, 70), (1, 9, 56), (1, 10, 81), (1, 11, 90), (1, 12, 42), (1, 13, 80), (1, 14, 6), (1, 15, 14), (1, 16, 40), (1, 17, 14), (1, 18, 46), (1, 19, 5), (1, 20, 77), (2, 2, 0), (2, 3, 29), (2, 4, 63), (2, 5, 70), (2, 6, 81), (2, 7, 67), (2, 8, 8), (2, 9, 15), (2, 10, 69), (2, 11, 26), (2, 12, 100), (2, 13, 95), (2, 14, 85), (2, 15, 60), (2, 16, 46), (2, 17, 93), (2, 18, 82), (2, 19, 27), (2, 20, 79), (3, 3, 0), (3, 4, 45), (3, 5, 85), (3, 6, 40), (3, 7, 58), (3, 8, 72), (3, 9, 95), (3, 10, 39), (3, 11, 28), (3, 12, 51), (3, 13, 29), (3, 14, 1), (3, 15, 69), (3, 16, 84), (3, 17, 98), (3, 18, 25), (3, 19, 35), (3, 20, 40), (4, 4, 0), (4, 5, 31), (4, 6, 27), (4, 7, 15), (4, 8, 82), (4, 9, 56), (4, 10, 76), (4, 11, 82), (4, 12, 53), (4, 13, 4), (4, 14, 47), (4, 15, 56), (4, 16, 81), (4, 17, 35), (4, 18, 78), (4, 19, 30), (4, 20, 85), (5, 5, 0), (5, 6, 81), (5, 7, 72), (5, 8, 15), (5, 9, 74), (5, 10, 87), (5,$

11, 59), (5, 12, 33), (5, 13, 62), (5, 14, 26), (5, 15, 95), (5, 16, 92), (5, 17, 68), (5, 18, 73), (5, 19, 6), (5, 20, 2), (6, 6, 0), (6, 7, 65), (6, 8, 45), (6, 9, 9), (6, 10, 91), (6, 11, 21), (6, 12, 13), (6, 13, 53), (6, 14, 70), (6, 15, 29), (6, 16, 88), (6, 17, 77), (6, 18, 97), (6, 19, 85), (6, 20, 97), (7, 7, 0), (7, 8, 1), (7, 9, 67), (7, 10, 10), (7, 11, 52), (7, 12, 79), (7, 13, 21), (7, 14, 66), (7, 15, 15), (7, 16, 86), (7, 17, 60), (7, 18, 91), (7, 19, 11), (7, 20, 22), (8, 8, 0), (8, 9, 29), (8, 10, 81), (8, 11, 93), (8, 12, 52), (8, 13, 46), (8, 14, 73), (8, 15, 48), (8, 16, 11), (8, 17, 55), (8, 18, 36), (8, 19, 1), (8, 20, 25), (9, 9, 0), (9, 10, 20), (9, 11, 85), (9, 12, 55), (9, 13, 25), (9, 14, 8), (9, 15, 47), (9, 16, 96), (9, 17, 61), (9, 18, 83), (9, 19, 58), (9, 20, 84), (10, 10, 0), (10, 11, 53), (10, 12, 55), (10, 13, 84), (10, 14, 85), (10, 15, 48), (10, 16, 86), (10, 17, 18), (10, 18, 23), (10, 19, 30), (10, 20, 24), (11, 11, 0), (11, 12, 63), (11, 13, 81), (11, 14, 4), (11, 15, 59), (11, 16, 87), (11, 17, 47), (11, 18, 8), (11, 19, 2), (11, 20, 66), (12, 12, 0), (12, 13, 16), (12, 14, 53), (12, 15, 90), (12, 16, 10), (12, 17, 70), (12, 18, 1), (12, 19, 64), (12, 20, 64), (13, 13, 0), (13, 14, 72), (13, 15, 4), (13, 16, 33), (13, 17, 24), (13, 18, 14), (13, 19, 45), (13, 20, 14), (14, 14, 0), (14, 15, 90), (14, 16, 19), (14, 17, 42), (14, 18, 95), (14, 19, 13), (14, 20, 82), (15, 15, 0), (15, 16, 7), (15, 17, 2), (15, 18, 49), (15, 19, 12), (15, 20, 85), (16, 16, 0), (16, 17, 46), (16, 18, 24), (16, 19, 76), (16, 20, 99), (17, 17, 0), (17, 18, 40), (17, 19, 96), (17, 20, 56), (18, 18, 0), (18, 19, 6), (18, 20, 65), (19, 19, 0), (19, 20, 43), (20, 20, 0)}.

Again, from the sorted list, the first route with minimal weight represents the dairy node. Now, out of 189 remaining routes, the

20 optimum routes of Hamiltonian cycle length 328 obtained in 0.2488 s are  $Hx = \{(3, 14, 1), (7, 8, 1), (8, 19, 1), (12, 18, 1), (5, 20, 2), (11, 19, 2), (15, 17, 2), (4, 13, 4), (11, 14, 4), (13, 15, 4), (6, 9, 9), (7, 10, 10), (12, 16, 10), (1, 17, 14), (2, 9, 15), (1, 4, 16), (10, 18, 23), (2, 3, 29), (5, 6, 81), (16, 20, 99)\}$ . The graph for the same case is portrayed in [→Figure 6](#) using the circular layout.



**Figure 6:** TSP solution graph arranged in the circular layout for  $n = 20$  houses.

## 5 Conclusion

The chapter studies the TSP and some of its renowned applications since 1932 and discusses recent heuristic solutions of TSP along with their benefits and drawbacks. Considering the vast scope and capability of the heuristic approaches, the chapter presents a simple and effective heuristic approach as the solution to the TSP. The approach suggests identifying  $n$  feasible routes from  $n^2C_2$  total routes using the degree constraint. It foretells that only those routes are selected in the set of feasible routes ( $Hx$ ), which have a degree less than or equal to 2. On implementing the presented approach to determine the traveling route of a milk van supplying milk to (i) 10 houses and (ii) 20 houses, from 45 and 190 routes, respectively. Ensuring that none of the houses is visited more than once, we obtained the optimal routes of Hamiltonian cycle lengths 188 in 0.1344s for (i), and 328 in 0.2488 s for (ii). Furthermore, the complexity of the algorithm is  $O(n^2 \log_2 n)$ , which is equivalent to the greedy approach and is free from the drawback of time consumption.

## References

[1] R. Matai, S. Singh and M. Lal, "Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution

Approaches. Traveling Salesman Problem,” in D. Davendra (ed.), Theory and Applications, 2010. doi: 10.5772/12909. [a](#), [b](#), [c](#), [d](#), [e](#)

**[2]** G. Bruno, A. Genovese and G. Improta, “Routing Problems: A Historical Perspective,” in M. Pitici (ed.), The Best Writing on Mathematics, Princeton: Princeton University Press, 2012, pp. 197–208. doi: 10.1515/9781400844678-021. [→](#)

**[3]** P. C. Mahalanobis, “A Sample Survey of the Acreage under Jute in Bengal,” in Proceedings of the second session of the Indian Statistical Conference, Lahore, 1939, pp. 73–42. [→](#)

**[4]** M. M. Flood, “The Traveling-Salesman Problem,” Operations Research, vol. 4, no. 1, pp. 61–75, 1956. [→](#)

**[5]** R. D. Plante, T. J. Lowe and R. Chandrasekaran, “The Product Matrix Traveling Salesman Problem: An Application and Solution Heuristics,” Operations Research, vol. 35, pp. 772–783, 1987. [→](#)

**[6]** M. Grötschel and O. Holland, “Solution of Large-scale Symmetric Traveling Salesman Problems,” Mathematical Programming, vol. 51, pp. 141–202, 1991. [→](#)

**[7]** R. E. Bland and D. E. Shallcross, “Large Traveling Salesman Problem Arising from Experiments in X-ray Crystallography: A

Preliminary Report on Computation," Operations Research Letters, vol. 8, no. 3, pp. 125–128, 1989. ➡

**[8]** R. D. Angel, W. L. Caudle, R. Noonan and A. Whinston, "Computer Assisted School Bus Scheduling," Management Science, vol. 18, pp. 279–288, 1972. ➡

**[9]** I. A. Vakhutinsky and L. B. Golden, "Solving Vehicle Routing Problems Using Elastic Net," Proceedings of the IEEE international conference on neural network, pp. 4535–4540, 1994. ➡

**[10]** A. E. Carter and C. T. Ragsdale, "Scheduling Pre-printed Newspaper Advertising Inserts Using Genetic Algorithms," Omega, vol. 30, pp. 415–421, 2002. ➡

**[11]** T. Zhang, W. A. Gruver and M. H. Smith, "Team Scheduling by Genetic Search," Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials, vol. 2, pp. 839–844, 1999. ➡

**[12]** K. H. Kim and Y. Park, "A Crane Scheduling Method for Port Container Terminals," European Journal of Operational Research, vol. 156, pp. 752–768, 2004. ➡

**[13]** X. Wang and A. C. Regan, "Local Truckload Pickup and Delivery with Hard Time Window Constraints," *Transportation Research Part B*, vol. 36, pp. 97–112, 2002. [→](#)

**[14]** H. D. Ratliff and A. S. Rosenthal, "Order-Picking in a Rectangular Warehouse: A Solvable Case for the Travelling Salesman Problem," *Operations Research*, vol. 31, pp. 507–521, 1983. [→](#)

**[15]** K. S. Ruland and E. Y. Rodin, "The Pickup and Delivery Problem," *Computers and Mathematics with Applications*, vol. 33, no. 12, 1997. [→](#)

**[16]** H. A. Saleh and R. Chelouah, "The Design of the Global Navigation Satellite System Surveying Networks Using Genetic Algorithms," *Engineering Applications of Artificial Intelligence*, vol. 17, pp. 111–122, 2004. [→](#)

**[17]** R. Van Dal, *Special Cases of the Traveling Salesman Problem*, Groningen: Wolters-Noordhoff, 1992. [a](#), [b](#)

**[18]** L. Tang, J. Liu, A. Rong and Z. Yang, "A Multiple Traveling Salesman Problem Model for Hot Rolling Scheduling in ShangaiBaoshan Iron & Steel Complex," *European Journal of Operational Research*, vol. 124, pp. 267–282, 2000. [a](#), [b](#)

**[19]** M. Mestria, "New Hybrid Heuristic Algorithm for the Clustered Traveling Salesman Problem," *Computers & Industrial Engineering*, vol. 116, pp. 1–12, 2018. doi: 10.1016/j.cie.2017.12.018. [a](#), [b](#)

**[20]** R. S. De Moraes and E. P. de Freitas, "Experimental Analysis of Heuristic Solutions for the Moving Target Traveling Salesman Problem Applied to a Moving Targets Monitoring System," *Expert Systems with Applications*, 2019. doi: 10.1016/j.eswa.2019.04.023. [a](#), [b](#)

**[21]** V. Pandiri and A. Singh, "Two Multi-start Heuristics for the K-traveling Salesman Problem," *Opsearch*, vol. 57, pp. 1164–1204, 2020. doi: 10.1007/s12597-020-00463-8. [a](#), [b](#), [c](#)

**[22]** R. Bernardino and A. Paias, "Heuristic Approaches for the Family Traveling Salesman Problem," *International Transactions in Operational Research*, vol. 28, no. 1, pp. 262–295, 2020. doi: 10.1111/itor.12771. [a](#), [b](#), [c](#), [d](#)

**[23]** M. Khachay, A. Kudriavtsev and A. Petunin, "PCGLNS: A Heuristic Solver for the Precedence Constrained Generalized Traveling Salesman Problem," in N. Olenov, Y. Evtushenko, M. Khachay and V. Malkova (eds.), *Optimization and Applications: Lecture Notes in Computer Science*, vol. 12422, Cham: Springer, 2020, pp. 196–208. doi: 10.1007/978-3-030-62867-3\_15. [→](#)

**[24]** E. Angelelli, C. Bazgan, M. G. Speranza and Z. Tuza, "Complexity and Approximation for Traveling Salesman Problems with Profits," Theoretical Computer Science, vol. 531, pp. 54–65, 2014. doi: 10.1016/j.tcs.2014.02.046. ➡

**[25]** M. H. Erol and F. Bulut, "Real-time Application of Travelling Salesman Problem Using Google Maps API," Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT), 2017. doi: 10.1109/ebbt.2017.7956764. ➡

**[26]** A. H. Ismail, "Domino Algorithm: A Novel Constructive Heuristics for Traveling Salesman Problem," IOP Conference Series: Materials Science and Engineering, vol. 528, 2018. doi: 10.1088/1757-899X/528/1/012043. ➡

**[27]** C. C. Murray and A. G. Chu, "The Flying Sidekick Traveling Salesman Problem: Optimization of Drone-assisted Parcel Delivery," Transportation Research Part C: Emerging Technologies, vol. 54, pp. 86–109, 2015. [a](#), [b](#), [c](#)

**[28]** R. G. Mbiadou Saleu, L. Deroussi, D. Feillet, N. Grangeon and A. Quilliot, "An Iterative Two-step Heuristic for the Parallel Drone Scheduling Traveling Salesman Problem," Networks, 2018. doi: 10.1002/net.21846. [a](#), [b](#)

**[29]** J. C. De Freitas and P. H. V. Penna, "A Randomized Variable Neighborhood Descent Heuristic to Solve the Flying Sidekick Traveling Salesman Problem," *Electronic Notes in Discrete Mathematics*, vol. 66, pp. 95–102, 2018. doi: 10.1016/j.endm.2018.03.013. ➡

**[30]** P. Kitjacharoenchai, M. Ventresca, M. Moshref-Javadi, S. Lee, J. M. A. Tanchoco and P. A. Brunese, "Multiple Traveling Salesman Problem with Drones: Mathematical Model and Heuristic Approach," *Computers & Industrial Engineering*, 2019. doi: 10.1016/j.cie.2019.01.020. ➡

**[31]** G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959. ➡

**[32]** S. L. Smith and F. Imeson, "GLNS: An Effective Large Neighborhood Search Heuristic for the Generalized Traveling Salesman Problem," *Computers and Operations Research*, vol. 87, pp. 1–19, 2017. doi: <https://doi.org/10.1016/j.cor.2017.05.010>. ➡

**[33]** Gurobi Optimization, LLC: Gurobi optimizer reference manual, 2020. ➡ <http://www.gurobi.com> ➡

**[34]** P. R. D. O. Costa, J. Rhuggenaath, Y. Zhang and A. Akcay, "Learning 2-opt Heuristics for the Traveling Salesman Problem

via Deep Reinforcement Learning,” SN Computer Science, vol. 2, no. 388, 2021. doi: 10.1007/s42979-021-00779-2. [a](#), [b](#), [c](#)

**[35]** L. Xin, W. Song, Z. Cao and J. Zhang, “NeuroLKH: Combining Deep Learning Model with Lin-Kernighan-Helsgaun Heuristic for Solving the Traveling Salesman Problem,” in M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang and J. Wortman Vaughan (eds.), Advances in Neural Information Processing Systems, vol. 34, Curran Associates Inc, 2021, pp. 7472–7483.

[→https://proceedings.neurips.cc/paper\\_files/paper/2021/file/3d863b367aa379f71c7afc0c9cdca41d-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/3d863b367aa379f71c7afc0c9cdca41d-Paper.pdf). [→](#)

**[36]** K. Helsgaun, “An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic,” European Journal of Operational Research, vol. 126, no. 1, pp. 106–130, 2000. [→](#)

**[37]** M. M. Alipour, S. N. Razavi, M. R. F. Derakhshi and M. A. Balafar, “A Hybrid Algorithm Using A Genetic Algorithm and Multiagent Reinforcement Learning Heuristic to Solve the Traveling Salesman Problem,” Neural Computing and Applications, vol. 30, pp. 2935–2951, 2018. doi: 10.1007/s00521-017-2880-4. [→](#)

**[38]** C. Dahiya and S. Sangwan, “Literature Review on Travelling Salesman Problem,” International Journal of Research, vol. 5, no. 16, pp. 1152–1155, 2018. [→](#)

**[39]** É. D. Taillard, "A Linearithmic Heuristic for the Travelling Salesman Problem," *European Journal of Operational Research*, vol. 297, no. 2, pp. 442–450, 2022. doi: 10.1016/j.ejor.2021.05.034.

[a](#), [b](#)

**[40]** N. Rokbani, R. Kumar, A. Abraham, A. M. Alimi, H. V. Long, I. Priyadarshini, et al, "Bi-heuristic Ant Colony Optimization-based Approaches for Traveling Salesman Problem," *Soft Computing*, vol. 25, pp. 3775–3794, 2021. doi: 10.1007/s00500-020-05406-5. ➡

**[41]** P. Du, N. Liu, H. Zhang and J. Lu, "An Improved Ant Colony Optimization Based on an Adaptive Heuristic Factor for the Traveling Salesman Problem," *Journal of Advanced Transportation*, vol. 2021. doi: 10.1155/2021/6642009. ➡

**[42]** A. S. Nejad and G. Fazekas, "Solving a Traveling Salesman Problem Using Meta-heuristics," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 1, pp. 41–49, 2022. ➡

**[43]** J. Zheng, Y. Hong, W. Xu, W. Li and Y. Chen, "An Effective Iterated Two-stage Heuristic Algorithm for the Multiple Traveling Salesmen Problem," *Computers & Operations Research*, vol. 143, 2022. doi: 10.1016/j.cor.2022.105772. ➡

**[44]** E. Horowitz and S. Sahni, "The Traveling Salesperson Problem," in *Fundamentals of Computer Algorithms*, New Delhi:

Galgotia, 1992, pp. 231–234. [a](#), [b](#), [c](#)

**[45]** T. H. Cormen, C. E. Leiserson, R. Rivest and C. Stein, “Chapter 7: Quicksort,” Introduction to Algorithms, 4th ed, Cambridge: MIT Press, 2022, pp. 182–204. [a](#), [b](#)

**[46]** NetworkX, “Drawing,” Network Analysis in Python.  
[→https://networkx.org/documentation/stable/reference/drawing.html](https://networkx.org/documentation/stable/reference/drawing.html) →

# Assessment of fake news detection from machine learning and deep learning techniques

**Megha Shah**

**Akshay Kumar**

**Shristi Kharola**

**Mangey Ram**

---

**Acknowledgments:** This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the editor and anonymous reviewers for their comments that help improve the quality of this work.

---

## **Abstract**

The overwhelming prevalence of news data across numerous online platforms is far less taxing, but it has become an unnecessary burden that has transformed people's lives. The dominance of mass media on the internet has had a substantial impact, with many individuals depending on it as a regular source of information. However, this widespread influence has also given rise to the creation of partially evident or entirely fabricated news stories. Intentionally spreading these fake

stories through online social networking sites has become a common practice. Websites now primarily aim to mold public opinion using false information. The core aim of this study is to develop a reliable model that can identify a given news report as true or false. For this, the authors have modeled a trust-based architecture for online shared news incorporating natural language processing in machine learning (ML) and deep learning (DL) techniques. To develop the architecture, six ML models, two long-short-term memory (LSTM) models, and two distinct feature extraction techniques have been utilized. The findings reveal that, out of all six ML models, random forest with TF-IDF and logistic regression with CountVectorizer yield the optimal results. In the case of DL models, the outcomes for the LSTM model and bi-directional LSTM have yielded the same results.

**Keywords:** Classification algorithm, fake news detection, machine learning, natural language processing,

## 1 Introduction

The amount of false news stories has been rising quickly. Although it is not a brand-new issue, it has recently seen significant growth. Wikipedia defines fake news as untrue or misleading material that is presented as news [[→1](#)]. Finding false news has been a difficult and complicated undertaking.

Humans have been found to have a propensity to trust false information, which makes the dissemination of false information much simpler. According to reports, the capacity of humans to recognize fraud without additional aid is about 54% [[→2](#)]. Fake news is hazardous because it readily deceives individuals and stirs up misunderstandings within society. This may have further negative effects on society. The dissemination of false information fuels rumors that circulate about, and the sufferers can suffer significantly. Fake news has gained a lot of attention since 2017 [[→3](#)]. The majority of information is obtained through these ways since it is free and accessible from anywhere at any time. The absence of accountability in this data, as opposed to the usual means of obtaining information like newspapers or other reliable sources, makes it less trustworthy. This study addresses the problem of identifying bogus news. Fake news problems emerge in the diplomatic, economic, and political spheres as an unresolved issue in social networks' data and information-consuming application layer and as a severe and difficult challenge in information progress.

Propaganda, which is created specifically for political reasons, is another topic connected to information extraction. The language used in fabricating fake news is particularly cunning in that it is intended to stir up and exacerbate users' emotions in order to distribute false information. Recent reports

indicated that the US Presidential Elections had been harmed by the increase of false news that was being produced online. People and organizations that are operating in their own interests or the interests of third parties may produce fake news. According to a poll by a nongovernmental group, there are many bogus accounts and material circulating the social networking site using appropriate channels for expressing both emotions and exchanging opinions. To offer the data center more room and handle the chaos and political issues in the situation, it is necessary to remove the damaging and undesired accounts from the network [[→4](#)]. Since users spend a lot of time on social media and most people prefer online sources of information, it is now challenging to determine the veracity of the news.

The realistic aspect of social media is the variety of shared information. The disclosure of false information highlights the wasteful use of network resources. Additionally, it comprises the completeness and legitimacy of the content based on the service that is offered [[→5](#)]. Because of this, using the quality of trust model to news delivery is relevant [[→6](#)]. The level of security required for everyday social media networking is improved by machine learning text categorization. The capacity to identify fake news is based on a content analysis of the disseminated information's veracity [[→7](#), [→8](#), [→9](#)]. There is a

need for an autonomous solution for the extraction of false news due to the increase of noisy and unstructured data, users, and news [[→10](#), [→11](#)]. Based on current advancements in artificial intelligence (AI), deep learning (DL), and machine learning (ML), these words become constrained. As a case study, we gathered the social media material from Facebook and Twitter, two well-known sites for sharing information, where millions of news articles and postings are posted every day on a range of subjects by thousands of users.

The main aim of this study is to combine techniques from ML and DL to authenticate false data and developers. To develop the model, the authors utilized natural language processing (NLP) to evaluate these approaches. Authors also contrasted various text vectorization techniques to select the one that produces the best results. The remainder of the chapter is organized as follows: Section 2 provides a literature background on the work on fake news detection. Section 3 describes the methodology used in the construction of the model, followed by the data analysis and results in Section 4. Section 5 draws conclusions from the analysis.

## **2 Literature survey**

Recent technological advancements and the usage of apps in everyday life have led to a mess on numerous social media platforms due to publishing and sharing of undesired and meaningless situations. One of the social media sites that was mentioned is Twitter, which has a sizable user base and daily shares millions of tweets on a range of subjects and phrases [[→12](#), [→13](#)]. In order to identify and highlight language trends in terms of fake or authentic news, majorly ML algorithms and NLP approaches have been utilized [[→14](#)]. The majority of the ML process involves classifier models, which can distinguish between phony and true information. [→Table 1](#) shows various kinds of false news available on the internet online.

**Table 1:** Some various types of fake news.

Mock or ridicule	Some websites produce fake news stories in an amusing attempt to spoof the media, but if they are circulated out of context, they can be misleading.
Untrue data that is somewhat true but provided in the wrong context	Real facts that are selected carefully and promoted to generate headlines, but which usually show an incorrect understanding of scientific inquiry.
Shoddy reporting that furtheres a certain agenda	News that includes unsubstantiated truth cubes and is used to support a specific perspective or position.
Imposter data	When legitimate sources are spoofs.
Manipulated	When actual news or images are

material

misrepresented.

False news that supports a persistent narrative but isn't found on facts

News when there is no clear-cut definition of what is true, when contradictory viewpoints or opinions are regularly stated, and when unconscious biases are prevalent. Conspiracy theories typically fall short here.

Deliberately misleading

News that is made with the intent to cause conflict or cause confusion. These stories are frequently spread by means of phony news websites or other venues that pass for "real" media. The usage of violent videos and digitally manipulated images is common.

By counting word vectors, Vijay et al. [[→15](#)] employed a random forest (RF) approach and NLP to identify false news. A survey on the use of DL and NLP for the identification of false news was provided by Chokshi and Mathew [[→16](#)]. For detection, a variety of DL techniques were offered based on vast amounts of data. A false news detector framework employing an amalgam of all three elements – a fake news detector, reinforcement

teaching, and annotator – was proposed by Wang et al. [[→17](#)]. This procedure was used to remove the article's poor labels and select samples of exceptional quality to detect false news. The absence of confidence in this procedure is what separates the related works from the currently suggested effort. Gilda [[→18](#)] examined the predicted percentage and assessed several ML techniques. The accuracy of several prediction models, such as support vector machines, gradient boosting (GB), and constrained decision tree (DT) models, was measured. Utilizing imperfect probability criteria, the mathematical models were evaluated. Qun et al. [[→19](#)] talked about rumor detection and countering erroneous data in the present moment. It employed novelty-based functionality and obtained its data on Kaggle. The model's precision rating was 74.5%. Since sensational and shady sources are not taken into account, accuracy was decreased. Aphiwongsophon and Chongstitvatana [[→20](#)] used multiple ML algorithms for false news detection. The naive Bayes (NB) classifier and support vector machine were the models for ML that were offered and implemented. No particular precision was noted because only the models were covered. By employing an NB classifier, Granik and Mesyura [[→21](#)] established a straightforward method for the identification of false news. With a dataset of social news posts, they tested it. Additionally, they used the Vox news database. On

the assessment set, they obtained an accuracy rate for classification of almost 73%.

Jain and Kasbe [[→22](#)] discussed the use of NB classifier to identify bogus news on several social media platforms. Twitter, Facebook, and other sites were the information resources for news articles. Since the information on these sites is not entirely reliable, the precision gained was relatively poor [[→23](#)]. In Twitter, it is used to distinguish between fraudsters and faux spammers. A straightforward false news sensor with excellent precision for classification tasks was created by Conroy et al. [[→24](#)]. They applied network evaluation and language clues techniques to it. Both strategies use ML to develop classifiers that match the analysis. Their 72% rate of accuracy was higher. This might be accomplished by doing cross-corpus evaluation on the models used for classification as well as by reducing the total length of the feature that was input vector. Barua et al. [[→25](#)] used an ensemble method using recurrent neural networks (RNNs) long-short-term memory (LSTM) and gated recurrent unit to determine if a news story was accurate or deceptive. The integrity of a news story may also be determined using an Android app. On a big dataset that they had created in their research, they tested their model. This method's restriction was that a specific size of the article was needed. If the article was not sufficient to provide a summary, it

would make incorrect predictions. Sentiment was a crucial factor employed by Bhutani et al. [[→26](#)] to increase the effectiveness of spotting false news. Three separate datasets were used. They employed cosine similarity, the TF-IDF vectorizer, the CountVectorizer, and the bi-grams and tri-grams approaches. NB and RF were the techniques used to train the model. They assessed the model using several performance criteria. Their accuracy was 81.6%. A rumored identification system that assesses the veracity of details and categorizes it as a rumor or not was proposed [[→27](#)]. The automated identification of bogus information in Portuguese was proposed by Monteiro et al. [[→28](#)]. Their method was based on identifying the linguistic feature using ML and automated detection algorithms in the system that is currently being used. Parikh and Atrey [[→29](#)] used a variety of mediums to examine the news article's dependability. The research provides information on the many available content types as well as how news stories are categorized. The models used in the article were less effective than the alternatives, including models based on language characteristics and predictive modeling. Pérez-Rosas et al. [[→30](#)] identified false news using a variety of techniques. Since the employment of the language model was promoted, the correctness rate was restricted to 76%.

As studied above, distributing false news is a big problem; hence, it is really important to identify it when it happens. Furthermore, the issue of identifying false news is highly challenging, and several academics are working to find a solution. Therefore, this study's primary goal is to develop a model that is used to determine whether or not a piece of news is authentic, using alternate techniques from ML and DL, in order to obtain higher accuracy in results. Two separate datasets (dataset 1 and dataset 2) from past information have been taken into account that will be combined to create a master dataset which will aid in learning the models and determine if the piece of information is false or true. Further, in the study, fake news dataset 1 is classified using ML models and fake news dataset 2 is classified using DL models.

### **3 Methodology**

The methodology utilizes NLP techniques in ML and DL models to construct a trust-based architecture for online shared news. These approaches are further explained below.

#### **3.1 Natural language processing (NLP)**

NLP refers to a method of AI that uses a language like English to communicate with an intelligent machine. An intelligent

machine, such as a robot, has to be able to process natural language in order to follow your commands [[→31](#)]. Some basic terminologies of NLP are stated as given below:

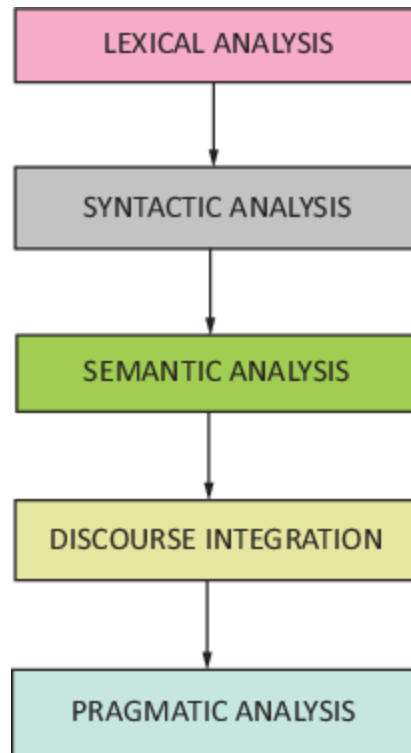
1. Phonology: The systematic study of sound.
2. Morphology: The study of how words are created from simple meaningful building blocks.
3. Morpheme: A basic linguistic unit of meaning.
4. Syntax: The process of placing words in a sentence.
5. Semantics: The meaning of words and putting them together to form meaningful and logical sentences and phrases.
6. Pragmatics: This subject examines how sentences are used and understood in a variety of contexts and how this affects the interpretation of phrases.
7. Discourse: It examines how the sentence that comes just after it may influence how the next statement is understood.
8. General knowledge of the world is included in word knowledge.

NLP involves the following steps and it is also represented in [→Figure 1](#):

1. *Lexical analysis*: It is the process of identifying and analyzing word structures. The entire collection of words and phrases

used in a language makes up its vocabulary. Lexical analysis divides the entire text into paragraphs, phrases, and words.

2. *Parsing*: It is often referred to as syntactic analysis and is the process of analyzing the grammar of a phrase's words, arranging them in such a way that their links to one another are obvious. The English structural analyzer disapproves of phrases like "The tree comes to the kid."
3. *Semantic analysis*: This method determines the text's exact meaning or definition from a dictionary. The text's applicability is evaluated. Syntactic components and task-specific objects are mapped to achieve this. The semantic analyzer ignores phrases like "beautifully ugly girl."
4. *Discourse integration*: Any statement's meaning is influenced by the context of the sentence that comes right before it. Additionally, it helps clarify the meaning of the statement that follows it.
5. *Pragmatic analysis*: In this process, what was stated is reinterpreted to determine its true meaning. It entails determining those features of language that need knowledge of the outside world.

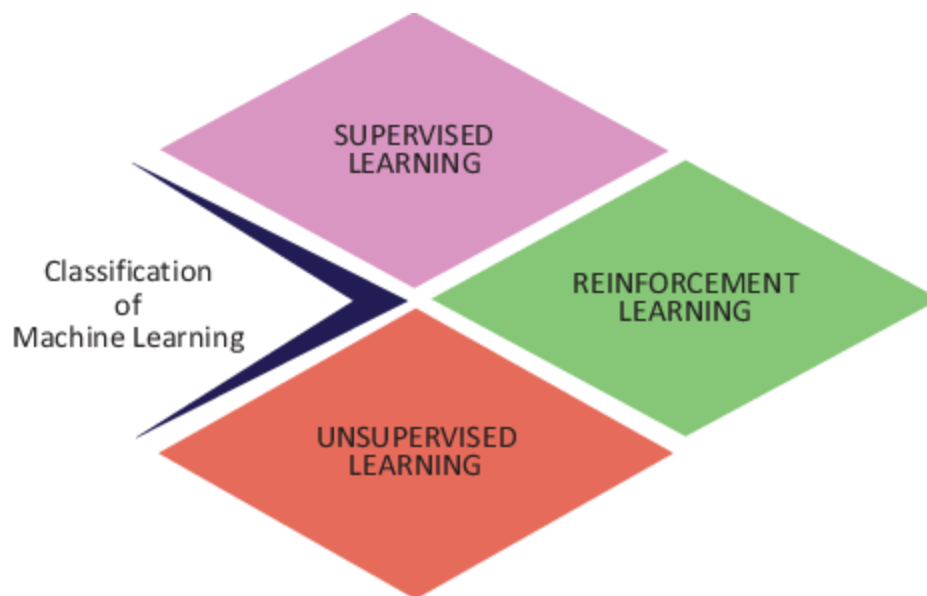


**Figure 1:** Steps in NLP.

### 3.2 Machine learning (ML)

ML is a growing approach that allows computers to acquire knowledge from past data automatically. It uses a number of approaches to build mathematical models and predict outcomes based on prior knowledge or data. It is now used for a variety of purposes including recommender systems, false news detection, Instagram auto-tagging, and photo identification. ML, according to some, is a branch of AI that focuses on creating algorithms that let a computer learn on its own using data and previous experiences. ML methods use “training data,” or previous sample data, to create a mathematical model that

helps with judgements or predictions without being explicitly programmed. To build prediction models, ML is used with disciplines like statistics and computer science [[→32](#)]. Generally, there are three categories into which ML may be divided, as shown in [→Figure 2](#).



**Figure 2:** Types of ML algorithms.

### 3.2.1 Supervised learning

The ML model is fed sample labeled data as training material, and it then utilizes this knowledge to predict the outcome. To interpret the databases and learn about each one, the system constructs a model utilizing data with labels. Following training and processing, the model is evaluated using sample data to determine whether it properly predicts the intended output. To

further characterize supervised learning, two types of algorithms might be utilized:

1. Classification
2. Regression

### **3.2.2 Unsupervised learning**

It is a sort of learning in which a computer learns without any human interaction. The machine is trained on unclassified, unlabeled, or uncategorized data, and the algorithm must reply independently on that data. Unsupervised learning has no predetermined conclusion. The computer sifts through massive amounts of data in quest for useful insights. It is also possible to split algorithms into two types:

1. Clustering
2. Association

### **3.2.3 Reinforcement learning**

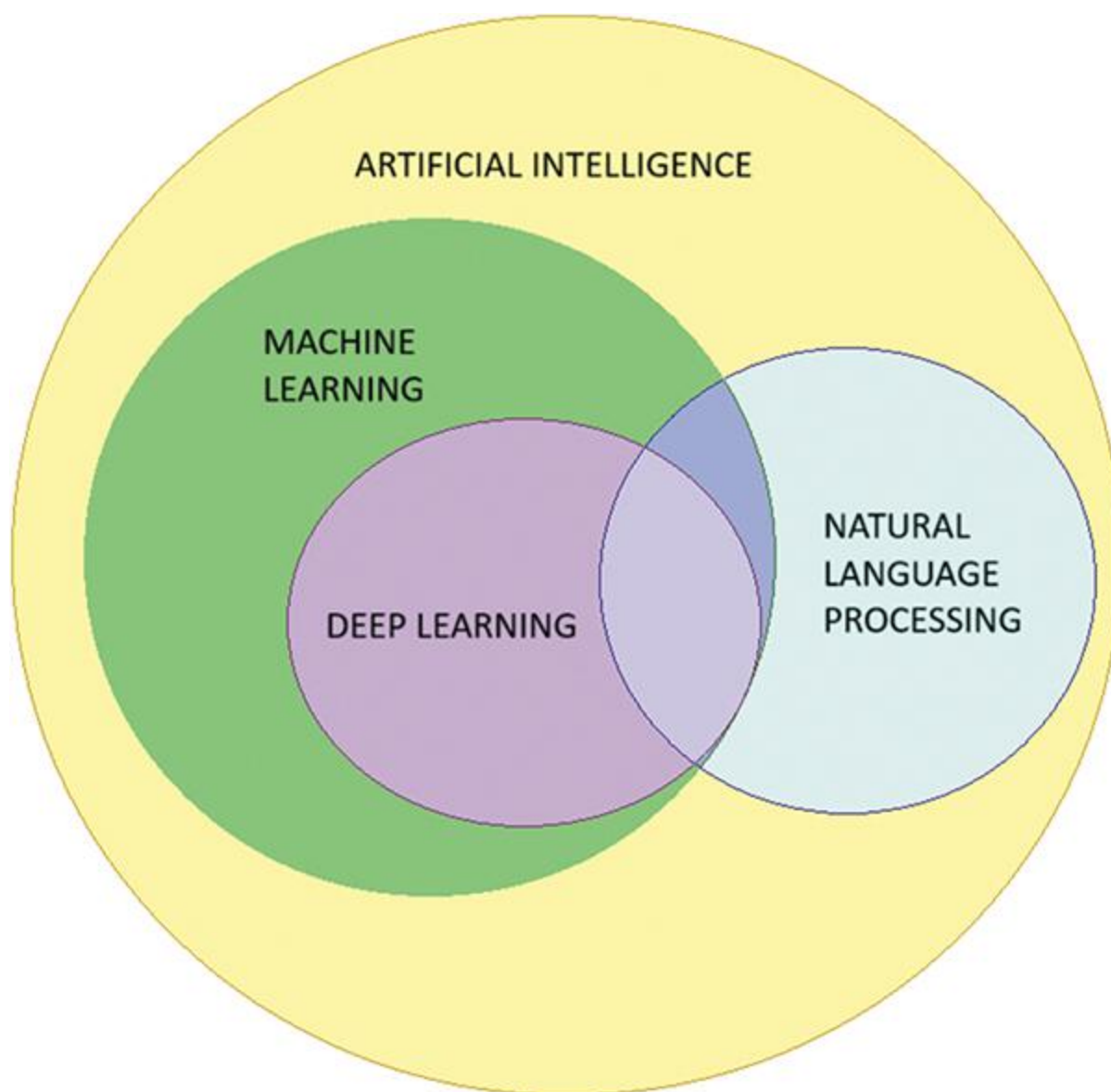
It is based on comments and reviews. In a reinforcement learning system, a learning agent obtains an incentive for each correct action and a penalty for each erroneous action. The agent automatically learns and improves as a result of these feedbacks. During reinforcement learning, the agent

investigates and interacts with the environment. An agent performs better as its goal is to accumulate the most reward points. A robot cat which automatically learns how to use its arms exemplifies reinforcement learning.

### **3.3 Deep learning (DL)**

AI and ML models that use DL describe how people learn certain sorts of information. Imagine a little toddler who says “cat” as her first word to get a sense of DL. The youngster understands what a cat is by the use of the phrase “cat” and pointing at various objects. If a parent sees a cat, they will either say, “Yes, that is a cat,” or “No, that is not a cat.” As he continues to point to numerous objects, the small child gains more knowledge about the traits that all cats have. The youngster unwittingly simplifies a complex abstraction – the concept of cat – by building a hierarchy where each new level of abstraction is built utilizing information from the previous layer of the hierarchy. Data science, which also covers statistics and predictive modeling, is fundamentally dependent on DL. It speeds up and simplifies this process, which is especially helpful for those working in the data science sector who are responsible for collecting, analyzing, and interpreting massive volumes of data [[→33](#)]. DL may be thought of as a method for automating predictive analytics at its most fundamental level.

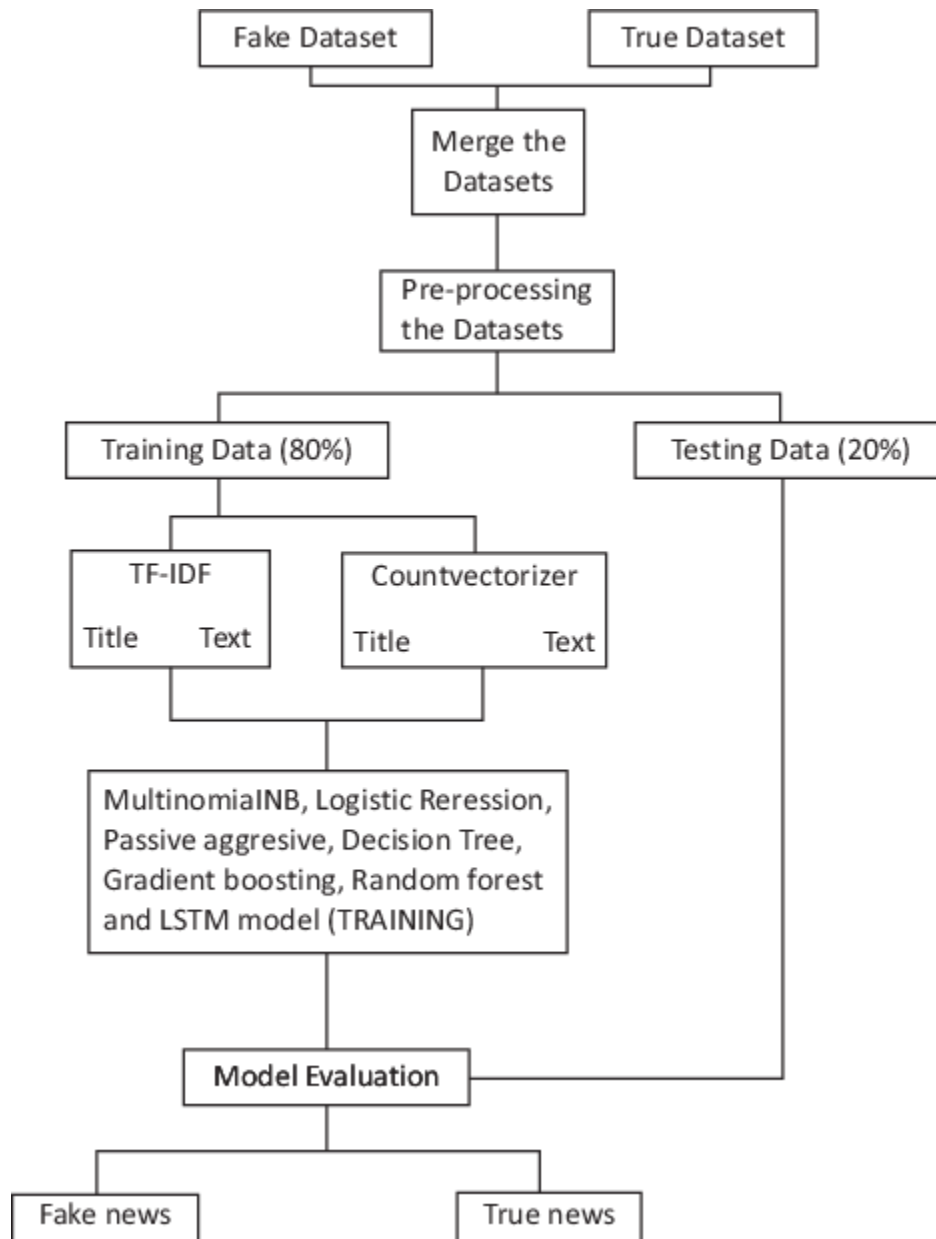
DL algorithms are layered in structures of escalating complexity and abstraction in contrast to the linearity of classical ML algorithms. →[Figure 3](#) shows the relationship between AI, ML, DL, and NLP with the help of a Venn diagram.



**Figure 3:** Venn diagram showing relationship among ML, DL, NLP, and AI.

### 3.4 Study framework

In this work, six related ML models are suggested that will be taught and fed with various text vectors of news title and news content in order to solve the challenge of identifying false news. This will assist in developing a solid model that can determine if the news is accurate or false. Two LSTM models are discussed in this research, and two distinct feature extraction techniques have been applied on two fresh datasets. The information sets are initially gathered using past references. After this, two datasets are combined to create a single master dataset, after which, the dataset is refined. In order to achieve the same size, the databases must undergo prepping that includes reducing the data, stop word removal, stemming, tokenization, and padding. Following this, the whole set is divided into testing and training data. The main process diagram of the suggested system framework for false news identification is shown in →Figure 4.



**Figure 4:** Flow chart for fake news classification.

### 3.5 Datasets

Four datasets have been used in this project out of which two are true datasets and two are fake datasets. The data is mainly true and fake news. True news is gathered from the website of

*Hindustan Times* and fake news are collected from Kaggle. True dataset 1 and Fake dataset 1 are used for ML models whereas True dataset 2 and Fake dataset 2 are used for DL models.

→Tables 2 and →4 represent True dataset 1 and True dataset 2, on the other hand →Tables 3 and →5 represent Fake dataset 1 and Fake dataset 2, respectively.

**Table 2:** True dataset 1 [[→34](#)].

Title	Subject	Date	Category
AAP gets “national party” status; Trinamool, National...	The 2012-founded AAP dominated the Punjab assembly elections early last year and five seats as well.	April 10, 2023	National
“Nothing less than a miracle”: Kejriwal as ...	The Election Commission said that AAP has been named as a national ... .	April 10, 2023	National

Title	Subject	Date	Category
The Gramin Dak Sevaks of Odisha have received their mark sheets.	According to S Barik, the Dept. of Post selected 1,380 individuals for branch work based on their matriculation test scores. ... .	April 10, 2023	National
Maharashtra reports 328 Covid cases in 24 h ... .	In the last 24 h, 247 patients recovered overall, with a 98.12% recovery rate ... .	April 10, 2023	National
On Azad's revelation about Rahul-Himanta episode ... .	Ghulam Nabi Azad's tirade against Congress leader Rahul Gandhi has triggered a political slugfest ... .	April 10, 2023	National

**Table 3:** Fake dataset 1 [[→35](#)].

Title	Text	Date	Subject
Pope Francis just made a statement.	In his yearly Christmas Day letter, Pope Francis chastised ... .	December 23, 2022	News
Racist Alabama Cops Brutalize Black ... .	The total amount of incidents where police have killed and brutalized persons of color ... .	December 23, 2022	News

Title	Text	Date	Subject
Fresh Off The Golf Course, Trump Lashes ... .	He spent a significant amount of the day playing his golf in the club, the 84th day in a row.	December 23, 2022	News
Papa John Founder Retires, Figures Out Racism Is Bad ... .	A centerpiece of Donald Trump s campaign, and now his presidency, has been his ... .	December 23, 2022	News

Title	Text	Date	Subject
A successor to Disney Empire is aware that GOP swindled us – SHREDS ... .	A princess with brass ovaries named Abigail Disney stands to gain from the GOP tax scheme. ... .	December 23, 2022	News

**Table 4:** True dataset 2 [[→34](#)].

Title	Text	Date	Category
Power sharing means ... : Cong ... .	The Congress on Thursday named Siddaramaiah as the new ... .	April 23, 2013	National
Justice should be served to all: Arjun ...	Meghwal was assigned the law and justice ministry as a ... .	April 23, 2013	National
Karnataka's secure future ... our top ... .	As Congress formally made its announcement on the new ... .	April 23, 2013	National
Delhi court orders FIR against woman ... .	Acquitting the man and his family members, the court noted ... .	April 23, 2013	National

Title	Text	Date	Category
“Stalker” arrested outside Prince Harry ... .	Kevin Garcia Valdovinos, 29, was allegedly seen “lurking” ... .	April 24, 2013	International

**Table 5:** Fake dataset 2 [[→35](#)].

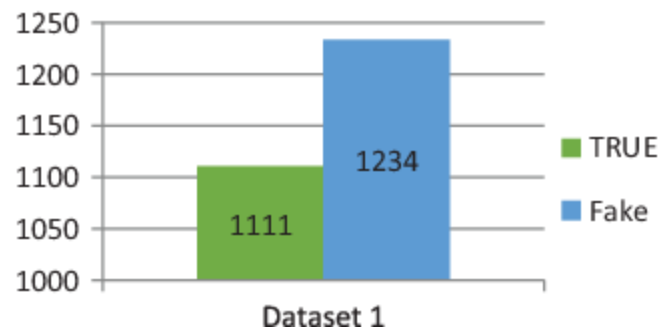
Title	Text	Date	Subject
Chaos Ensues After Man Accidentally ... .	Maybe thoughts and prayers aren't the remedy following a ... .	January 30, 2023	News
New Accuser Confirms She Got ... .	Alabama Senate Candidate Roy Moore (R- of course) has ... .	January 30, 2023	News
Roy Moore Is Asking People To Snitch ... .	Roy Moore is desperate to paint all these child molestation ... .	January 30, 2023	News
The Internet Lights Up After The ... .	On Thursday, the Pentagon's official Twitter account ... .	January 30, 2023	News

Title	Text	Date	Subject
Anti-Gay GOP Rep Gets BUSTED For ... .	Republicans are hypocrites and Ohio state GOP ... .	January 30, 2023	News

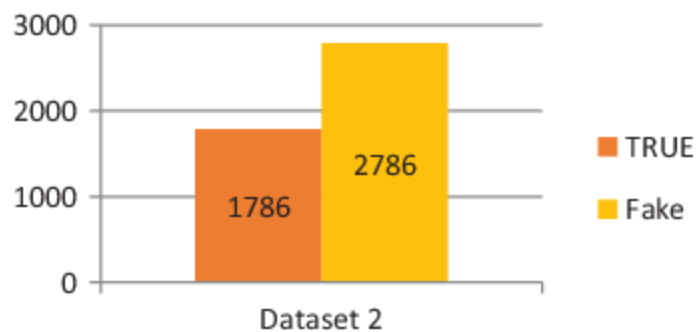
→[Table 6](#) shows the size of both datasets in tabular form as well as in bar chart formgraph as shown in →[Figures 5](#) and →[6](#). True dataset 1 has 1,111 news and fake dataset 1 has 1,234 news as input. True dataset 2 and fake dataset 2 have 1,786 and 2,786 number of news, respectively.

**Table 6:** Total number of news in each dataset.

Information	Dataset 1		Dataset 2	
Total number of news in each datasets	True	Fake	True	Fake
	1,111	1,234	1,786	2,786



**Figure 5:** Size of news dataset 1.



**Figure 6:** Size of news dataset 2.

### 3.6 Data cleaning steps (preprocessing the dataset)

Preprocessing refers to the adjustments made to the data before feeding it into the computation. The preprocessing of texts is a method of getting ready and cleaning up text data for use in NLP. Real-world knowledge is likely to be full of inaccuracies since it frequently lacks specific actions or patterns and is irregular or missing. Preprocessing information is a tried-and-true technique for handling these problems. In a ML project, preprocessing is essential for improving the model's performance, and the data preparation must be lawful. Before

the data is accurately represented by various assessment models, it must first go through multiple changes. We may reduce the quantity of genuine data by removing any extraneous data from the data [[→36](#), [→37](#), [→38](#)].

The text of the article and headline was tokenized using *NLTK* in Python. Lemmatizing the remaining portion of the data was made easier by eliminating the stop words (using the NLTK stop-word list). The subsequent processing processes were used to create the labeled text list for each course. The Punkt (an unsupervised trainable model) expression tokenizer from the library called NLTK to create tokens for the body and headline was used. This tokenizer can recognize sentence punctuation marks and the placement of words in a statement because it uses an unsupervised ML algorithm that has been trained on a generic English corpus. Each sample is tagged with the tokens from the complete headline and body sets.

### **3.6.1 Train-test split**

To design a training set that is successful, one must comprehend the issue that is being addressed. For instance, what results from the ML computation are anticipated. ML commonly uses the training and test informative sets. Each of the two should randomly evaluate a larger variety of data. The main set in use

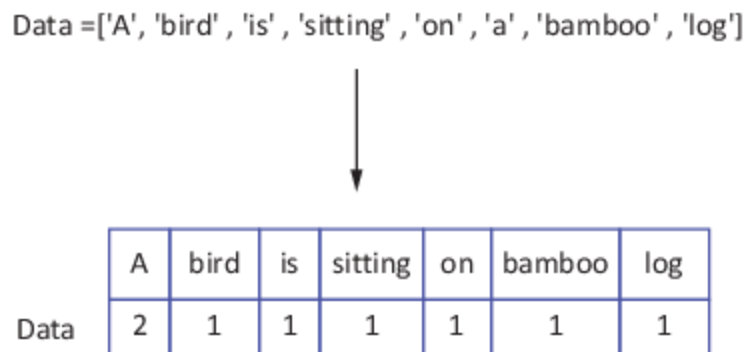
is the training set, which is the bigger of the two sets. An ML framework that makes use of a training set shows how to balance different highlights by changing their coefficients in accordance with how probable they are to eliminate mistakes in the results. Since they include the structure of data elements that the algorithm is being trained on, tensors comprising such variables, also known as coefficients, are collectively termed as the model's components. These are the most crucial lessons discovered when creating a machine learning system. The second set contains the test set. It acts as a seal of approval and is not utilized until the final end. Once the neural network has been developed and given input, it may be tested against this last arbitrary test. The outcomes it produces should attest to the fact that the internet at least recalls [x] number of photos with accuracy. Return to the training sample and look at the mistakes that were made if, in the unlikely occasion, correct predictions are not made. There won't be any problems if the right dataset is utilized, and everything will work as it should. The data will then be divided into test and training information as the following phase in the process [[→39](#)]. In this case, the split is 80% train information and 20% test information.

### **3.6.2 Feature extraction**

The extraction of features is the subsequent stage. ML algorithms are taught a predetermined set of characteristics from the training data in order to provide results for the test data. However, the main problem with processing languages is that unprocessed text cannot be directly handled by ML algorithms. Therefore, specific feature extraction methods are required to convert text into a matrix (or vector) of properties. There will be two methods for extracting features:

1. TF-IDF (term frequency/inverse document frequency) is an acronym for document frequency-inverse term frequency. It focuses attention to a specific issue that might not arise frequently, but is highly important. The TF-IDF value increases in proportion to the frequency of a textual expression and decreases in proportion to the total number of texts that utilize the phrase. The TF-IDF technique is one of the most used information retrieval methods for figuring out how important a word is in a text. TF-IDF is the outcome of TF and IDF. A phrase will have an elevated TF-IDF score if it occurs frequently in one text but infrequently throughout the corpus of documents. The TF-IDF also approaches zero as the IDF value for a phrase that appears in virtually all texts hits zero. The TF-IDF value is large if both the IDF and TF values are large, or if a phrase is prevalent in one part but uncommon elsewhere in the document.

2. CountVectorizer-text must be tokenized or processed to exclude specific terms if one wants to utilize linguistic information for predictive modeling. Then, in order to be used as submits in algorithms for ML, these words need to be encoded as floating point or integers-point values. The vectorization of features is the method in question. Scikit-learn CountVectorizer is used to convert a group of text documents into a vector of term/token counts. Additionally, it enables preprocessing text data prior to vector representation creation. Due to its features, it may express text qualities in a variety of ways [[→40](#)]. An example of how CountVectorizer works is shown in [→Figure 7](#).



**Figure 7:** Example of working of CountVectorizer.

## 3.7 Models utilized in the study

### 3.7.1 Logistic regression (LR)

A classification technique called as logistic regression (LR) is used to assign observations to a limited number of groups. LR alters the findings in contrast to linear regression, which produces continuous numerical values, by regenerating a value for probability that may then be moved to at least two distinct groups utilizing the calculated sigmoid capacity. To determine the optimum weights provided to the training set, the LR model employs gradient descent. The sigmoid function is the hypothesis that underlies our model [[→41](#)]:

$$z = b_0 + b_1x, \quad z \text{ is a linear model} \quad (1)$$

$$y = 1/(1 + e^{(-z)}), \quad y \text{ is a logistic model} \quad (2)$$

### 3.7.2 Multinomial naive Bayes (NB)

NB is a learning technique that is commonly employed in text categorization applications since it can be effective in computation and easy in implementation. There are two typical event models:

- Bernoulli multivariate event model
- Multivariate event model

Multinomial NB is another name for the multivariate event model. Texts are often categorized using the multinomial NB method after being subjected to statistical examination of their

contents. It significantly streamlines the classification of textual material and offers a feasible substitute for “heavy” AI-based semantic analysis [[→42](#)].

The formula used to compute it is given as follows:

$$P(A|B) = P(A) \times P(B|A)/P(B) \quad (3)$$

While predictor  $B$  has already been provided, we calculate the likelihood of class  $A$ .

$P(B)$  is the prior probability of  $B$ ,  $P(A)$  is the prior probability of class  $A$ , and  $P(B|A)$  is the occurrence of predictor  $B$  given class  $A$  probability.

### 3.7.3 Passive aggressive classifier (PAC)

For classification problems, an ML technique known as passive aggressive classifier (PAC) is used. This approach, which falls under the category of online learning algorithms, can handle large datasets and adjusts its model in reaction to new information. Since the PAC method is an online learning virtual system, its weights may change as new data is received. The magnitude of the margin and the quantity of misclassifications can be traded off using the regularization parameter,  $C$ , of the PAC algorithm. At each iteration, PAC looks at a fresh occurrence, assesses whether it was correctly

detected or not, and then adjusts its weights accordingly. If the case is correctly classified, the weight doesn't change. To improve the classification of succeeding examples, the PAC algorithm updates its weights depending on the mistakenly classified instance if it is erroneously categorized. How much the PAC algorithm updates its weights depends on the regularization parameter  $C$  and the degree of confidence in the classification of that particular occurrence [[→43](#)].

### **3.7.4 Decision tree (DT) classifier**

Although the supervised learning technique called a DT may be used to address classification and regression issues, this approach is generally chosen. It is a classifier having a tree-like structure, with internal nodes representing dataset attributes, branches representing decision-making procedures, and each leaf node representing the classification conclusion [[→44](#)]. It runs judgments or tests based on the features of the supplied dataset. It offers a visual picture of all viable answers for a certain problem or option based on established circumstances. It is called as a DT because it starts at its root node and spreads on successive branches to form the structure of a tree. A DT simply asks a question and divides into subtrees based on the response (yes/no).

### **3.7.5 Gradient boosting (GB) algorithm**

GB is a popular boosting technique in ML for both classification and regression applications. Boosting is a form of ensemble learning method that teaches the model repeatedly, trying to get better with each new model. It turns a number of poor learners into effective ones. AdaBoost and GB are the two most often used boosting algorithms. In contrast with AdaBoost, every predictor is taught using the antecedent's residual faults or errors as labels rather than altering the training examples' weights. Classification and regression trees is a technique that GB trees employs as its foundation learner. The essential premise of this method is to generate models sequentially while aiming to minimize the faults of the previous model [[→45](#)].

### **3.7.6 Random forest (RF) classifier**

Renowned and notable ML algorithm – RF is a part of the supervised learning methodology. It might be used to resolve ML issues involving both regression and classification. It is a method of combining several classifiers to tackle complicated problems and enhance model performance and is based on the idea of ensemble learning. The RF classifier, as its name suggests, uses a large number of sets of DTs based on different subsets of the information being used and chooses the mean or

average to increase the dataset's predictive power. Instead of depending just on one DT, the RF uses predictions from each tree and determines the result based on the votes of the majority of predictions. Higher precision and overfitting are prevented by the growing number of trees in the forest [[→46](#)].

### **3.7.7 The long-short-term memory network (LSTM)**

Numerous approaches have been used to discuss disappearing and exploding gradients in deep RNNs. The LSTM is one of the best known of these. Theoretically, an LSTM recurrent unit aims to “forget” unimportant input and “remember” every previous piece of information that the network has seen up to this moment. To do this, a number of “gates” – activation function layers used for various goals – are introduced. The data that the previous LSTM recurrent unit decided to keep track of is theoretically described by the internal cell state vector, which each LSTM recurrent unit likewise monitors. LSTM networks are the most often used variety of RNNs. The memory cell and gates, which include the forget gate as well as the input gate, are the two most important parts of the LSTM. The input gates and forget gates regulate the inner contents of the memory cell [[→47](#)].

### **3.7.8 Bidirectional LSTM**

The sequence processing model known as a BiLSTM, often referred to as a bidirectional LSTM, consists of two LSTMs, one of which gets input forward and the other of which receives it backward. BiLSTMs successfully boost the network's capacity for information access, which improves the context of the algorithm [[→48](#)].

## 4 Results and analysis of datasets

The accuracies of all the models to determine fake news have been achieved and have been discussed in this section. We have used confusion matrix that tells about the performance of any algorithm through visualization which makes it easier to understand the results.

### 4.1 Results for dataset 1

For the body of our dataset, a network of words has been made. A word cloud is a brand-new type of visual representation of content data that is frequently utilized to locate metadata (catchphrase labels) on websites or to imagine free-form text. [→Figures 8](#) and [→9](#) represent word cloud for dataset 1.



The accuracy, precision, recall, and  $F1$  score of the first dataset has been achieved using the aforementioned techniques, namely the multinomial NB, passive aggressive classifier, logistic regression, DT classifier, GB classifier, and RF classifier. Precision, recall,  $F1$  score, and accuracy are used to gauge performance.

*Accuracy:* It displays the overall accuracy of correctly categorized cases relative to the total number of occurrences. The formula used to compute it is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

*Precision:* It displays the proportion of amusing headlines. In other words, it compares the proportion of headlines labeled as sardonic to the overall proportion of such headlines. The formula used to compute it is as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \cdot \quad (5)$$

*Recall:* It shows the proportion of satirical titles that have been googled. In other words, the number of headlines that are often categorized as sardonic was evaluated versus the overall number of sarcastic headlines. The formula used to compute it is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \cdot \quad (6)$$

→[Table 7](#) tells about the accuracy, precision, recall, and *F1* score of dataset 1 for all six ML models. These six different models along with two different feature extraction techniques give 12 outcomes.

**Table 7:** Results using ML classification models.

Machine learning models	Accuracy	Precision	Recall	F1 score
Multinomial NB (CountVectorizer)	0.853	0.940	0.665	0.775
Passive aggressive classifier (CountVectorizer)	0.871	0.760	0.974	0.855
Logistic regression (CountVectorizer)	0.923	0.880	0.929	0.900
Decision tree classifier (CountVectorizer)	0.912	0.889	0.884	0.888

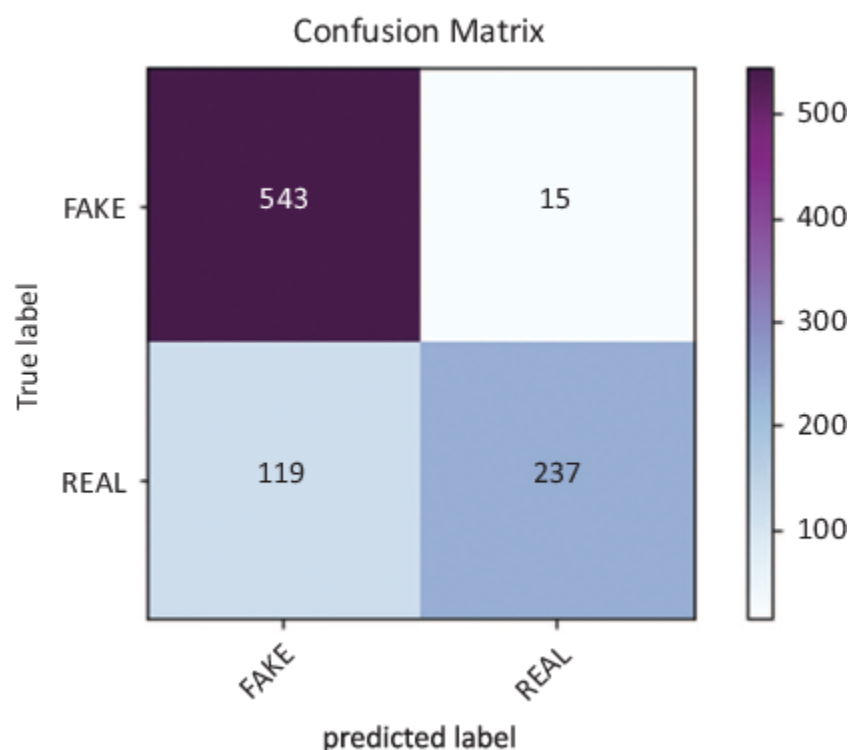
Machine learning models	Accuracy	Precision	Recall	F1 score
Gradient boosting classifier (CountVectorizer)	0.915	0.849	0.949	0.890
Random forest classifier (CountVectorizer)	0.920	0.894	0.901	0.895
MultinomialNB (TF-IDF)	0.886	0.934	0.761	0.830
Passive aggressive classifier (TF-IDF)	0.925	0.854	0.971	0.900
Logistic regression (TF-IDF)	0.940	0.928	0.915	0.928

<b>Machine learning models</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1 score</b>
Decision tree classifier (TF-IDF)	0.938	0.938	0.898	0.911
Gradient boosting classifier (TF-IDF)	0.916	0.846	0.957	0.899
Random forest classifier (TF-IDF)	0.947	0.940	0.924	0.935

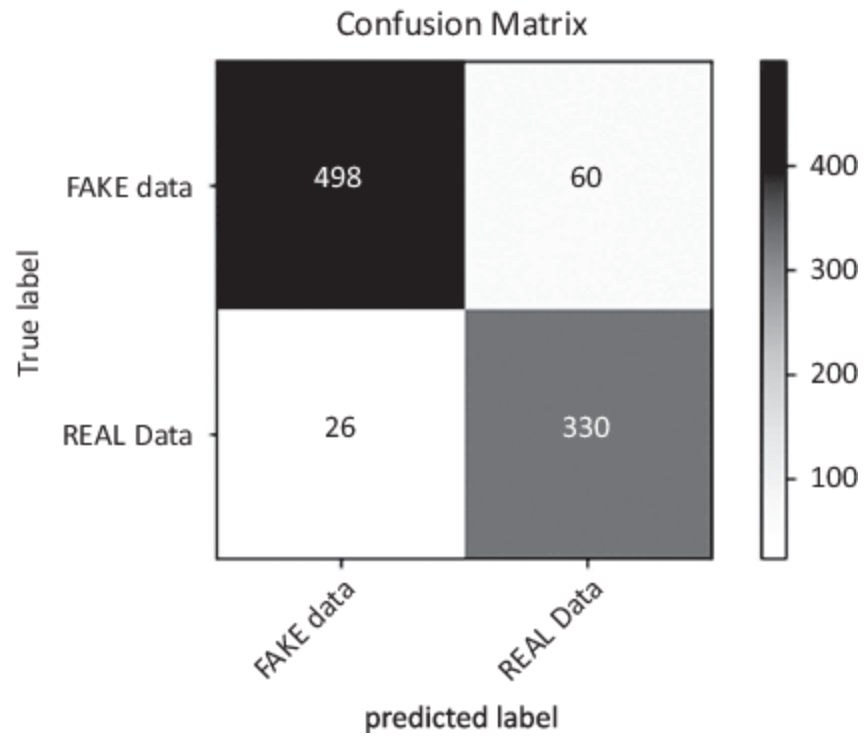
Using the RF classifier with TF-IDF, we achieve the highest accuracy of 94% on the provided training set, whereas the lowest accuracy is 85% which we achieved using multinomial NB and CountVectorizer. Multinomial NB classifier has not only given lowest accuracy with CountVectorizer but also with TF-IDF. So we can conclude that multinomial NB is not a good classifier while working on detecting fake news. Passive

aggressive classifier along with CountVectorizer gave 87% which is second lowest and along with TF-IDF gave 92% accuracy. PAC's accuracy is good when we use TF-IDF. Logistic regression in both the cases delivers good accuracy, that is, 92% with CountVectorizer and 94% with TF-IDF which is the second highest. DT classifier and GB classifier have also given good results with both extracting methods. Both DT and GB gave 91% using CountVectorizer and 93% and 91%, respectively, using TF-IDF.

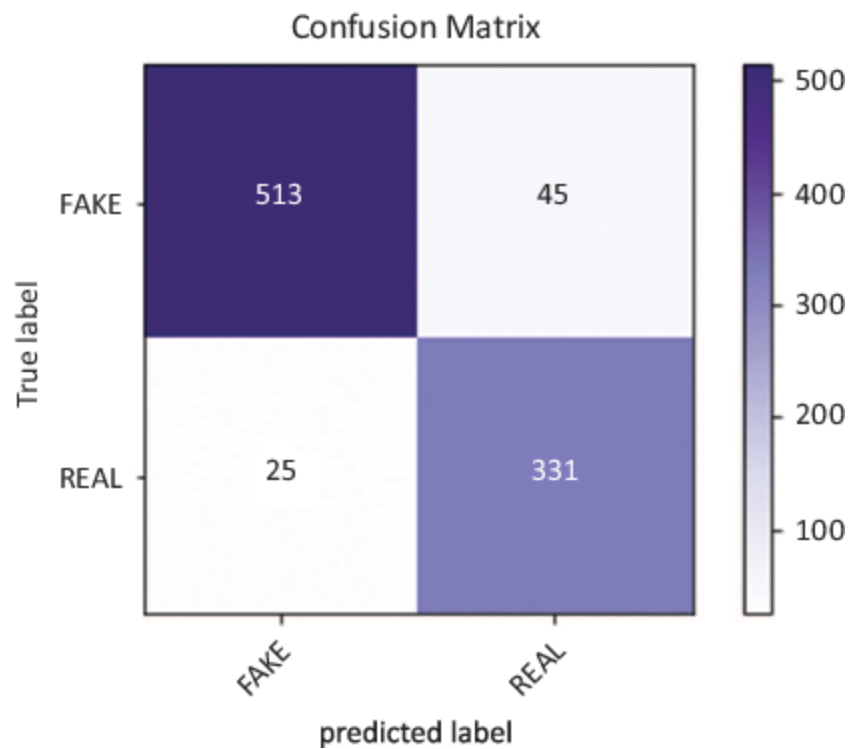
→[Figures 10](#)→[21](#) show confusion matrix for every ML model and for every feature extraction tool we have used.



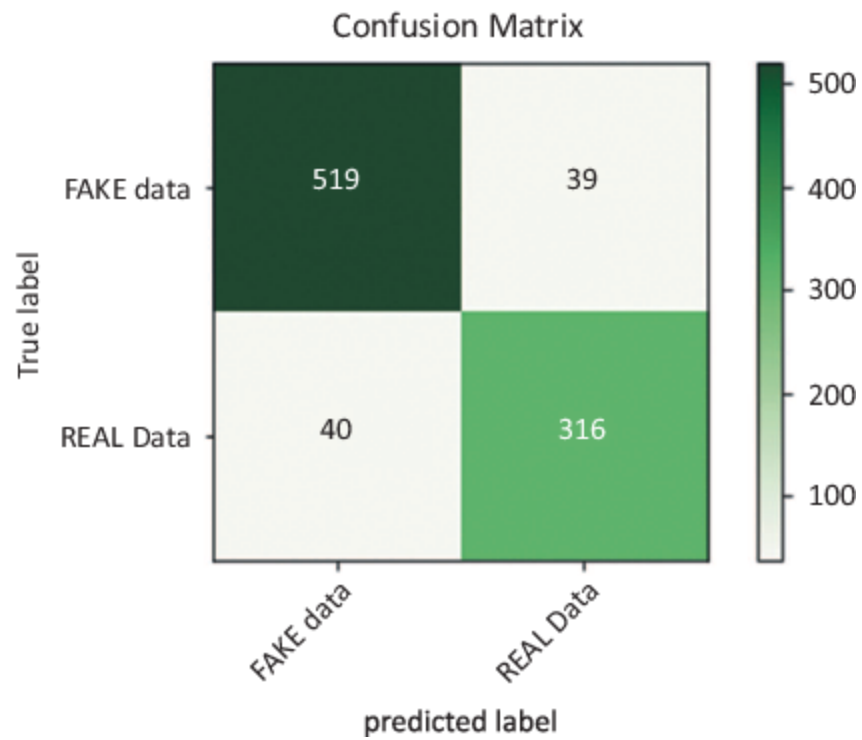
**Figure 10:** Using multinomial NB algorithm and CountVectorizer.



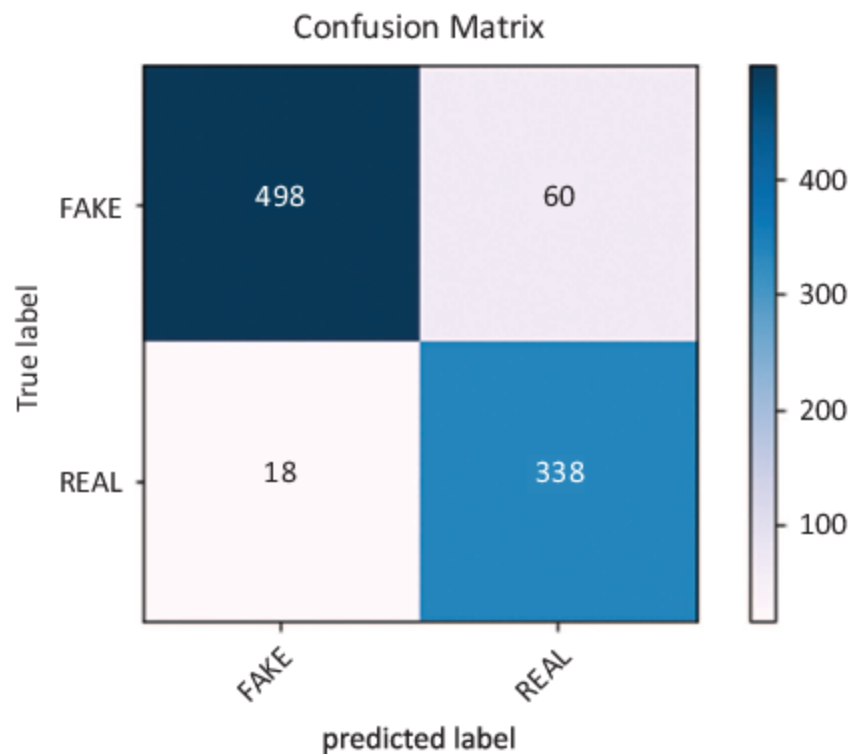
**Figure 11:** Using PAC algorithm and CountVectorizer.



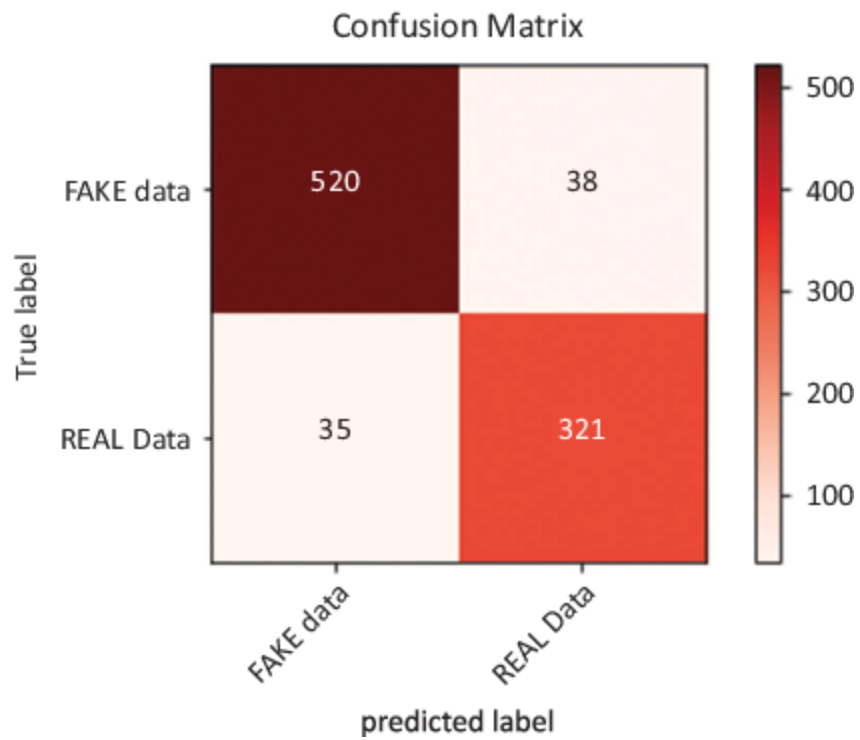
**Figure 12:** Using logistic regression and CountVectorizer.



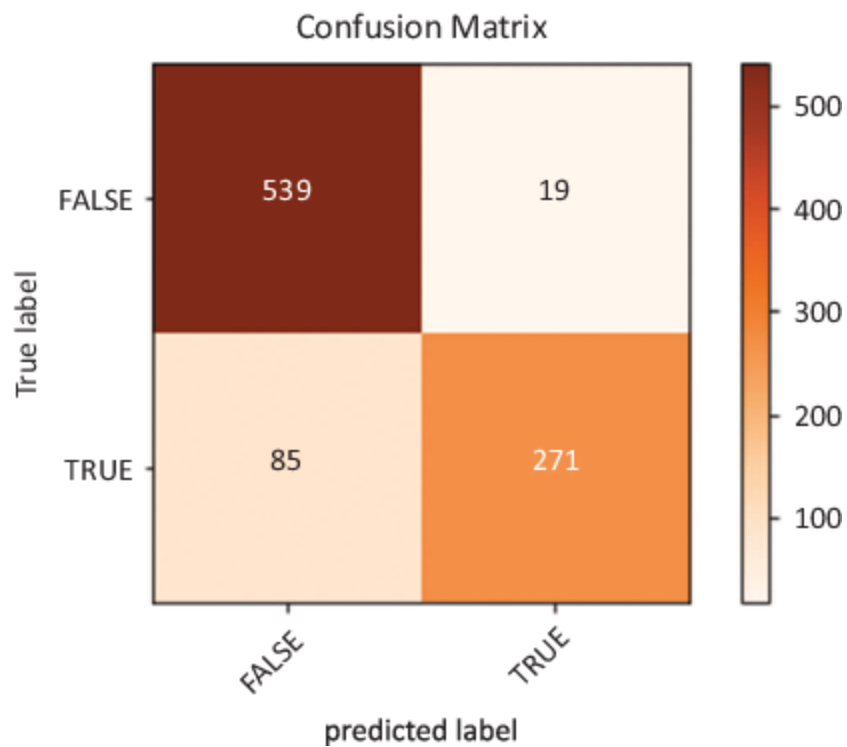
**Figure 13:** Using decision tree classifier and CountVectorizer.



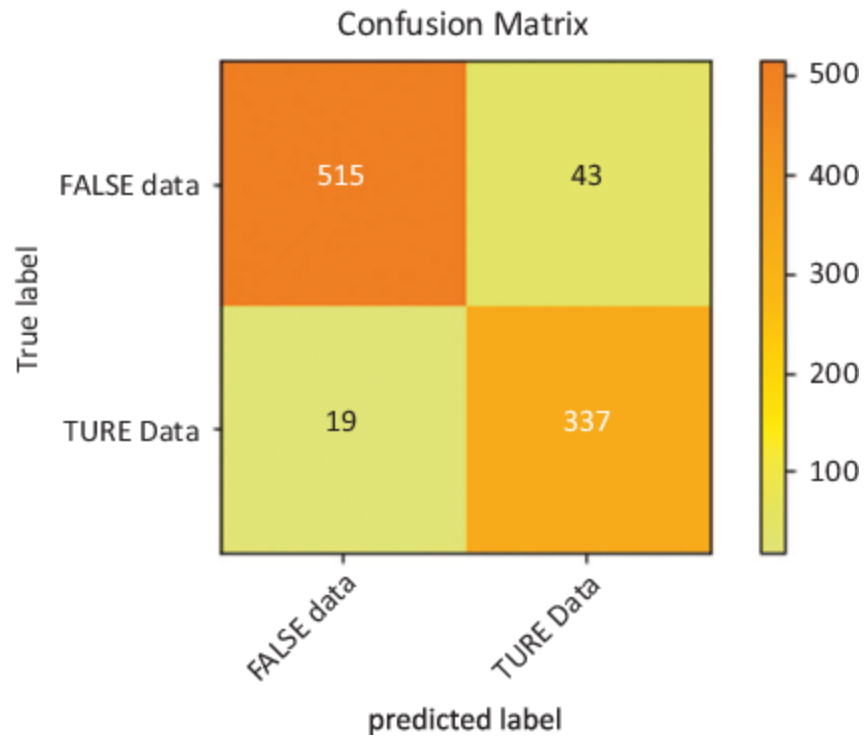
**Figure 14:** Using gradient boosting and CountVectorizer.



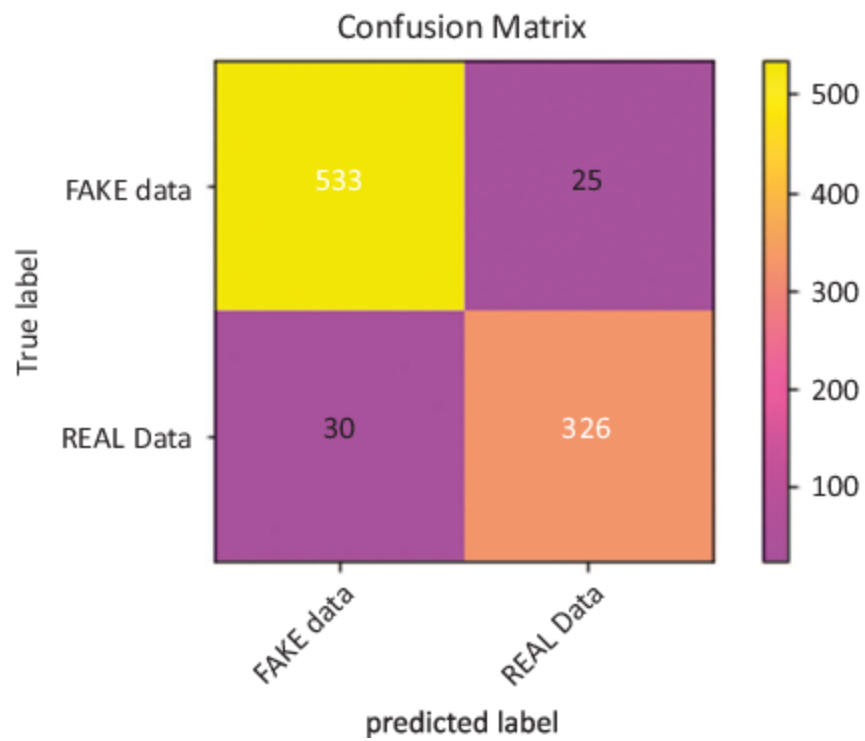
**Figure 15:** Using random forest and CountVectorizer.



**Figure 16:** Using multinomial NB algorithm and TF-IDF.



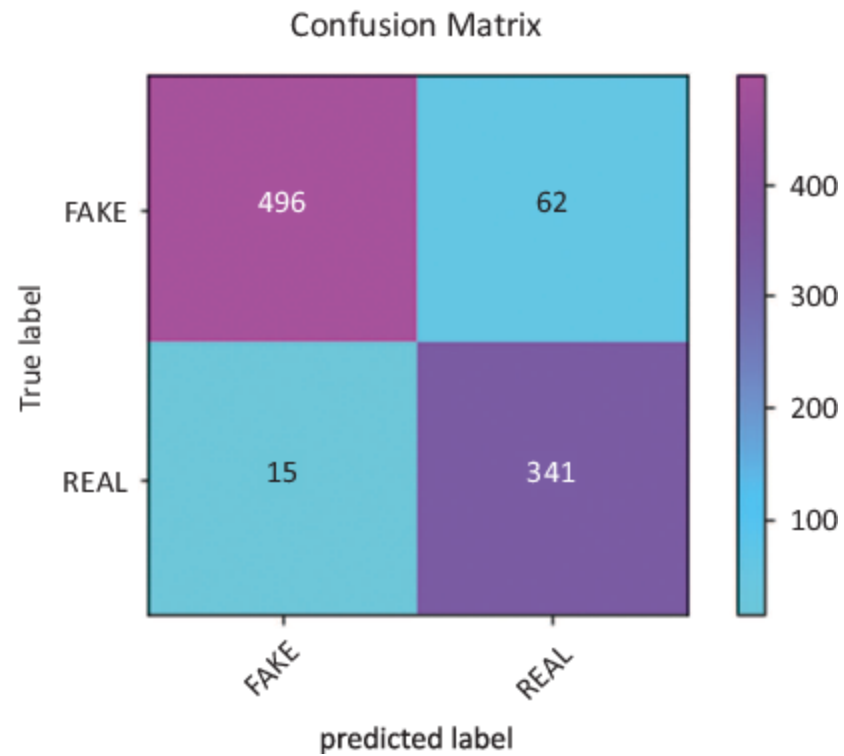
**Figure 17:** Using PAC algorithm and TF-IDF.



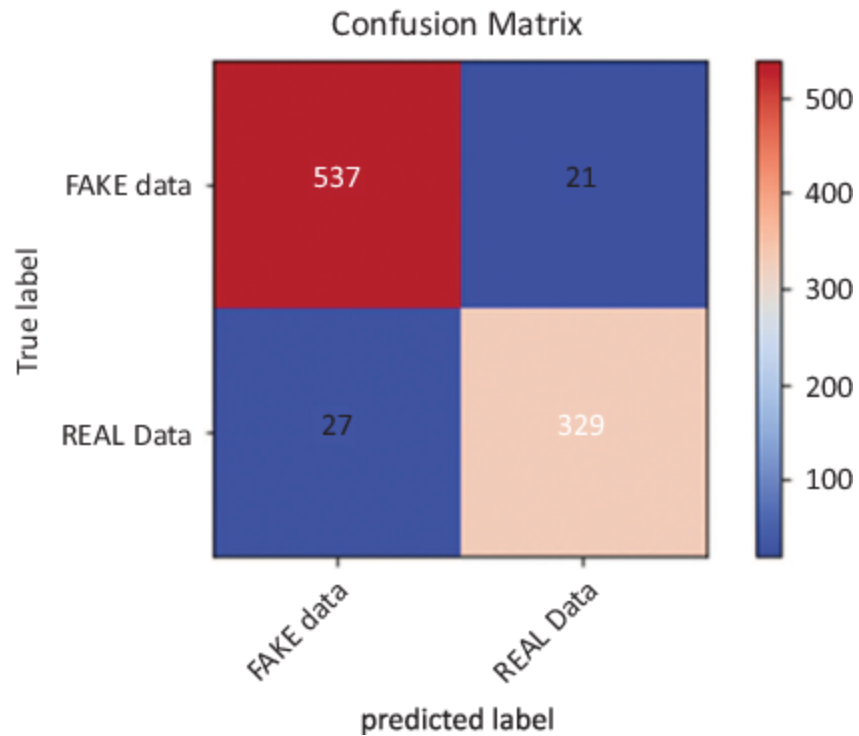
**Figure 18:** Using logistic regression and TF-IDF.



**Figure 19:** Using decision tree classifier and TF-IDF.



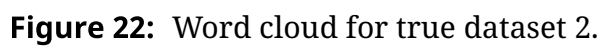
**Figure 20:** Using gradient boosting and TF-IDF.



**Figure 21:** Using random forest classifier and TF-IDF.

## 4.2 Result for dataset 2

→Figures 22 and →23 represent word cloud for true and fake dataset 2.

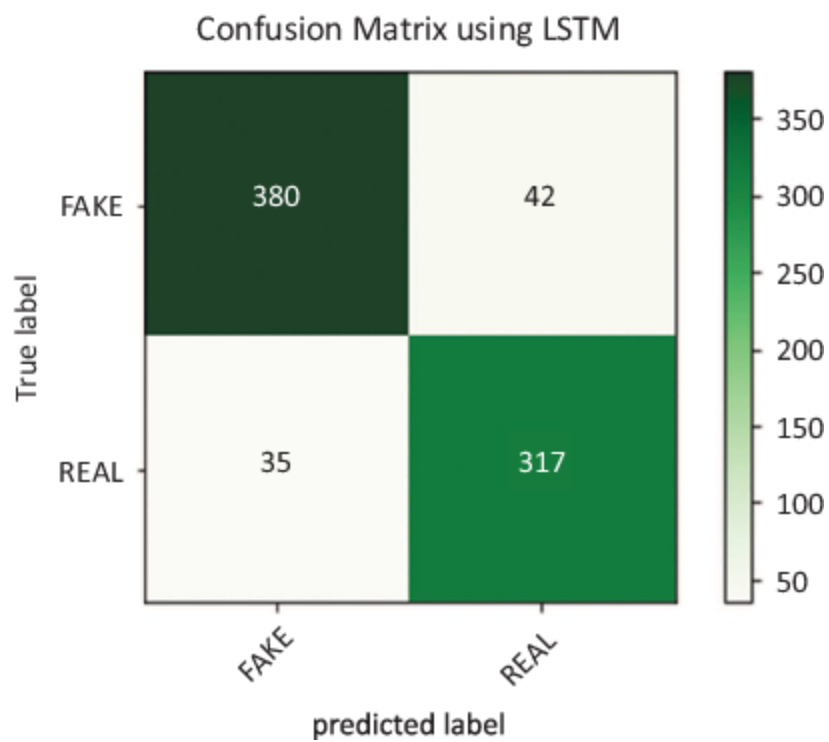




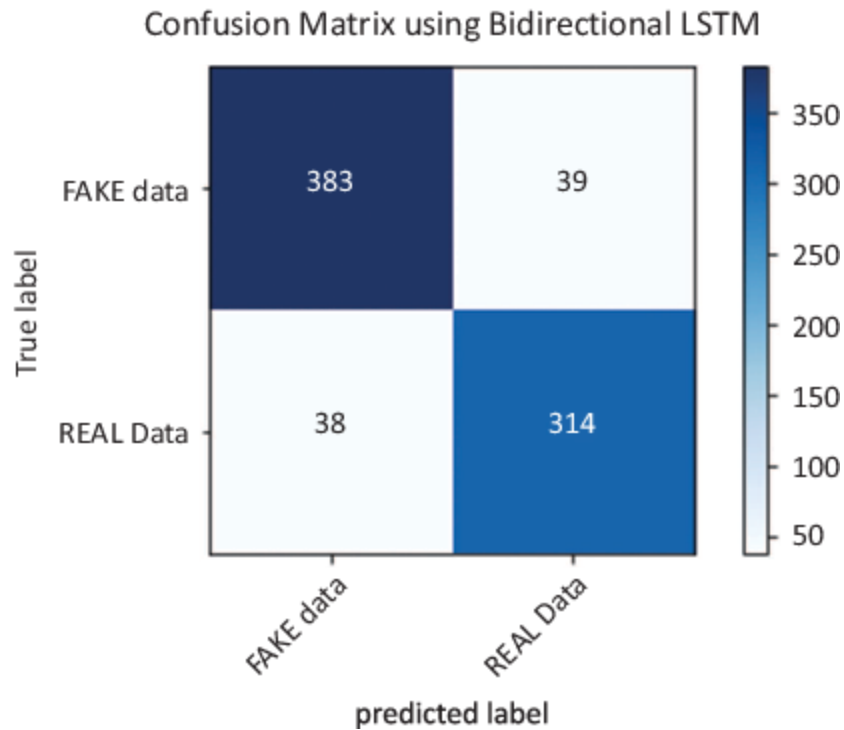
**Table 8:** Results using deep learning models.

Deep learning models	Accuracy	Precision	Recall	<i>F1</i> score
LSTM	0.901	0.883	0.900	0.891
Bidirectional LSTM	0.901	0.889	0.892	0.890

The outputs for dataset 2 are shown in [→Figures 24](#) and [→25](#).



**Figure 24:** Confusion matrix using LSTM.



**Figure 25:** Confusion matrix using BiLSTM.

Out of all the ML models logistic regression and RF with both the feature extraction tools have given good results in classifying fake news, whereas both the DL models have given the same results.

## 5 Conclusion and future work

In recent years, there has been an upsurge in fake news, which have had a negative impact on society. Sharing fake news is one of the most well-known research issues in modern technology, and it stems from a lack of security and confidence in the veracity of the material that is shared on social media. In this

chapter, we discussed how to construct a trust-based architecture for online shared news with NLP, ML, and DL methods. And so, we intended to create a model utilizing all these approaches to determine if a news article or title is false or not and to determine which strategies provide the best results. Six ML models, two LSTM models were given in this research, and two distinct feature extraction techniques were applied on two fresh datasets created by us. To execute false news identification, we have used a variety of variables including the title and content of the item. Out of all six ML models, RF with TF-IDF and logistic regression with CountVectorizer have given the best results whereas both the DL models have given same results. We got more than 85% accuracy for all the rest of the models which is good and satisfactory. Modeling on a new dataset using ML and DL approaches is a little contribution to the research community of fake news classification.

Many of our findings point to the requirement for increasing accuracy. In broad terms, simple algorithms outperform complex ones on smaller (less varied) datasets. Multinomial NB and passive aggressive classifiers underperformed due to the size of our set of data. If we have sufficient time to collect additional misleading data and develop our Python skills, we will try to analyze the information more effectively using  $n$ -

grams and review our DL algorithm. The study involves utilizing our own software; however, the algorithms ran slowly. In the future, we will employ reliable and robust programs that are already accessible to adjust every knob of numerous algorithms. We may expand our dataset for future work and conduct research on photos and videos to help us improve our models.

## References

[1] Fake news, Wikipedia, 2023.

→[https://en.wikipedia.org/wiki/Fake\\_news](https://en.wikipedia.org/wiki/Fake_news). Accessed 18 August 2023. →

[2] G. Krishnamurthy, N. Majumder, S. Poria and E. Cambria, "A Deep Learning Approach for Multimodal Deception Detection," arXiv [cs.CL], 2018. →

[3] V. P. Miletskiy, D. N. Cherezov and E. V. Strogetskaia, "Transformations of Professional Political Communications in the Digital Society (By the Example of the Fake News Communication Strategy)," Communication Strategies in Digital Society Workshop (ComSDS), pp. 121–124, 2019. →

**[4]** “What Is Fake News? Definition, Types, and How to Detect Them,” IONOS Digital Guide, 2020.

→<https://www.ionos.com/digitalguide/online-marketing/social-media/what-is-fake-news/>. Accessed 18 August 2023. →

**[5]** N. R. De Oliveira, D. S. Medeiros and D. M. Mattos, “A Sensitive Stylistic Approach to Identify Fake News on Social Networking,” IEEE Signal Processing Letters, vol. 27, pp. 1250–1254, 2020. →

**[6]** G. Liu, Y. Wang and M. Orgun, “Optimal Social Trust Path Selection in Complex Social Networks,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 24, no. 1, pp. 1391–1398, 2010. →

**[7]** M. Mahyoob, J. Al-Garaady and M. Alrahaili, “Linguistic-based Detection of Fake News in Social Media,” International Journal of English Linguistics, vol. 11, no. 1, 2020. →

**[8]** A. Koirala, “COVID-19 Fake News Classification with Deep Learning,” Preprint, pp. 1–6, 2020. →

**[9]** H. Gill and H. Rojas, “Chatting in a Mobile Chamber: Effects of Instant Messenger Use on Tolerance toward Political Misinformation among South Koreans,” Asian Journal of Communication, vol. 30, no. 6, pp. 470–493, 2020. →

**[10]** J. L. Alves, L. Weitzel, P. Quaresma, C. E. Cardoso and L. Cunha, "Brazilian Presidential Elections in the Era of Misinformation: A Machine Learning Approach to Analyse Fake News," in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 24th Ibero American Congress, CIARP 2019, Havana, Cuba, Proceedings, Springer International Publishing, pp. 72–84, 2019. ➡

**[11]** D. Mouratidis, M. N. Nikiforos and K. L. Kermanidis, "Deep Learning for Fake News Detection in a Pairwise Textual Input Schema," Computation, vol. 9, no. 2, pp. 1–15, 2021. ➡

**[12]** S. Paul, J. I. Joy, S. Sarker, S. Ahmed and A. K. Das, "Fake News Detection in Social Media Using Blockchain," in 7th International Conference on Smart Computing & Communications, pp. 1–5, 2019. ➡

**[13]** S. Ahmed, K. Hinkelmann and F. Corradini, "Development of Fake News Model Using Machine Learning through Natural Language Processing," International Journal of Computer and Information Engineering, vol. 14, no. 12, pp. 454–460, 2020. ➡

**[14]** A. Karbowski, "A Note on Patents and Leniency Gospodarka Narodowa," The Polish Journal of Economics, vol. 301, no. 1, pp. 97–108, 2020. ➡

**[15]** J. Antony Vijay, H. Anwar Basha and J. Arun Nehru, "A Dynamic Approach for Detecting the Fake News Using Random Forest Classifier and NLP," in Computational Methods and Data Engineering: Proceedings of ICMDE 2020, Singapore: Springer, vol. 2, pp. 331–341, 2020. ➡

**[16]** A. Chokshi and R. Mathew, "Deep Learning and Natural Language Processing for Fake News Detection: A Survey," International Conference on IoT based Control Networks and Intelligent Systems (ICICNIS 2020), pp. 716–728, 2020. ➡

**[17]** Y. Wang, W. Yang, F. Ma, J. Xu, B. Zhong, Q. Deng and J. Gao, "Weak Supervision for Fake News Detection via Reinforcement Learning," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 1, pp. 516–523, 2020. ➡

**[18]** S. Gilda, "Evaluating Machine Learning Algorithms for Fake News Detection/subscribe," Neural Computation Archive, 2017. ➡

**[19]** Y. Qin, W. Dominik and C. Tang, "Predicting Future Rumours," Chinese Journal of Electronics, vol. 27, no. 3, pp. 514–520, 2018. ➡

**[20]** S. Aphiwongsophon and P. Chongstitvatana, "Detecting Fake News with Machine Learning Method," in 2018 15th

international conference on electrical engineering/electronics, computer, telecommunications and information technology, pp. 528–531, 2018. ➡

**[21]** M. Granik and V. Mesyura, “Fake News Detection Using Naive Bayes Classifier,” in 2017 IEEE first Ukraine conference on electrical and computer engineering (UKRCON), pp. 900–903, 2017. ➡

**[22]** A. Jain and A. Kasbe. “Fake News Detection,” in 2018 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS), pp. 1–5, 2018. ➡

**[23]** A. Gupta and R. Kaushal, “Improving Spam Detection in Online Social Networks,” in 2015 International conference on cognitive computing and information processing (CCIP), pp. 1–6, 2016. ➡

**[24]** N. K. Conroy, V. L. Rubin and Y. Chen, “Automatic Deception Detection: Methods for Finding Fake News,” Proceedings of the Association for Information Science and Technology, vol. 52, no. 1, pp. 1–4, 2015. ➡

**[25]** R. Barua, R. Maity, D. Minj, T. Barua and A. K. Layek, “F-NAD: An Application for Fake News Article Detection Using Machine

Learning Techniques,” in 2019 IEEE Bombay section signature conference (IBSSC), pp. 1–6, 2019. ➡

**[26]** B. Bhutani, N. Rastogi, P. Sehgal and A. Purwar, “Fake News Detection Using Sentiment Analysis,” in 12th International conference on contemporary computing, 2019. ➡

**[27]** M. Vohra and M. Kakkar, “Detection of Rumor in Social Media,” in 8th International conference on cloud computing, data science & engineering, pp. 485–490, 2018. ➡

**[28]** R. A. Monteiro, R. L. Santos, T. A. Pardo, T. A. De Almeida, E. E. Ruiz and O. A. Vale, “Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results,” in Computational Processing of the Portuguese Language: 13th International Conference, Canela, Brazil, Proceedings Springer International Publishing, pp. 324–334, 2018. ➡

**[29]** S. B. Parikh and P. K. Atrey, “Media-rich Fake News Detection: A Survey,” in 2018 IEEE conference on multimedia information processing and retrieval, pp. 436–441, 2018. ➡

**[30]** V. Pérez-Rosas, B. Kleinberg, A. Lefevre and R. Mihalcea, “Automatic Detection of Fake News,” arXiv preprint arXiv:1708.07104, 2017. ➡

[31] Europa.eu. →[https://edps.europa.eu/press-publications/publications/techsonar/fake-news-detection\\_en](https://edps.europa.eu/press-publications/publications/techsonar/fake-news-detection_en).

Accessed 18 August 2023. →

[32] Ai – Natural Language Processing Tutorialspoint.

→[https://www.tutorialspoint.com/artificial\\_intelligence/artificial\\_intelligence\\_natural\\_language\\_processing](https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_natural_language_processing). Accessed 18 August

2023. →

[33] “Machine Learning: What It is, Tutorial, Definition, Types – javatpoint,” →[www.javatpoint.com](http://www.javatpoint.com).

→<https://www.javatpoint.com/machine-learning>. Accessed 18 August 2023. →

[34] “News Headlines, English News, Today Headlines, Top Stories,” Hindustan Times. →<https://www.hindustantimes.com/>.

Accessed 18 August 2023. [a](#), [b](#)

[35] “Kaggle: Your Machine Learning and Data Science

Community,” Kaggle.com. →<https://www.kaggle.com/>. Accessed

18 August 2023. [a](#), [b](#)

[36] V. Kumar, A. Kumar, A. K. Singh and A. Pachauri, “Fake News Detection Using Machine Learning and Natural Language Processing,” in 2021 International Conference on Technological Advancements and Innovations, pp. 547–552, 2021. →

**[37]** S. Naik and A. Patil, "Fake News Detection Using NLP," International Journal for Research in Applied Science & Engineering Technology, vol. 9, no. 12, pp. 1-9, 2021. ➡

**[38]** S. Sarkar and M. Nandan, "A Comprehensive Approach to AI-Based Fake News Prediction in Digital Platforms by Applying Supervised Machine Learning Techniques," Handbook of Research on Applications of AI, Digital Twin, and Internet of Things for Sustainable Development, pp. 61-86, 2023. ➡

**[39]** Z. Shahbazi and Y. C. Byun, "Fake Media Detection Based on Natural Language Processing and Blockchain Approaches," IEEE Access, vol. 9, pp. 128442-128453, 2021. ➡

**[40]** "CountVectorizer in Python," *Educative: Interactive Courses for Software Developers*.  
➡<https://www.educative.io/answers/countvectorizer-in-python>.  
Accessed 18 August 2023. ➡

**[41]** S. Swaminathan, "Logistic Regression – Detailed Overview," Towards Data Science, 2018.  
➡<https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>. Accessed 18 August 2023. ➡

**[42]** Sriram, "Multinomial naive Bayes explained: Function, advantages & disadvantages, applications in 2023," 2022. ➡

[43] A. Kumar, "Passive Aggressive Classifier: Concepts & Examples," *Data Analytics*, 2022. →<https://vitalflux.com/passive-aggressive-classifier-concepts-examples/>. Accessed 18 August 2023. →

[44] *Javatpoint.com*. →<https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>. Accessed 18 August 2023. →

[45] *Geeksforgeeks.org*. →<https://www.geeksforgeeks.org/ml-gradient-boosting/>. Accessed 18 August 2023. →

[46] *Simplilearn.com*.  
→<https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm>. Accessed 18 August 2023. →

[47] A. Follow, "Long Short Term Memory Networks Explanation," *GeeksforGeeks*, 2019.  
→<https://www.geeksforgeeks.org/long-short-term-memory-networks-explanation/>. Accessed 18 August 2023. →

[48] "Papers with Code – BiLSTM Explained,"  
*Paperswithcode.com*.  
→<https://paperswithcode.com/method/bilstm>. Accessed 18 August 2023. →

---

## NOTES

---

**1**      **Conflicts of interest:** The authors confirm that there is no conflict of interest to declare for this publication.

---

# Spam mail detection various machine learning methods and their comparisons

**Harendra Singh Negi**

**Aditya Bhatt**

**Vandana Rawat**

## **Abstract**

The application of machine learning techniques for spam mail detection is explored in this chapter. Utilizing both the CountVectorizer and TF-IDF vectorizer techniques, five algorithms were created: naive Bayes, decision tree, random forest (RF), support vector machine (SVM), and XGBoost. Performance metrics such as AUC-ROC, precision-recall curve, *F1* score, recall, accuracy, and precision were utilized to evaluate each method. With an accuracy of 96.67%, RF outperformed the other algorithms while using CountVectorizer. SVM and RF were found to be the top-performing algorithms by using TF-IDF vectorizer, with accuracy ratings of 97.33% and 96.50%, respectively. The results imply that these algorithms are effective in detecting spam mail. The study's conclusions on the effectiveness of several

machine learning algorithms for spam mail identification might be useful to researchers in the domains of natural language processing and machine learning.

**Keywords:** Spam mail detection, machine learning algorithms, naive Bayes, support vector machines, random forest, decision tree, extreme gradient boosting, processing techniques, CountVectorizer, TF-IDF vectorizer, natural language processing,

## 1 Introduction

Spam mail, or unwanted and unsolicited email, is a pervasive problem that affects millions of users worldwide [[→1](#)]. Spam can not only waste time and resources but also carry malicious content that can compromise security and privacy. As such, spam detection has become a crucial task for email service providers and users alike [[→2](#)]. In the current digital environment, spam email presents considerable hurdles since fraudsters are continuously coming up with new ways to get past established filters. To reduce spam's negative effects on people, businesses, and society as a whole, it is essential to comprehend its features and create effective detection tools. This research evaluates machine learning algorithms and preprocessing methods to enhance spam detection strategies. The knowledge gathered may be used to create filters that are more precise and effective, giving users the ability to secure

their email exchanges and preserve important data. To form a unified front against spam and provide a safer email environment for all users, interdisciplinary partnerships between machine learning scientists, cybersecurity experts, and email service providers are crucial. Machine learning algorithms are widely used in spam detection because they can learn patterns and features of large numbers of email data [→3]. The effectiveness of these algorithms might, however, be influenced by a variety of factors, such as the pre-processing technique utilized. Here, we evaluate the potency of [→4, →5, →6] five well-known machine learning techniques – naive Bayes, SVM, decision tree, random forest (RF), and KNN – that are used to identify spam mail. We will evaluate these algorithms on two different preprocessing techniques: CountVectorizer and TF-IDF vectorizer. The two preprocessing techniques that are used here are very commonly used to convert raw text into a numerical representation; that is, natural language processing (NLP) is used to assist in the training of machine learning models. In order to determine the frequency of each word in a text, we utilized Countvectorizer and TF-IDF vectorizer. We measured the algorithms' accuracy, precision, recall, *F1* score, and AUC-ROC using a dataset of spam and nonspam emails to assess their performance. We also

evaluated the computational effectiveness of the algorithms and looked at how preprocessing methods affected them.

The overall goal of this chapter is to aid email users by improving the precision and effectiveness of spam filters by offering insights into the efficacy of various machine learning algorithms and preprocessing approaches for spam mail identification. This information will be valuable for anyone working in the domains of machine learning, NLP, and cyber security including researchers, professionals, and students.

## **2 Related works**

Numerous academics have used different ML algorithms to classify mail as spam. Additionally, the performance of the process of recognizing spam mail has been exploited, as have a number of preprocessing approaches. Here, a thorough analysis of the relevant literature on spam mail detection is provided, including various feature extraction, preprocessing, and machine learning techniques. We will also go through the advantages and disadvantages of each strategy and point out the research hole that our study attempts to fill.

Vijay Srinivas Tida creates a unique spam detection approach using a pretrained BERT-based uncased model from Google. The

authors trained each model separately using one of four datasets. Four datasets were used to build USDM, which made use of the hyperparameters from each model. The study's findings show a 97% overall accuracy [[→7](#)]. The work being presented attempts to classify spam mail using four different sorting algorithms utilizing a dataset of 4,601 incidents. However, naive Bayes works best when performance is assessed using execution time or time measurements [[→8](#)]. The paper provides a comprehensive analysis of how to classify emails using unsupervised clustering techniques into the spam and ham categories.

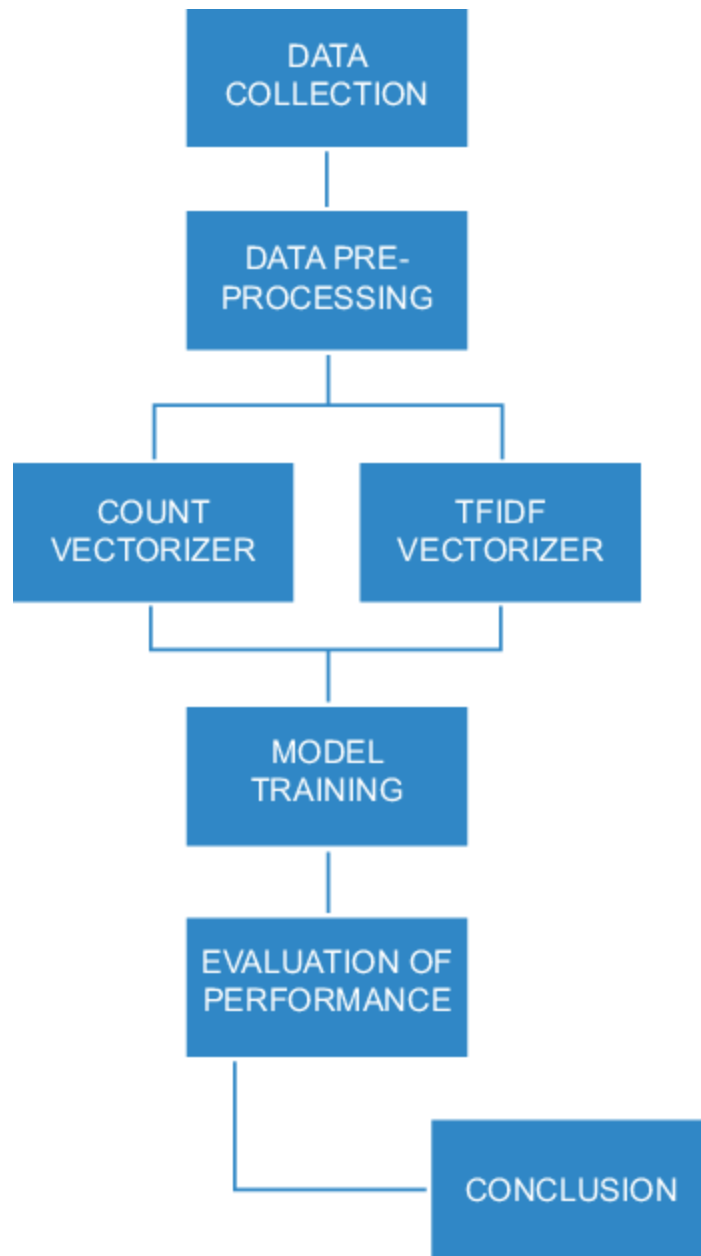
To bridge this gap, the authors provide an antispam architecture that makes use of multialgorithm clustering and completely unsupervised methods. Another research examined email header-related data, and it offered a special feature reduction methodology that makes use of a variety of unsupervised feature extraction techniques. They used a 100,000 spam and junk email collections as their data source. The results demonstrate that the k-means clustering algorithms performed as expected, whereas OPTICS accurately predicted the best grouping among the top three methods, with an accuracy of 94.91% and a high level of confidence in the framework for feature reduction that was proposed [[→9](#)]. In order to enhance spam email detection systems, Al-Rawashdeh

et al. [[→10](#)] present a feature selection approach that uses meta-heuristic optimization methods to choose the best option. The results showed that SVM performed better than other classifiers with an *F1* score of 96.3%, while layered pollination outperformed other feature selection strategies [[→10](#)]. With the use of the email's text and header, Karim et al. [[→11](#)] develop a solution that is unsupervised for separating spam from legitimate emails. The study made use of a fresh binary dataset containing 22,000 trashes with legit emails and 10 criteria. With a cumulative accuracy of 75.76%, the authors examined five alternative clustering methods and found that OPTICS provided the most efficient clustering. The study is presented in a creative way that uses unsupervised machine learning techniques, and overall, the writing is excellent [[→11](#)]. To detect spam emails, Gibson et al. [[→12](#)] improved machine learning approaches using bio-inspired metaheuristic methodologies. Using evolutionary algorithms and the optimization of particle swarms, the classifier's performance was improved. All other approaches were surpassed by multinomial naive Bayes and genetic algorithm, which achieved 100 accuracies by optimizing GA on randomized data distribution. To decide which model was most suited, the researchers compared their findings with those from different machine learning and bio-inspired models [[→12](#)]. The MIS is a brand-new spam email detection model that

examines email headers, bodies, characters, and words using an enhanced recurrent convolutional neural network model with multilevel vectors and attention mechanism. It earned a 99.848% overall accuracy rate and a 0.043% false-positive rate (FPR). In order to combat the growing problem of spam emails, the report emphasizes the need of more efficient phishing detection systems [→13]. Using GRU-RNN and SVM, Alauthman [→14] suggests a novel method for identifying botnet spam emails that achieves 98.7% accuracy on the Spambase dataset. A unique spam detection technique was created by Venkatraman et al. [→15] which does a Bayesian classification using conceptual-semantic similarity. The suggested technique outperformed current strategies in tests on benchmark data sets, achieving a precision of 98.89% [→15]. On 800 Turkish email datasets, the SMO and MLP algorithms performed best, with *F*-measure values of 0.985 and 0.984, respectively [→16]. Hnini et al. [→17] put out a technique for extracting feature vectors for email's text and image content using PV-DBOW and CNN. They integrated the feature vectors and entered them into an RF model, which produced a 99.16% improvement in accuracy over the most recent cutting-edge techniques [→17].

### **3 Purposed work**

In order to identify spam emails from a preprocessed dataset, this chapter offers to investigate the efficacy of five machine learning algorithms. [→Figure 1](#) visualizes the workflow of proposed work. Here, over 30,000 emails from the Enron Corporation, a now-defunct energy corporation in the United States that was embroiled in a significant accounting controversy in the early 2000s, are included in the dataset used in this research, which was obtained through Kaggle [[→18](#)]. The dataset consists of thousands of emails covering a wide variety of topics and communication styles including sales pitches, phishing scams, and fraudulent schemes. It also includes legitimate emails sent and received by Enron employees in order to compare spam and nonspam messages. The author has organized her mail into six distinct folders, each having ham and spam folders. From there, we organized the emails into two folders: ham and spam. The emails were preprocessed by eliminating punctuation, stop-words, headers, footers, and HTML elements before to utilizing the dataset for training and testing. For training and testing, 3,000 emails were randomly chosen from the preprocessed dataset. The preprocessed emails were turned into numerical feature vectors using the NLP [[→19](#)] methods CountVectorizer and TF-IDF vectorizer. From the features obtained we trained the model and evaluated their performance.



**Figure 1:** Workflow of purposed work.

**Table 1:** Email dataset.

Emails	Spam
receivables 2 pm scheduled cibctuesday discus ...	0
p 9 uayfoiwrppornanslixrmrgited time pro 3 ...	1
med girl happy girl unsatisfied potency wait f ...	1
revised revisedeastrans nomination change eff ...	0
13 1891 collect call alyssamilano 1824 small ...	1

→[Table 1](#) displays a tiny subset of an email sample consisting of five emails. Each email is tagged as either spam (1) or nonspam (0). The emails vary in substance, with some including nonsense or meaningless language and others having a more logical structure. Using this dataset, models for machine learning may be trained, evaluated, and improved.

CountVectorizer and TF-IDF vectorizer are two prominent algorithms for text feature extraction in NLP. Text documents are transformed into token counts by CountVectorizer and TF-IDF features by TF-IDF vectorizer. In the present project, both

CountVectorizer [[→20](#)] and TF-IDF vectorizer [[→21](#)] were built individually on the preprocessed Enron spam dataset. Raw text data was transformed using the CountVectorizer method into a token count matrix, where each row corresponds to each email and each column to a term in the corpus. Text data was transformed using the TF-IDF vectorizer technique into a matrix of TF-IDF features, where each row corresponds to each email and each column to a vocabulary token. The two methods were put into practice with scikit-learn, a Python machine learning toolkit. The vectorizer was created, fitted to the data used for training utilizing the `sklearnfit_transform()` function, and then converted to the test data using the `sklearn transform()` method to create the feature matrices required for training as well as testing models. On a prepared dataset, the study that is suggested compares the effectiveness of five machine learning methods: accuracy [[→22](#)], precision [[→23](#)], recall [[→24](#)], and [[→25](#)]. *F1* score was used as performance parameter. Based on time elapsed [[→26](#)], we also will assess the effectiveness of two NLP methodologies. And at conclusion, comparing the performance based on AUC-ROC scores [[→27](#)] and precision-recall curve (PR curve) [[→28](#)] of machine learning models and in both case of NLP approaches, all these words will be examined further in the chapter. For purposes of training and testing, the dataset was randomly split into two

halves of 80:20. Study will also undertake cross-validation to verify that the findings are consistent and credible. The SVM [→29] machine learning approach is a common option for categorization assignments. SVM separates the data points into as many classes as feasible by selecting the optimum hyperplane. More precisely, we employed sklearn's support vector classification, which by default employs the radial basis function kernel and enables nonlinear data separation. The decision tree approach separates the data into smaller groups based on numerous criteria in a recursive fashion. The decision tree [→30] approach can handle category and numerical data and is straightforward to grasp. RF is another well-liked approach. To provide a final prediction, RF [→31] employs the combined results of several decision trees. This algorithm is recognized for being dependable and exact. We will also study the probabilistic approach known as naive Bayes [→32] in this recommended endeavor. This strategy is computationally efficient and suited for high-dimensional data as it functions under the notion that the features are independent of one another. Last but not least, "Extreme Gradient Boosting" or XGBoost [→33], it is a prominent machine learning approach for addressing regression and classification issues. It is based on the gradient boosting technique, but it incorporates various modifications and optimizations that make it quicker and more

accurate. These algorithms were selected for their ability to handle high-dimensional text input and their success in binary classification tasks. Additionally, they are well-established and commonly used algorithms in the machine learning field, making them a perfect candidate for spam mail identification. In order to improve spam detection systems, the study compares the effectiveness of five machine learning algorithms when identifying spam emails from a preprocessed dataset.

## 4 Implementation and result

### 4.1 Performance metrics

Performance metrics are measures that analyze predictive quality of model. When doing classification tasks, a variety of standard performance indicators are used to judge how well a model predicts. Accuracy, precision, recall, and *F1* score are some of these metrics. For the classification of illnesses affecting wheat and rice, we studied the performance of five machine learning algorithms in this research. Building confusion matrix and obtaining the true positives (TP) and true negatives (TN), positive- and negative-false results will be done for doing the performance matrix calculation. A confusion matrix [→34] is used to analyze how well a machine learning model performs in classification tasks. It shows the proportion

of accurate and inaccurate predictions made by the model for every category by comparing the labels that were predicted with the actual labels. The number of TP, TN, false positives (FP), and false negatives (FN) in a confusion matrix is often represented in the four cells. TP is an instance in which the model correctly identifies as positive, TN is an instance in which the model incorrectly identifies as negative, FP is an instance in which the model incorrectly identifies as positive despite being negative in reality, and FN is an instance in which the model incorrectly identifies as negative despite being positive in reality. Using a confusion matrix, it is possible to evaluate the effectiveness of a classification model by measuring its accuracy, precision, recall,  $F1$  score, among other performance indicators. These measures are created using the confusion matrix's values. These are represented in [→Table 2](#).

**Table 2:** Confusion matrix using CountVectorizer.

Algorithms	TP	TN	FP	FN
Naive Bayes	275	279	14	32
Decision tree	290	262	31	17
Random forest	300	280	13	7
SVM	307	227	66	0
XGBoost	300	274	19	7

**Table 3:** Confusion matrix using TF-IDF vectorizer.

Algorithms	TP	TN	FP	FN
Naive Bayes	290	284	6	17
Decision tree	293	264	24	14
Random forest	301	278	15	6
SVM	304	280	13	3
XGBoost	302	272	21	5

The machine learning models' confusion matrices are shown in [→Table 2](#) for training using CountVectorizer. The machine learning models' confusion matrices are given in [→Table 3](#) for training using TF-IDF vectorizer. To determine the performance matrix for associated models trained using appropriate NLP techniques, the confusion matrices were then employed. The degree to which a model is accurate determines how effectively it can classify events. This can be defined as the proportion between all forecasts and accurate predictions. A model can predict the class labels for the majority of situations with high

accuracy. Accuracy can be represented in  $(TP + TN)/(TP + TN + FP + FN)$  manner. Precision measures how well a model can spot good things happening. The ratio of genuine positives, or instances that are correctly categorized as positive, to all instances of projected positive occurrences, is known as the true positive rate (TPR). An accurate model may be able to predict the majority of positive events with reliability. Precision is equal to  $TP/(TP + FP)$ . Recall is a metric that gauges how well a model can identify successful cases. This can be defined as the proportion of actual positive events to all positive events. Most of the positive examples may be accurately identified by a model with high memory.  $Recall = TP/(TP + FN)$ . By applying the harmonic mean of recall and accuracy,  $F1$  may be determined.  $F1 \text{ score} [F1 \text{ score} = 2 * (precision * recall)/(precision + recall)]$ , the  $F1$  score provides a balanced evaluation of a model's performance, taking into account both precision and recall. Table 4 shows each of these performance matrices:

**Table 4:** Performance matrix using CountVectorizer.

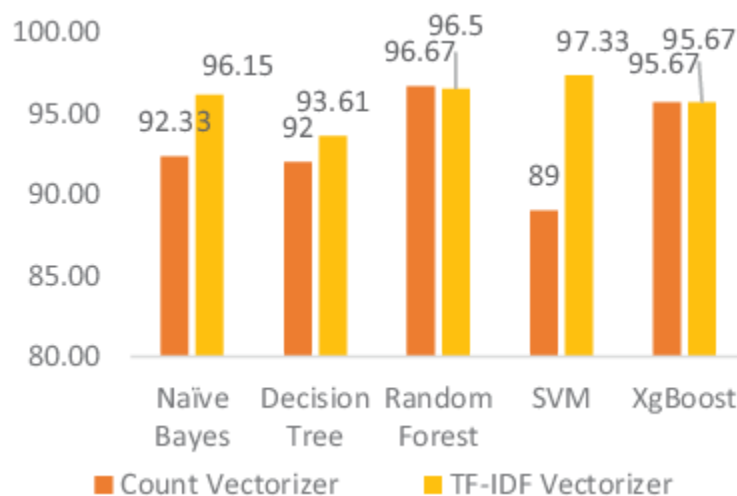
Algorithms	Accuracy	Precision	Recall	<i>F1</i>
Naive Bayes	92.33	95.16	89.58	92.28
Decision tree	92.00	90.34	94.46	92.36
Random forest	96.67	95.85	97.72	96.77
SVM	89.00	82.31	100	90.29
XGBoost	95.67	94.04	97.72	95.85

**Table 5:** Performance matrix using TF-IDF vectorizer.

Algorithms	Accuracy	Precision	Recall	<i>F1</i>
Naive Bayes	96.15	97.97	94.46	96.19
Decision tree	93.61	92.43	95.44	93.91
Random forest	96.50	95.25	98.05	96.63
SVM	97.33	95.90	99.02	97.44
XGBoost	95.67	93.50	98.37	95.87

From [→Table 4](#) we can observe the performance of different algorithms using the CountVectorizer, naive Bayes achieved an accuracy of 92.33%, XGBoost with 95.67%, decision tree with 92.00%, RF with 96.67%, and SVM with 89.00%. From [→Table 5](#), we can observe performance of algorithms using the TF-IDF vectorizer, SVM performed the best with an accuracy of 97.33%, followed by RF with 96.50%, naive Bayes with 96.15%, XGBoost with 95.67%, and decision tree with 93.61%.

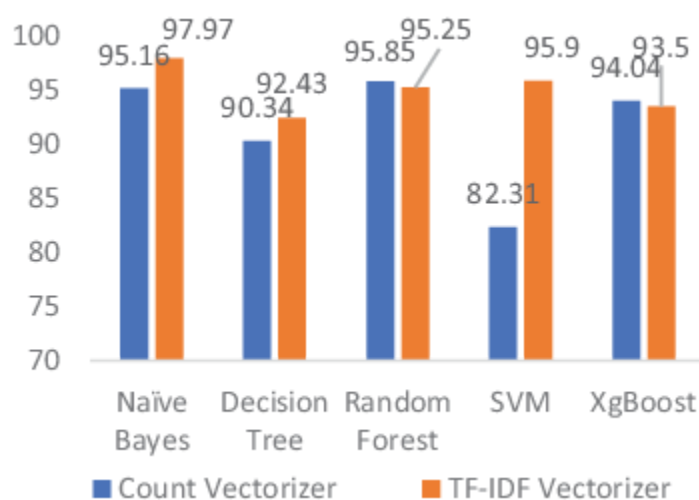
As shown in [→Figure 2](#), the RF method consistently outperformed both vectorizers, obtaining high accuracy values of 96.5% and 96.6%. SVM also performed well using the TF-IDF vectorizer, but its accuracy was lower when using CountVectorizer. Naive Bayes and XGBoost achieved relatively high accuracy using both vectorizers. We can also observe that TF-IDF vectorizer is producing excellent results for all models consistently, whereas CountVectorizer performed poor in case of SVM.



**Figure 2:** Accuracy comparison for vectorizers with models.

[→Figure 3](#) suggests that RF has the highest precision in both CountVectorizer and TF-IDF vectorizer with 95.85% and 95.25%, respectively. With accuracy scores of 95.16% for the CountVectorizer and 97.97% for the TF-IDF vectorizer, respectively, naive Bayes likewise performs well. SVM has the

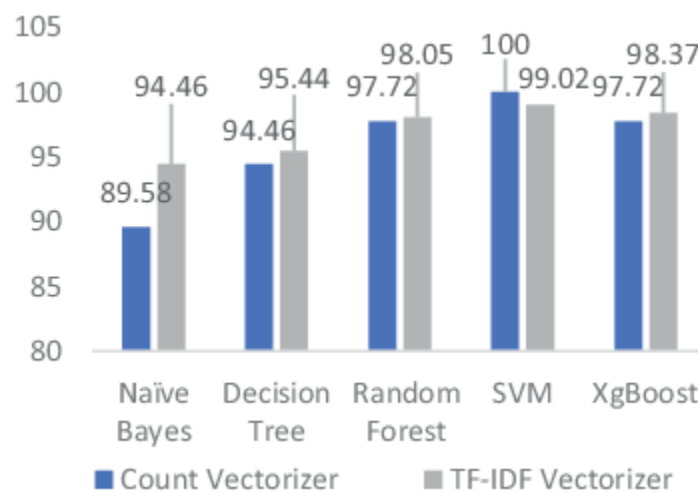
lowest precision with CountVectorizer at 82.31% but performs much better with TF-IDF vectorizer with precision of 95.90%. It is worth noting that the precision of SVM improved significantly with the use of TF-IDF vectorizer. Overall, the precision results suggest that RF and naive Bayes are strong algorithms for spam detection with both vectorizers, while SVM and decision tree perform relatively worse. XGBoost shows consistent precision results with both vectorizers but does not outperform RF or naive Bayes in this metric.



**Figure 3:** Precision comparison for vectorizers with models.

It can be seen by comparing the recall metrics for models employing the CountVectorizer and TF-IDF vectorizer in [→Figure 4](#) that the recall values for naive Bayes are 89.58% and 94.46% for the corresponding models. The recall values for decision tree are 94.46% and 95.44% for CountVectorizer and

TF-IDF vectorizer, respectively. The recall values for RF are 97.72% and 98.05% for CountVectorizer and TF-IDF vectorizer, respectively. The recall values for SVM are 100% and 99.02% for CountVectorizer and TF-IDF vectorizer, respectively. The recall values for XGBoost are 97.72% and 98.37% for CountVectorizer and TF-IDF vectorizer, respectively.

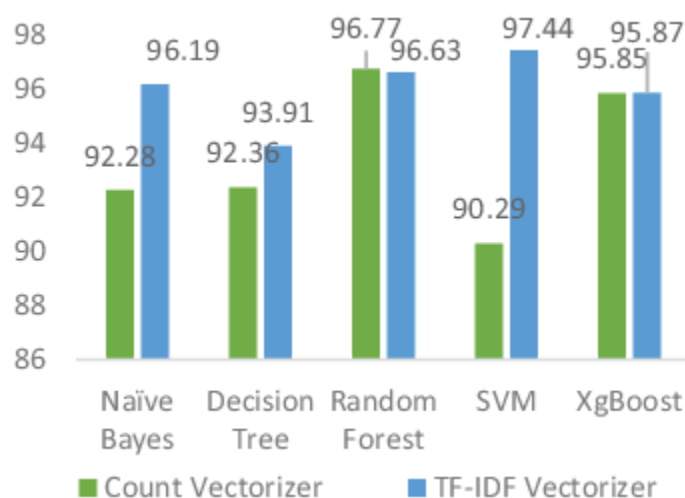


**Figure 4:** Recall comparison for vectorizers with models.

According to the aforementioned data, SVM has the greatest recall for both the CountVectorizer and the TF-IDF vectorizer. However, the recall values for all models using both vectorizers are relatively high, indicating that they are all effective at identifying TP.

SVM and RF utilizing TF-IDF vectorizer have the best *F1* scores among the algorithms, with 97.44% and 96.63%, respectively, when we compare the *F1* scores of the methods in [→figure 5](#).

With an  $F1$  score of 96.19%, naive Bayes utilizing TF-IDF vectorizer comes in second place. On the other hand, SVM and naive Bayes using CountVectorizer have the lowest  $F1$  scores at 90.29% and 92.28%, respectively. Decision tree, RF, and XGBoost using CountVectorizer have similar  $F1$  scores ranging from 92.36% to 96.77%. From the comparison of  $F1$  scores for both CountVectorizer and TF-IDF vectorizer, it can be inferred that the RF algorithm performed well in both cases. It achieved an  $F1$  score of 96.77% using CountVectorizer and 96.63% using TF-IDF vectorizer. Naive Bayes also performed well in both cases with an  $F1$  score of 92.28% using CountVectorizer and 96.19% using TF-IDF vectorizer. Although SVM has best  $F1$  score for TF-IDF vectorizer it performed poorly in CountVectorizer. XgBoost also performed well in both cases with an  $F1$  score of 95.85% using CountVectorizer and 95.87% using TF-IDF vectorizer.



**Figure 5:**  $F1$  score comparison for vectorizers with models.

Overall, based on the performance matrices provided in the tables, the RF algorithm appears to be the best algorithm for CountVectorizer, and for TF-IDF vectorizer SVM seems to be performing best. But it is noteworthy that there is not much difference between the performance of SVM and RF using TF-IDF vectorizer.

## **4.2 Time elapsed**

Time elapsed refers to the amount of time it takes for a machine learning algorithm to train and test on a given dataset. It is a useful metric in comparative analysis because it can give insight into the computational efficiency of different algorithms. In other words, if one algorithm takes significantly longer to train and test than another algorithm while producing similar results, it may not be the best choice for the task at hand. Time elapsed can also be used to identify if an algorithm is suitable for real-time applications that require quick response times. When comparing multiple algorithms, time elapsed can help in determining the most efficient algorithm in terms of both time and accuracy. However, it is important to note that time elapsed should not be the only metric considered in comparative analysis, as other factors such as model accuracy and interpretability should also be taken into account. In this section, we have presented the time elapsed to train

machine learning algorithms on the Enron Spam dataset with the CountVectorizer and TF-IDF vectorizer approaches. The time elapsed was measured in seconds, and the results were presented in a tabular format.

From the values obtained and given in [→Table 6](#), it can be observed that for the CountVectorizer and TF-IDF vectorizer approaches, naive Bayes was the quickest, while SVM was the slowest.

**Table 6:** Time elapsed for model training.

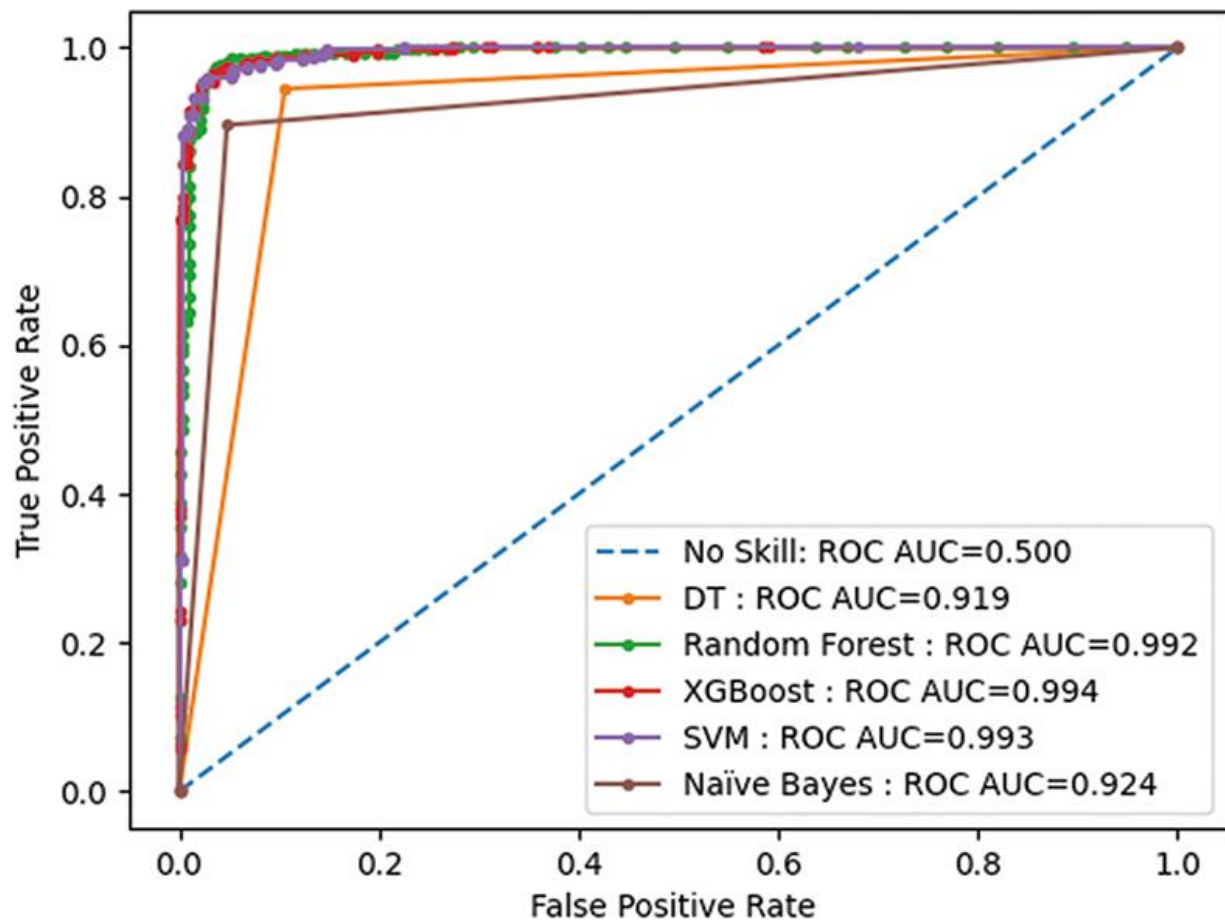
Algorithms	CountVectorizer	TF-IDF vectorizer
Naive Bayes	1.64 s	0.41 s
Decision tree	20.93 s	2.287 s
Random forest	19.85 s	2.420 s
SVM	202.96 s	5.37 s
XGBoost	59.95 s	4.06 s

### 4.3 AUC-ROC

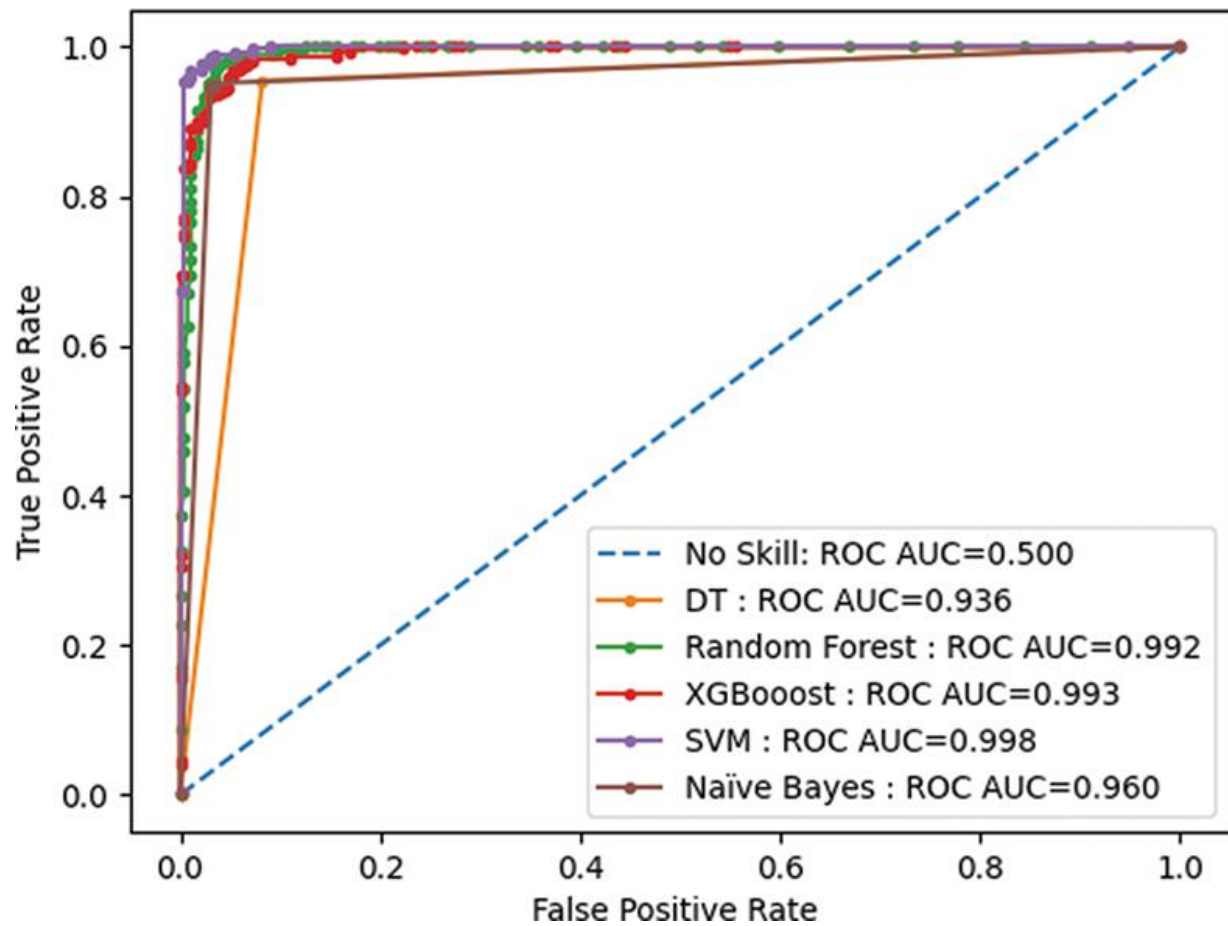
An evaluation statistic called AUC-ROC is used to gauge how well a model performs in binary classification. At different levels of classification, it compares the TPR and FPR. The proportion of positive instances the model properly identifies as positive is known as the TPR, while the proportion of negative cases the model incorrectly labels as positive is known as the FPR. Perfect classification yields an AUC-ROC value of 1, whereas random classification yields a score of 0.5. In general, a model performs better at differentiating between the positive and negative classes when it has a higher AUC-ROC score. Since it offers a single performance statistic that allows one to compare the efficiency of multiple models, AUC-ROC is valuable in comparative study of machine learning methods. Given that it takes into account every conceivable threshold, it is especially helpful when comparing models with various decision thresholds. The AUC-ROC scores of several models may be compared to see which one performs better at separating the positive and negative groups.

We have plotted the two graphs of AUC-ROC for both Count and TF-IDF vectorizers separately including all models are compiled in one graph. They can be observed in [→Figure 6](#) for models trained using CountVectorizer and [→Figure 7](#) for models

trained using TF-IDF vectorizer. Based on the values obtained we have compiled all values into [→Table 7](#) and all the algorithms performed well for both vectorizer techniques. However, SVM and RF show the highest AUC-ROC values for both CountVectorizer and TF-IDF vectorizer. Naive Bayes has the lowest AUC-ROC value for CountVectorizer, but it still performs well with a value of 0.924. For the TF-IDF vectorizer technique, naive Bayes has an AUC-ROC value of 0.96, which is the second-highest value after SVM and RF. Decision tree and XGBoost show similar AUC-ROC values for both techniques. The results suggest that SVM and RF could be the best options for spam mail detection using both vectorizer techniques. Overall, it can be inferred that all the models have performed well in both vectorizers. However, SVM and RF have shown slightly higher performance than the other models in both techniques.



**Figure 6:** AUC-ROC for models trained using CountVectorizer.



**Figure 7:** AUC-ROC for models trained using TF-IDF vectorizer.

**Table 7:** AUC-ROC for models trained.

Algorithms	CountVectorizer	TF-IDF vectorizer
Naive Bayes	0.924	0.960
Decision tree	0.919	0.936
Random forest	0.992	0.992
SVM	0.998	0.998
XGBoost	0.994	0.993

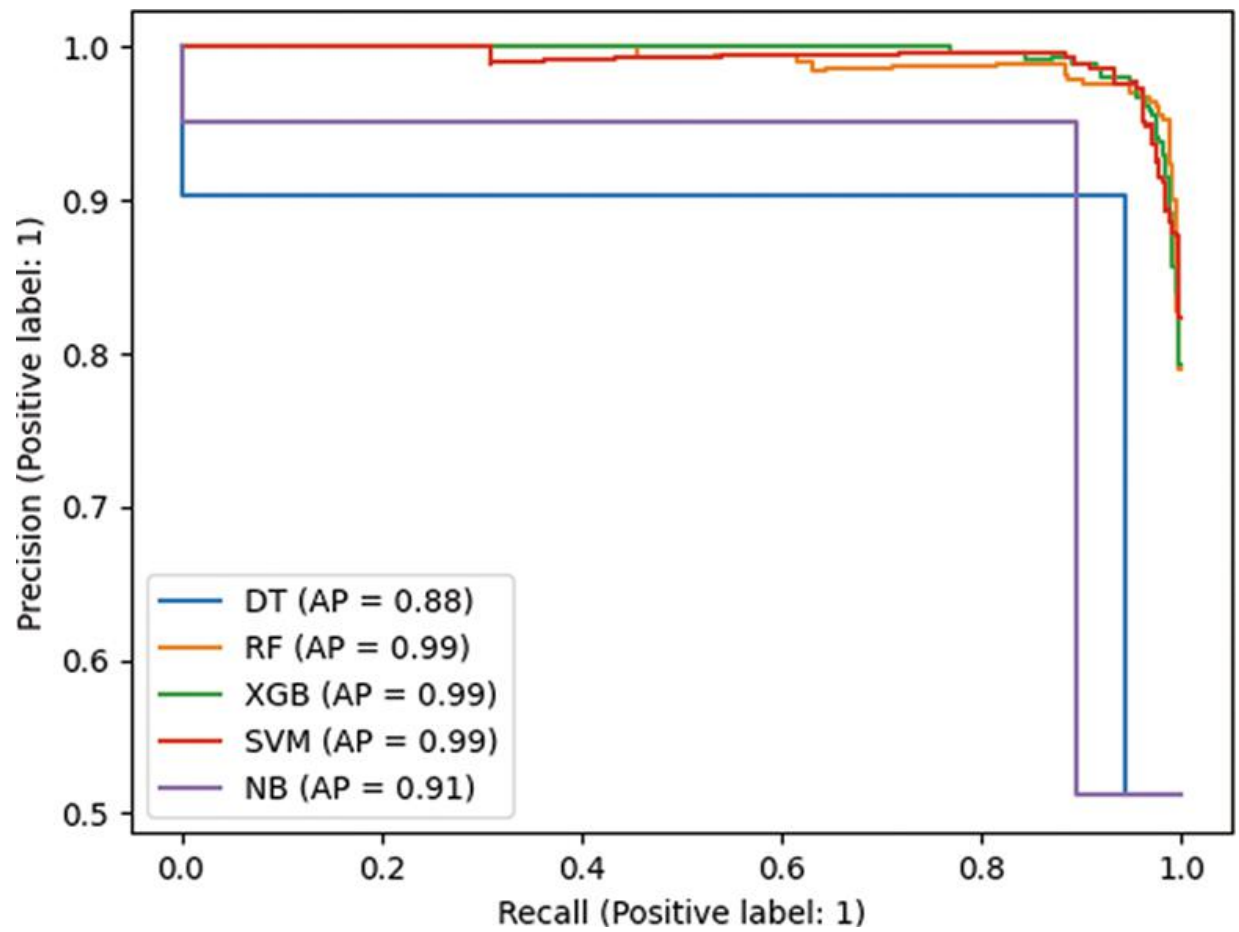
## 4.4 PR curve

The precision-recall trade-off for various threshold values in a binary classification task is shown graphically by the PR curve. It is helpful to compare machine learning algorithms and assess the effectiveness of various models. A step function with a beginning point of (0,0) and an ending point of (1,1) would be the perfect PR curve. The PR curve of a successful model should be as near to the ideal step function as feasible, demonstrating great accuracy and recall across all threshold values. The

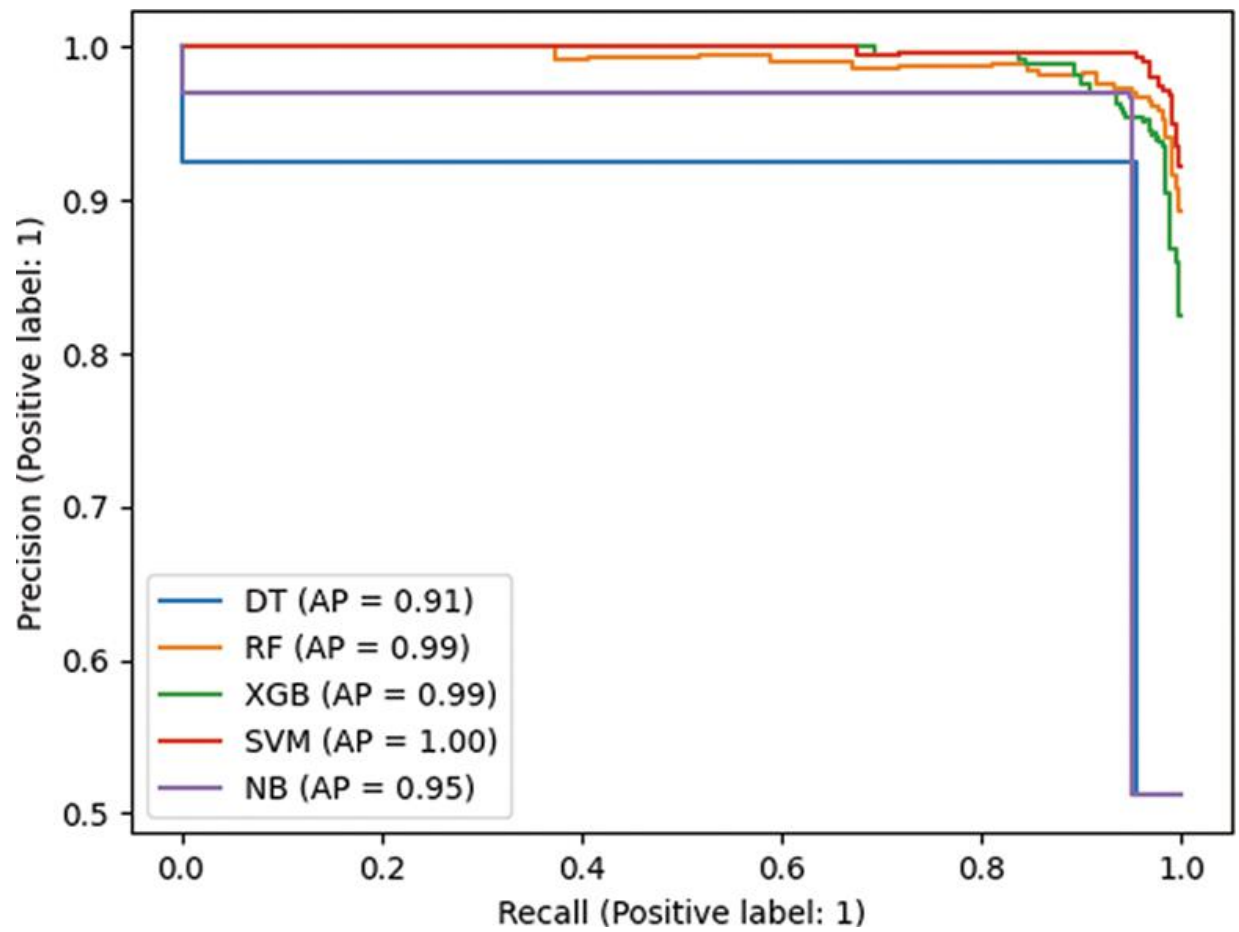
performance of a model across all threshold values is summed up by the area under the PR curve (AUC-PR), which is a single number. Better performance is indicated by a greater AUC-PR.

Comparing the AUC-PR of different models can help in selecting the best performing model for a given task.

We have plotted the two PR curves for both CountVectorizer and TF-IDF vectorizers separately in and 9, respectively. Based on that, PR curve values are compiled and represented in [→Table 8](#). From [→Table 8](#) it can be inferred that all algorithms perform well for both CountVectorizer and TF-IDF vectorizer. However, SVM has a perfect precision-recall trade-off for the TF-IDF vectorizer, indicating that it is the best algorithm for that TF-IDF vectorizer. For the CountVectorizer, RF has the highest precision-recall trade-off, making it the best algorithm for that vectorizer. Overall, both vectorizers benefit from the performance of all algorithms, and the selection of an approach relies on the particular needs and priorities of the work at hand.



**Figure 8:** PR curve for models trained using CountVectorizer.



**Figure 9:** PR curve for models trained using TF-IDF vectorizer.

**Table 8:** PR curve for models trained.

Algorithms	CountVectorizer	TF-IDF vectorizer
Naive Bayes	0.91	0.95
Decision tree	0.88	0.97
Random forest	0.99	0.99
SVM	0.99	1
XGBoost	0.99	0.99

## 5 Conclusion

The study explored different machine learning algorithms for spam mail detection using the Enron spam dataset from Kaggle. We preprocessed the dataset by removing punctuation, stop-words, header and footer, and HTML tags from the email data. We then randomly selected 3,000 emails out of the total dataset of 30,000 emails and split it into 80:20 for training and testing purposes. We used two NLP techniques, CountVectorizer and

TF-IDF vectorizer, for feature extraction and applied five different algorithms.

Using a variety of criteria, including accuracy, precision, recall, *F1* score, AUC-ROC, and PR curve, we assessed each algorithm's performance. From the results obtained, it can be inferred that the best-performing algorithm for CountVectorizer and TF-IDF vectorizer is RF based on *F1* score, AUC-ROC, and PR curve. We also observed that SVM has the highest accuracy and recall for TF-IDF vectorizer feature extraction techniques.

Furthermore, we compared the time elapsed for model training and found that naive Bayes is the fastest algorithm for both feature extraction techniques, while SVM is the slowest. Also, it is observed that TF-IDF vectorizer helped producing best results for all algorithms. The performance of all algorithms was enhanced, regardless of whether it was PR curve TF-IDF vectorizer or performance matrix time elapsed AUC-ROC.

Overall, this chapter demonstrates the importance of selecting the right algorithm and feature extraction technique for spam mail detection. The outcomes may be helpful for academics and professionals involved in machine learning and NLP.

## References

**[1]** F. Rustam, N. Saher, A. Mehmood, E. Lee, S. Washington and I. Ashraf, "Detecting Ham and Spam Emails Using Feature Union and Supervised Machine Learning Models," *Multimedia Tools and Applications*, vol. 82, no. 17, pp. 26545–26561, 2023. doi: 10.1007/s11042-023-14814-2. ➡

**[2]** Q. Ouyang, J. Tian and J. Wei, "E-mail Spam Classification Using KNN and Naive Bayes," *Highlights in Science, Engineering and Technology*, vol. 38, pp. 57–63, 2023. doi: 10.54097/hset.v38i.5699. ➡

**[3]** G. Gattani, S. Mantri and S. Nayak, "Comparative Analysis for Email Spam Detection Using Machine Learning Algorithms," *Modern Electronics Devices and Communication Systems*, pp. 11–21, 2023. doi: 10.1007/978-981-19-6383-4\_2. ➡

**[4]** U. Srinivasarao and A. Sharaff, "Spam Email Classification and Sentiment Analysis Based on Semantic Similarity Methods," *International Journal of Computational Science and Engineering*, vol. 26, no. 1, pp. 65, 2023. doi: 10.1504/ijcse.2023.129147. ➡

**[5]** S. Muthurajkumar and S. Rahmath Nisha, "Semantic Graph Based Convolutional Neural Network for Spam E-mail Classification in Cybercrime Applications," *International Journal*

of Computers Communications & Control, vol. 18, no. 1, 2023.  
doi: 10.15837/ijccc.2023.1.4478. ➡

**[6]** A. E. Karyawati, K. D. Y. Wijaya, I. W. Supriana and I. W. Supriana, "A Comparison of Different Kernel Functions of SVM Classification Method for Spam Detection," JITK (Jurnalilmupengetahuan Dan TeknologiKomputer), vol. 8, no. 2, pp. 91–97, 2023. doi: 10.33480/jitk.v8i2.2463. ➡

**[7]** V. S. Tida and S. Hsu, "Universal Spam Detection using Transfer Learning of BERT Model," 2022. doi: 10.48550/ARXIV.2202.03480. ➡

**[8]** S. Saha, S. DasGupta and S. K. Das, "Spam Mail Detection Using Data Mining: A Comparative Analysis," Smart Intelligent Computing and Applications, pp. 571–580, 2018. doi: 10.1007/978-981-13-1921-1\_56. ➡

**[9]** A. Karim, S. Azam, B. Shanmugam and K. Kannoorpatti, "Efficient Clustering of Emails into Spam and Ham: The Foundational Study of a Comprehensive Unsupervised Framework," IEEE Access, vol. 8, pp. 154759–154788, 2020. doi: 10.1109/access.2020.3017082. ➡

**[10]** G. Al-Rawashdeh, R. Mamat and N. Hafhizah Binti Abd Rahim, "Hybrid Water Cycle Optimization Algorithm with

Simulated Annealing for Spam E-mail Detection,” IEEE Access, vol. 7, pp. 143721–143734, 2019. doi: 10.1109/access.2019.2944089. [a](#), [b](#)

**[11]** A. Karim, S. Azam, B. Shanmugam and K. Kannoorpatti, “An Unsupervised Approach for Content-Based Clustering of Emails into Spam and Ham through Multiangular Feature Formulation,” IEEE Access, vol. 9, pp. 135186–135209, 2021. doi: 10.1109/access.2021.3116128. [a](#), [b](#)

**[12]** S. Gibson, B. Issac, L. Zhang and S. M. Jacob, “Detecting Spam Email With Machine Learning Optimized With Bio-Inspired Metaheuristic Algorithms,” IEEE Access, vol. 8, pp. 187914–187932, 2020. doi: 10.1109/access.2020.3030751. [a](#), [b](#)

**[13]** A. N. Soni, Spam-e-mail-detection-using-advanced-deep-convolution-neural-network-algorithms. SSRN, 2021.  
[→https://ssrn.com/abstract=3729703](https://ssrn.com/abstract=3729703) →

**[14]** M. Alauthman, “Botnet Spam E-Mail Detection Using Deep Recurrent Neural Network,” International Journal of Emerging Trends in Engineering Research, vol. 8, no. 5, pp. 1979–1986, 2020. doi: 10.30534/ijeter/2020/83852020. →

**[15]** S. Venkatraman, B. Surendiran and P. Arun Raj Kumar, “Spam E-mail Classification for the Internet of Things

Environment Using Semantic Similarity Approach,” The Journal of Supercomputing, vol. 76, no. 2, pp. 756–776, 2019. doi: 10.1007/s11227-019-02913-7. [a](#), [b](#)

**[16]** E. E. Eryilmaz, D. OzkanSahin and E. Kilic, “Machine Learning Based Spam E-mail Detection System for Turkish,” 2020 5th International Conference on Computer Science and Engineering (UBMK). IEEE, 2020. doi: 10.1109/ubmk50275.2020.9219487. [→](#)

**[17]** G. Hnini, J. Riffi, M. A. Mahraz, A. Yahyaouy and H. Tairi, “MMPC-RF: A Deep Multimodal Feature-Level Fusion Architecture for Hybrid Spam E-mail Detection,” Applied Sciences, vol. 11, no. 24, pp. 11968, 2021. doi: 10.3390/app112411968. [a](#), [b](#)

**[18]** Enron-Spam Dataset, Kaggle, 2019.  
[→https://www.kaggle.com/datasets/wanderfj/enron-spam](https://www.kaggle.com/datasets/wanderfj/enron-spam) [→](#)

**[19]** K. Napier, T. Bhowmik and S. Wang, “An Empirical Study of Text-based Machine Learning Models for Vulnerability Detection,” Empirical Software Engineering, vol. 28, no. 2, 2023. doi: 10.1007/s10664-022-10276-6. [→](#)

**[20]** I. Lasri, A. Riadsolh and M. Elbelkacemi, “Real-time Twitter Sentiment Analysis for Moroccan Universities Using Machine Learning and Big Data Technologies,” International Journal of

Emerging Technologies in Learning (Ijet), vol. 18, no. 05, pp. 42–61, 2023. doi: 10.3991/ijet.v18i05.35959. ➡

**[21]** R. Fatima, M. Sadiq, S. Ullah, G. Ahmed and S. Mahmood, “An Optimized Approach for Detection and Classification of Spam Email’s Using Ensemble Methods,” Research Square Platform LLC, 2023. doi: 10.21203/rs.3.rs-2051142/v1. ➡

**[22]** S. K. Singh and A. Goyal, “Performance Analysis of Machine Learning Algorithms for Cervical Cancer Detection,” Research Anthology on Medical Informatics in Breast and Cervical Cancer. IGI Global, pp. 347–370, 2022. doi: 10.4018/978-1-6684-7136-4.ch019. ➡

**[23]** B. Wu, X. Lv, A. Alghamdi, H. Abosaq and M. Alrizq, “Advancement of Management Information System for Discovering Fraud in Master Card Based Intelligent Supervised Machine Learning and Deep Learning during SARS-CoV2,” Information Processing & Management, vol. 60, no. 2, pp. 103231, 2023. doi: 10.1016/j.ipm.2022.103231. ➡

**[24]** A. Dhyani, A. Bansal, A. Jain and S. Seniaray, “Credit Card Fraud Detection Using Machine Learning and Incremental Learning,” Proceedings of International Conference on Recent

Trends in Computing. Springer Nature Singapore, pp. 337–349, 2023. doi: 10.1007/978-981-19-8825-7\_29. ➡

**[25]** Z. Salekshahrezaee, J. L. Leevy and T. M. Khoshgoftaar, “The Effect of Feature Extraction and Data Sampling on Credit Card Fraud Detection,” Journal of Big Data, vol. 10, no. 1, 2023. doi: 10.1186/s40537-023-00684-w. ➡

**[26]** S. González, W.-T. Hsieh and T. P.-C. Chen, “A Benchmark for Machine-learning Based Non-invasive Blood Pressure Estimation Using Photoplethysmogram,” Scientific Data, vol. 10, no. 1, 2023. doi: 10.1038/s41597-023-02020-6. ➡

**[27]** A. Gupta, J. Patil, S. Soni and A. Rajan, “Email Spam Detection Using Multi-head CNN-BiGRU Network,” Communications in Computer and Information Science. Springer Nature Switzerland, pp. 29–46, 2023. doi: 10.1007/978-3-031-28180-8\_3. ➡

**[28]** Z. Ye, et al., “The Prediction of In-hospital Mortality in Chronic Kidney Disease Patients with Coronary Artery Disease Using Machine Learning Models,” European Journal of Medical Research, vol. 28, no. 1, 2023. doi: 10.1186/s40001-023-00995-x.

➡

**[29]** A. N. Ahmed and R. Saini, "A Survey on Detection of Fraudulent Credit Card Transactions Using Machine Learning Algorithms," 2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT). IEEE, 2023. doi: 10.1109/icct56969.2023.10076122. ➡

**[30]** C. Cody, V. Ford and A. Siraj, "Decision Tree Learning for Fraud Detection in Consumer Energy Consumption," 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2015. doi: 10.1109/icmla.2015.80. ➡

**[31]** S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang and C. Jiang, "Random Forest for Credit Card Fraud Detection," 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC). IEEE, 2018. doi: 10.1109/icnsc.2018.8361343. ➡

**[32]** F. Itoo, Meenakshi and S. Singh, "Comparison and Analysis of Logistic Regression, Naïve Bayes and KNN Machine Learning Algorithms for Credit Card Fraud Detection," International Journal of Information Technology, vol. 13, no. 4, pp. 1503–1511, 2020. doi: 10.1007/s41870-020-00430-y. ➡

**[33]** J. Hancock and T. M. Khoshgoftaar, "Performance of CatBoost and XGBoost in Medicare Fraud Detection," 2020 19th IEEE International Conference on Machine Learning and

Applications (ICMLA). IEEE, 2020. doi:  
10.1109/icmla51294.2020.00095. ➡

**[34]** R. Butuner, I. Cinar, Y. S. Taspinar, R. Kursun, M. H. Calp and M. Koklu, "Classification of Deep Image Features of Lentil Varieties with Machine Learning Techniques," European Food Research and Technology, vol. 249, no. 5, pp. 1303–1316, 2023. doi: 10.1007/s00217-023-04214-z. ➡

# Cybersecurity threats in modern digital world

**Rahul Bijalwan**

**Vandana Rawat**

**Akshita Patwal**

**Sudhanshu Maurya**

---

**Acknowledgment:** The authors would like to give special thanks to the Department of Computer Application, Graphic Era University, for the tremendous support in this research work.

---

## **Abstract**

New and enhanced methods of protection against malware have been designated as an immediate priority by the cybersecurity community. This also includes the rising number of cyberattacks during the COVID-19 era and shows the problems faced by the organization due to data breaches which cause loss of secured data. The issue of cybersecurity has prompted the development of a number of frameworks and models. It also explains what cybersecurity is, how it works, and how to keep your personal data safe online. The effectiveness and limitations of current cutting-edge mitigation strategies are then discussed. We then take a look at how hackers are using emerging platforms like social media, the cloud, mobile devices, and key infrastructure to launch their attacks. This chapter gives the review about the most common cyberattack used by the hacker during COVID era.

**Keywords:** Cybersecurity, vulnerability, Sql, attacks, cyberthreads, Unsw-Nbis,

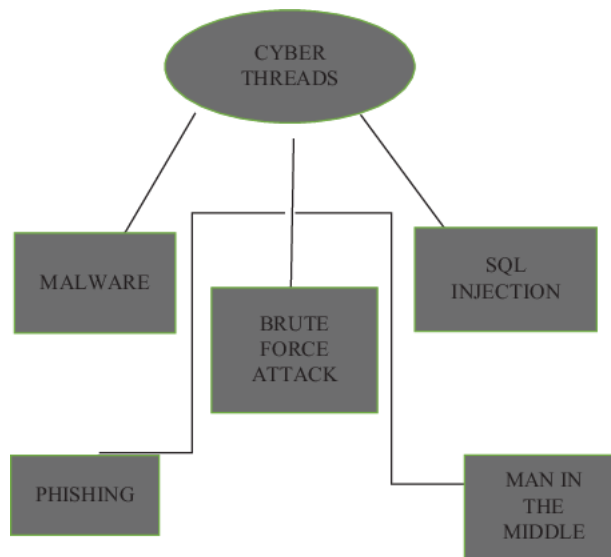
## **1 Introduction**

Cybercrime is a worldwide issue that has dominated the news cycle. It poses a threat to personal security and an even more significant threat to the protection of multinational corporations, banks, and governments [→1]. Crime rings operate like start-ups and frequently hire developers to continuously develop online attacks. In the past year, cybercriminals have unleashed a wave of attacks that were not only well-organized but also far more complex than anything ever seen. Attacks once limited to a single endpoint now require multiple steps to complete [→2, →3]. Both large and small organizations are vulnerable to ransomware attacks [→4].

Attacks involving crypto-mining provided cybercriminals with a simple entry point into corporate computer systems [→5]. This year saw numerous high-profile data breaches, substantial ransomware payouts, and the emergence of a broad and novel set of security challenges [→6]. And this year, cybercriminals significantly increased the danger they posed. Cybersecurity is a measure that protects computer systems, networks, and information from

disruption or unauthorized access, use, disclosure, modification, or destruction, according to our definition [→7].

This chapter summarizes the cybersecurity threads and models. It includes datasets and data breaches occurred during the COVID-19 era, and how this caused the reduction of efficiency in organizations [→8]. This chapter helps academics and professionals to understand contemporary cybersecurity [→9]. The different types of cybersecurity are displayed in →Figure 1.



**Figure 1:** Various types of cyberattacks.

## 2 Literature review

This chapter shows that the number of cyberattacks is increasing day by day. However, the security and encryption technology is also improved in the past decade [→10]. This chapter also shows the various types of attacks used for data breaches. It also shows [→11] the increasing challenges faced by organization from data breaches [→12].

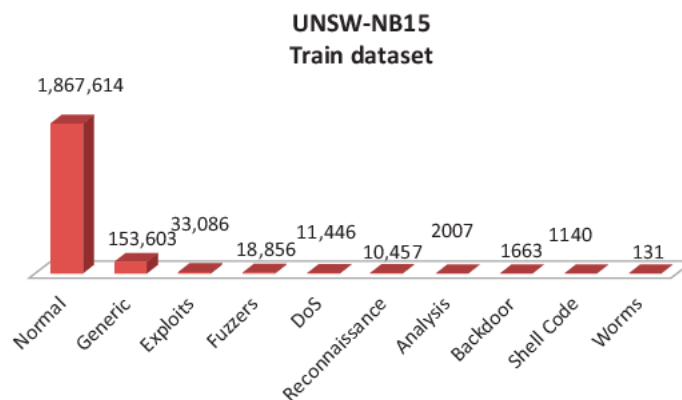
In recent cybersecurity reports, the attackers are using new technology algorithms to access data easily. The technologies used by attackers are cryptography and emerging machine learning [→13, →14]. Cyberattacks also increase the use of new programming languages like Rust, Ruby, Sql, and python because of their low compile time and encryption.

There have been incredible technological breakthroughs in the digital era, but there is also a much more complicated environment of cybersecurity dangers. The breadth and severity of these challenges are shown in this chapter. From advanced persistent threats to new dangers in

AI-driven security, no area is safe. Organizations and individuals need to take a multipronged strategy to tackle these issues, one that incorporates technical solutions, increased awareness and education, new legal frameworks, and increased international cooperation. Continued study and adjustments will be required to remain ahead of the constantly shifting cyber threats of the digital age [→15].

### 3 Issues in cybersecurity

Cybersecurity involves understanding cyberattacks and devising defense strategies (i.e., countermeasures) that preserve digital and information technology confidentiality, integrity, and availability. Some of the malware attacks [→16] datasets are shown in →Figure 2.



**Figure 2:** Malware attacks on UNSW-NB15 train dataset.

1. Confidentiality prevents unauthorized individuals or systems from accessing information.
2. Integrity prevents unauthorized changes/deletions.
3. Availability ensures that information-delivery, storage, and processing systems are accessible when needed [→17].

Malware is widely regarded as a vital tool for cybercriminals by many industry experts [→18]. Malware refers to a category of attacks installed on a system secretly for the benefit of an opponent. Malware includes programs like viruses, worms, Trojan horses, spyware, and bot executables [→19]. Infected computers can then infect other computers, users can be tricked into opening infected files, or users can be led to malicious websites. Malware can spread to other computers when a USB drive is introduced into one that is already infected. Malware can be disseminated via embedded systems and computational logic. Any point in time is a potential entry point for malware [→20]. In addition to infecting end-user devices, servers, and network gear like routers and switches, malware can also make its way into SCADA systems used for

controlling industrial processes. The development and spread of malware are serious issues in the modern Internet [[→21](#)].

Historically, malware assaults took use of flaws in the hardware, software, and network layers [[→22](#)]. Perimeter defense creates an outside barrier around an organization's resources to deter outsiders from gaining access to those resources. Firewalls and antivirus programs are commonly used in perimeter security systems. All incoming data from the outside world is filtered and checked for viruses. Due to the ease and lower cost of securing a single perimeter [[→23](#)], this style of defense has gained widespread acceptance. Perimeter security and access control systems are used to regulate who has access to what within an organization. Perimeter defense has been ineffective as malware evolves and becomes more complex. Hackers are continuously discovering new ways for malware to escape detection.

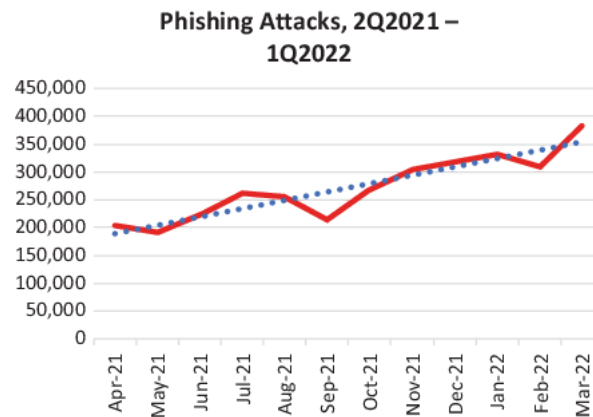
## 4 Attacks with their classification

### 4.1 Phishing

The growing number of people using the internet has led to an increase in the number of security issues [[→24](#)]. Phishing attacks are carried out, for the most part, to obtain the personal information of the user as well as access to the user's private accounts. Since COVID-19 occurred, attackers have thought of innovative ways to attract a variety of internet users, such as using a variety of articles involving the subject matter of the pandemic [[→25](#), [→26](#)]. The articles or links that the attackers provide may give the impression that they are genuine and trustworthy; however, in the end, private information may be given to individuals engaging in phishing activities.

Most phishing attackers target uninformed people. Internet users who are familiar with phishing are not typical victims [[→27](#), [→28](#)]. Phishers create enticing websites, links, or articles in order to steal sensitive information from unsuspecting Internet users [[→29](#)]. Intruders can then utilize this knowledge to steal more sensitive data or money [[→30](#)].

There has been prior research into spam and spam filtering. Bayesian analysis, blacklist/whitelist, keyword verification, and email header analysis are all used together to identify incoming spam. Spam filtering strategies are based on message content: Support vector machine; Bayesian classifier (BC); K-nearest neighbors; distributed adaptive blocklists (DABLs). There is no foolproof text filtering method currently available. [→Figure 3](#) shows the phishing attacks occurred in 2021–2022.



**Figure 3:** Phishing attacks occurred in 2021–2022.

## 4.2 SQL injection

SQL-infected code gets access to the application which allows the hacker to modify the database. The effects of an SQL injection attack on a business can be devastating. SQL injection attacks frequently target organizations because of their access to private company data and customer information. Any of the following can happen if a malicious user can complete an SQL injection attack: Using SQL injection, attackers can access and modify data on the SQL server [→31], potentially exposing sensitive company information.

User privacy is at risk because an SQL server attack could reveal sensitive information like credit card numbers. To counteract this risk, you should create a database user with minimum permissions possible. If you use insecure SQL commands to verify user credentials, an attacker could gain access to your entire system. An intruder can do even more harm by gaining access to and manipulating private data once they completely control your system [→32].

SQL injection allows attackers to insert, update, or delete data from your system without your knowledge. Businesses need to take precautions and reduce their exposure to SQL injection attacks because the consequences of a successful attack can be significant. One must be familiar with the inner workings of an SQL injection attack to effectively defend against such an assault [→33]. Consider the following SQL code, which a web interface would use to display all records from the database “Users” in response to a user-supplied username and password: To select all users who match the criteria:

Password = “\$password” AND Username = “\$username”

When requested for login information on a website, a malevolent user could enter the following:

Ingest \* FROM Users AND SELECT \* WHERE The correct credentials are : Username =

The above code injects an OR condition into the authentication procedure, allowing the attacker to bypass security measures. Since “1” = “1” is always authentic, using this SQL query is the same as using no authentication.

### 4.3 Brute force attack

Attempting to crack security features, such as a password or login credentials, through sheer repetition of the process is known as a brute force attack. It's a foolproof method for breaking into people's personal accounts and corporate networks. Hackers try every conceivable combination until they succeed.

Brute-force assaults occur when an attacker repeatedly and aggressively tries to gain access to a system, usually a private one(s). This is a tried and true hacker favorite despite being decades old because the time required to crack a password can range from a few seconds to several years, depending on its length and complexity.

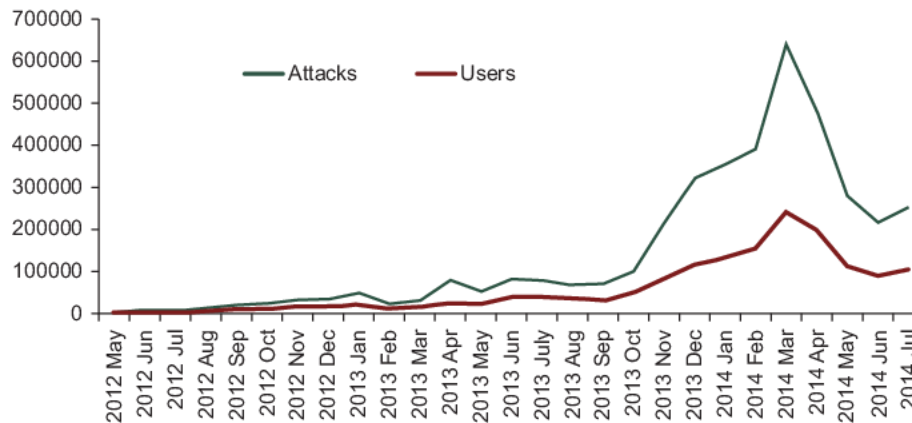
### 4.4 Malware

Malware is a type of software that has been designed to affect your computer system or network. It may steal your identity, destroy files, or take control of the system. A variety of ways, including email attachments, downloads, and infected websites, can be used to spread malware.

Viruses are the most commonly used type of malware. They're linked to other files and programs, spreading as soon as they open them. Viruses are capable of damaging files and stealing sensitive information [[→34](#)].

*Trojan:* Trojan is a malicious program that's disguised as legitimate software. The Trojan can steal personal information, destroy files, or even control the system when it is downloaded and started by a user.

*Rootkits:* A type of malicious software that enables an attacker to have complete control over the system. It's hard to detect and remove rootkits, and they can be used for stealing personal data, files that have been damaged, or even controlling the system. [→Figure 4](#) shows the number of attacks increase in the past decade.



**Figure 4:** Malware attacks increase in decade.

## 4.5 Man-in-the-middle

In MIM attacker intercepts and eavesdrops on conversations between two parties, potentially modifying or stealing sensitive information.

*Application-level attack:* An attacker can exploit weaknesses in a firewall's web or email server to obtain unauthorized access or compromise data.

*Buffer overflow attack:* Sending extra data to a firewall crashes or makes it subject to further exploitation.

*Exploitation of configuration errors:* Misconfigured firewalls might allow attackers to obtain unauthorized access to a network. Organizations must update and maintain their firewalls to prevent these and other assaults.

This involves updating firmware and security patches, installing firewalls, and monitoring logs and warnings for unexpected activities.

## 5 Data breaches

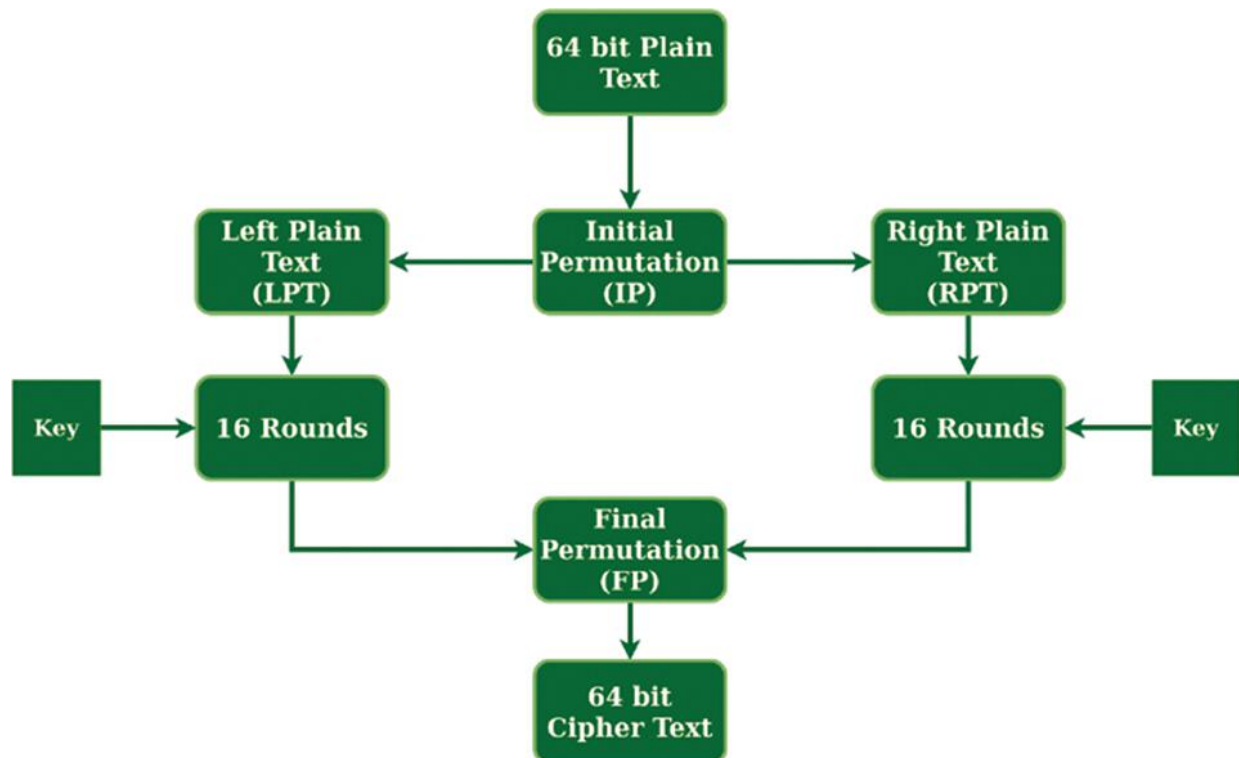
When protected, important, and confidential data of any organization is accessed by any unethical group, this is known as data breach. The following terms are also used: inadvertent disclosure, data leakage, information leaks, and data degradation. Any organization, regardless of size or industry, may be subject to a data breach. Any organization, regardless of size or industry, may be subject to a data breach. A variety of factors can lead to this:

*Human error:* This is the most frequent reason for a data breach. Human error may include things like an accidental click to a malicious link, the opening of infected attachments, or not

following security procedures.

*Hacking:* By exploiting vulnerabilities in software or systems, hackers are often able to gain access to sensitive data.

*Insider threat:* If an employee or contractor intentionally or accidentally steals or leaks sensitive information, an insider threat can occur. [→Figure 5](#) and [→Table 1](#) show that the data breaches occur in over the decade.



**Figure 5:** Working of encryption.

**Table 1:** Data breaches occur due to cyberattacks.

Twenty-first century oncology	2016	2,200,000	Healthcare	Hacked
Intel	2020	4,970,604	Hardware	Hacked
Accendo Insurance Co.	2020	175,350	Healthcare	Poor security
Adobe Inc.	2013	152,000,000	Tech	Hacked
Advocate Medical Group	2019	7,500,000	Healthcare	Lost/stolen
Airtel	2017	4,000,000	Telecommunication	Poor security
Amazon Japan Gk	2018	75,000	Web	Accidently published

## 6 Conclusion

The COVID-19 epidemic has become a useful tool for cybercriminals. There has been a rise in cyberattacks after the appearance of COVID-19. Primary objectives are institutions in the healthcare industry that own significant patient data. The main purpose of these attacks on any organization is to affect their services; due to this organization faces lots of problems. Huge amount of money is charged by the attackers for resuming the services of organizations.

Increasing work from home business in COVID-19 era faced this type of issues. When workers are in the comfort of their own homes, they may be less inclined to follow safety protocols. The targets of the attacks include healthcare providers and pandemic management agencies like the World Health Organization. Since it is now simple to send emails that cover their tracks by concealing information about COVID-19, phishing has become the primary method of attack. To better identify the tendencies and protect against phishing assaults, we investigated the use of neural network to optimize data on these attacks as depicted in [→Figure 5](#).

## References

- [1] S. N. Brohi, N. Z. Jhanjhi, N. N. Brohi and M. N. Brohi, "Key Applications of State-of-the-Art Technologies to Mitigate and Eliminate COVID-19," 2020. ➔
- [2] S. Perez, "Videoconferencing apps saw a record 62M downloads during one week in March," 2020. Available: ➔<https://techcrunch.com/2020/03/30/video-conferencing-apps-saw-a-record-62m-downloads-during-one-week-in-march/>. ➔
- [3] B. Vigliarolo, "Who has banned Zoom? Google, NASA, and more," 2020. ➔
- [4] D. Craigen, N. Diakun-Thibault and R. Purse, "Defining Cybersecurity," Technology Innovation Management Review, vol. 4, no. 10, pp. 13–21, 2014. ➔
- [5] E. A. Fischer, "Cybersecurity issues and challenges: In brief," 2014. ➔
- [6] G. N. Reddy and G. J. Reddy, "A Study of Cyber Security Challenges and Its Emerging Trends on Latest Technologies," 2014, arXiv preprint arXiv:1402.1842. ➔
- [7] Y. Harel, I. B. Gal and Y. Elovici, "Cyber Security and the Role of Intelligent Systems in Addressing Its Challenges," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 8, no. 4, pp. 1–12, 2017. ➔
- [8] K. M. Rajasekharaiah, C. S. Dule and E. Sudarshan, "Cyber Security Challenges and Its Emerging Trends on Latest Technologies," in IOP Conference Series: Materials Science and Engineering (Vol. 981, No. 2, p. 022062). IOP Publishing. ➔
- [9] I. Frank and E. Odunayo, "Approach to Cyber Security Issues in Nigeria: Challenges and Solution," International Journal of Cognitive Research in Science, Engineering and Education, vol. 1, no. 1, pp. 100–110, 2013. ➔
- [10] L. Tawalbeh, F. Muheidat, M. Tawalbeh and M. Quwaider, "IoT Privacy and Security: Challenges and Solutions," Applied Sciences, vol. 10, no. 12, pp. 4102, 2020. ➔
- [11] T. Mais, M. Quwaider and A. Tawalbeh Lo'ai, "Authorization Model for IoT Healthcare Systems: Case Study," in 2020 11th International Conference on Information and Communication Systems (ICICS), IEEE, pp. 337–342, 2020. ➔
- [12] Song and A. Kunz, "Towards Standardized Prevention of Unsolicited Communications and Phishing Attacks," Journal of ICT Standardization, pp. 109–122, 2021. ➔

**[13]** S. Nasiri, M. Tahghighi Sharabian and M. Aajami, "Using Combined One-Time Password for Prevention of Phishing Attacks," *Engineering, Technology & Applied Science Research*, vol. 7, no. 6, pp. 2328–2333, 2017. ➡

**[14]** A. Alhashmi, A. Darem and J. Abawajy, "Taxonomy of Cyber Security Awareness Delivery Methods: A Countermeasure for Phishing Threats," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 10, 2021. ➡

**[15]** A. Baiomy, M. Mostafa and A. Youssif, "Anti-Phishing Game Framework to Educate Arabic Users: Avoidance of URLs Phishing Attacks," *Indian Journal of Science and Technology*, vol. 12, no. 44, pp. 01–10, 2019. ➡

**[16]** M. Jensen, M. Dinger, R. Wright and J. Thatcher, "Training to Mitigate Phishing Attacks Using Mindfulness Techniques," *Journal of Management Information Systems*. ➡

**[17]** F. Benevenuto, T. Rodrigues, A. Veloso, J. Almeida, M. Gonçalves and V. Almeida, "Practical Detection of Spammers and Content Promoters in Online Video Sharing Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 3, pp. 688–701, 2012. ➡

**[18]** M. Cha, F. Benevenuto, H. Haddadi and K. Gummadi, "The World of Connections and Information Flow in Twitter," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 4, pp. 991–998, 2012. ➡

**[19]** S. Delany, M. Buckley and D. Greene, "SMS Spam Filtering: Methods and Data," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9899–9908, 2012. ➡

**[20]** R. Kumar, G. Poonkuzhali and P. Sudhakar, "Comparative Study on Email Spam Classifier Using Data Mining Techniques," in *The International Multiconference of Engineers and Computer Scientists*, volume 1, Hong Kong, China, pp. 14–16, 2012. ➡

**[21]** C. Ten, C. Liu and G. Manimaran, "Vulnerability Assessment of Cybersecurity for Scada Systems," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1836–1846, 2008. ➡

**[22]** B. Alfayyadh, J. Ponting, M. Alzomai and A. Jøsang, "Vulnerabilities in Personal Firewalls Caused by Poor Security Usability," in *IEEE International Conference on Infor. Theor. and Infor. Security*, Beijing, China: IEEE, pp. 682–688, 2010. ➡

**[23]** J. Li, "The Research and Application of Multi-firewall Technology in Enterprise Network Security," *International Journal of Security and Its Applications*, vol. 9, no. 5, pp. 153–162, 2015. ➡

- [24] N. Rani, A. Satyanarayana and P. Bhaskaran, "Coastal Vulnerability Assessment Studies over India: A Review," *Natural Hazards*, vol. 77, no. 1, pp. 405–428, 2015. ➡
- [25] X. Ye, J. Zhao, Y. Zhang and F. Wen, "Quantitative Vulnerability Assessment of Cyber Security for Distribution Automation Systems," *Energies*, vol. 8, no. 6, pp. 5266–5286, 2015. ➡
- [26] H. Sun, Y. Chen and Y. Lin, "Opass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 651–663, 2012. ➡
- [27] H. Hu, G. Ahn and K. Kulkarni, "Detecting and Resolving Firewall Policy Anomalies," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 3, pp. 318–331, 2012. ➡
- [28] K. Gai, M. Qiu, B. Thuraisingham and L. Tao, "Proactive Attribute Based Secure Data Schema for Mobile Cloud in Financial Industry," in *The IEEE International Symposium on Big Data Security on Cloud*, New York, USA, 1332–1337, 2015. ➡
- [29] K. Gai, M. Qiu, L. Tao and Y. Zhu, "Intrusion Detection Techniques for Mobile Cloud Computing in Heterogeneous 5G," *Security and Communication Networks*, pp. 1–10, 2015. ➡
- [30] L. Tao, S. Golikov, K. Gai and M. Qiu, "A Reusable Software Component for Integrated Syntax and Semantic Validation for Services Computing," in *9th Int'l IEEE Symposium on Service-Oriented System Engineering*, San Francisco Bay, USA, pp. 127–132, 2015. ➡
- [31] K. Cabaj, Z. Kotulski, B. Księżopolski and W. Mazurczyk, "Cybersecurity: Trends, Issues, and Challenges," *EURASIP Journal on Information Security*, vol. 2018, no. 1, pp. 1–3, 2018. ➡
- [32] L. Thames and D. Schaefer, *Cybersecurity for Industry*, Heidelberg: Springer, 2017, pp. 1–33. ➡
- [33] H. Santos, T. Pereira and I. Mendes, "Challenges and Reflections in Designing Cyber Security Curriculum," in *2017 IEEE World Engineering Education Conference (EDUNINE)*, IEEE, pp. 47–51, 2017. ➡
- [34] Z. Zhang, H. Ning, F. Shi, F. Farha, Y. Xu, J. Xu ... K. K. R. Choo, "Artificial Intelligence in Cyber Security: Research Advances, Challenges, and Opportunities," *Artificial Intelligence Review*, pp. 1–25, 2022. ➡

# Mechanism to protect the physical boundary of organization where the private and public networks encounter

**Parth Gautam**

**Apurva Omer**

**Jeetendra Pande**

**Devesh Bora**

## **Abstract**

With the increasing connectivity and reliance on digital networks, organizations face a growing challenge in protecting their physical boundaries where private and public networks intersect. This study explores the mechanisms that organizations can employ to safeguard these boundaries, considering the unique security risks associated with the convergence of private and public networks. The chapter discusses various approaches including network segmentation, access control, intrusion prevention, and physical security measures. By understanding and implementing these mechanisms effectively, organizations can enhance their overall security posture and mitigate potential threats.

**Keywords:** Physical boundary protection, private-public network convergence, network security, virtual private network, security protocols,

## **1 Introduction**

In today's interconnected world, organizations are all the time more reliant on both private and public networks to conduct their operations. Private networks provide a secure and controlled environment for internal communication and data exchange, while public networks, such as the Internet, facilitate global connectivity and information sharing. However, the convergence of these networks at the physical boundary of an organization presents unique security challenges.

The physical boundary where private and public networks encounter is a critical point of vulnerability. It serves as a gateway for potential threats to enter an organization's internal infrastructure compromising the integrity, availability, and privacy of sensitive data and systems. Attackers can exploit this convergence to launch various attacks, such as unlawful access attempts, data breaches, malware infections, and distributed denial-of-service (DDoS) attacks.

## **2 Literature review**

The paper by Saklani and Dimri [[→1](#)] provides a technical comparison between IPv4 and IPv6, emphasizing the limitations of IPv4 and the benefits of adopting IPv6. IPv6 offers a larger address space, enhanced security features, and improved efficiency. The paper also discusses the challenges and strategies involved in migrating from IPv4 to IPv6. Dobhal and Dimri [[→2](#)] present a design and implementation of a TCP-friendly layered solution for mobile ad hoc networks (MANET). The solution aims to improve the performance of TCP-based applications in MANET scenarios. The research addresses the challenges of congestion control and fairness in MANETs, contributing to the efficiency of data transmission. Goki [[→3](#)] propose a novel approach for network authentication, identification, and secure communication using optical physical unclonable function (OPUF). OPUFs leverage the unique properties of optical devices to establish secure communication channels, ensuring confidentiality and integrity of data transmission. Singhal's survey paper [[→4](#)] delves into security issues on mobile cloud computing (MCC) and explores preventive measures. The research highlights various threats such as data breaches, unauthorized access, and data loss in MCC environments. The preventive measures discussed aim to fortify the security posture of MCC systems.

Bahuguna et al.'s study [[→5](#)] presents a comprehensive assessment of country-level cybersecurity practices. The research analyzes the cybersecurity measures and practices adopted by different countries, providing insights into various strategies to strengthen national cyber defense. In another study, Bahuguna et al. [[→6](#)] investigate the cybersecurity maturity of organizations in the Indian context. The research assesses the current state of cybersecurity practices in organizations, identifying areas for improvement and offering valuable recommendations. Marin [[→7](#)] offers a primer on network security fundamentals. The paper provides an overview of security principles, mechanisms, and best practices to safeguard networks from potential threats. These Cisco documents [[→8](#), [→9](#)] serve as guides for deploying and configuring Cisco's adaptive security virtual appliance (ASAv) and secure firewall ASA virtual, which are critical components of network security infrastructures. The configuration guide [[→10](#)] by Cisco Systems covers the software configuration aspects of Cisco 850 Series and 870 Series Access Routers, which are widely used for network connectivity and security. This guide [[→11](#), [→12](#)] by Cisco Systems elaborates on the network address translation (NAT) configurations in Cisco IOS Release 12.4T. NAT plays a crucial role in enhancing network security by concealing internal IP addresses.

Natalino [[→13](#)] proposes an autonomous security management approach for optical networks. The research focuses on self-protecting and self-healing mechanisms to mitigate security threats in optical communication systems. Shin et al. [[→14](#)] present the concept and prototype of network security virtualization, an innovative approach to improving the efficiency and scalability of network security solutions. Lara and Ramamurthy [[→15](#)] introduce OpenSec, a policy-based security framework that leverages software-defined networking (SDN) to enhance network security management and control. Hu [[→16](#)] proposes an adaptive secure transmission mechanism for physical layer security in cooperative wireless networks. The research focuses on leveraging the physical layer properties to enhance wireless communication security. Pande and Maheshwari [[→17](#), [→18](#)] present an identity-based encryption algorithm that employs hybrid encryption and MAC address for secure key generation, ensuring confidentiality and integrity in data transmission.

This literature review has explored various aspects of network security and related technologies including IPv6 migration, secure communication, VPNs, and SDN. The surveyed papers and documents provide valuable insights into the challenges, solutions, and best practices for enhancing network security and mitigating cyber threats. The research works collectively

contribute to a deeper understanding of the evolving field of network security and the measures required to safeguard modern network infrastructures.

### **3 Purpose and objectives**

This chapter aims to explore the mechanisms that organizations can employ to protect the physical boundary where private and public networks intersect. By understanding and implementing effective security measures, organizations can safeguard their assets and minimize the risks associated with network convergence. The objectives of this chapter are as follows:

- Provide an overview of private and public networks, highlighting their characteristics, security considerations, and the challenges that arise at their intersection.
- Discuss the concept of network segmentation as a mechanism to protect the physical boundary. This includes exploring techniques such as virtual local area networks (VLANs), demilitarized zones (DMZ), and virtualization as well as the use of firewalls and intrusion prevention systems (IPS) to enforce segmentation.
- Explore access control mechanisms that regulate the flow of traffic between private and public networks. This includes

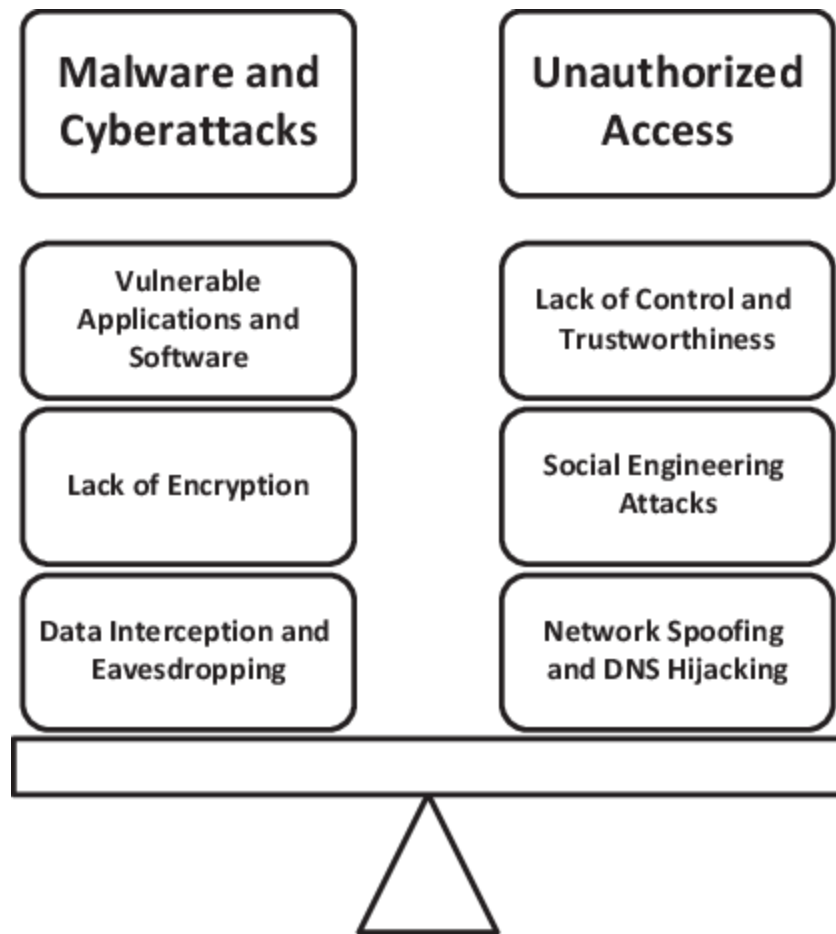
authentication, authorization, and role-based access control (RBAC) and network access control (NAC) solutions.

- Examine the role of IPS in identifying and mitigating potential threats at the physical boundary. This includes discussing network-based intrusion prevention systems (NIPS) and host-based IPS (HIPS) as well as signature-based and anomaly-based detection techniques.
- Investigate the significance of physical security measures in protecting the physical boundary. This includes restricted access areas, visitor management, surveillance systems, environmental controls, and power protection.
- Advocate for a defense-in-depth approach that combines multiple layers of security mechanisms to strengthen the protection of the physical boundary. This includes encryption, virtual private network (VPN) technologies, security awareness training, incident response planning, and regular security assessments.

By addressing these objectives, this chapter aims to provide organizations with insights into effective mechanisms to protect the physical boundary where private and public networks encounter. By implementing these mechanisms, organizations can enhance their overall security posture, mitigate potential threats, and ensure the integrity and confidentiality of their critical assets.

## 4 Overview of private and public networks

Private networks are internal networks within an organization that are designed to facilitate secure communication and data exchange among authorized users and devices. These networks are typically isolated from public networks, such as the internet; provide global connectivity, to maintain confidentiality, integrity, and availability of sensitive information. While they offer numerous benefits, they also introduce significant risks and vulnerabilities that organizations must be aware of when protecting the physical boundary that private and public networks encounter. Understanding these risks is essential for implementing effective security measures. →[Figure 1](#) shows risks associated with both private and public networks:



**Figure 1:** Risks associated with both private and public networks.

- *Unauthorized access:* Public networks are inherently open and accessible to anyone with an internet connection. This accessibility increases the risk of unauthorized individuals or malicious entities attempting to gain access to an organization's network resources, sensitive data, or confidential information.
- *Malware and cyberattacks:* Public networks are prime targets for cybercriminals who deploy various malware, viruses, and ransom-ware to exploit vulnerabilities in systems and

compromise security. Attacks like phishing, denial of service, and man-in-the-middle are common threats on public networks.

- *Data interception and eavesdropping:* Public networks transmit data over shared infrastructure, making it possible for attackers to intercept and eavesdrop on network traffic. This can lead to unauthorized access to sensitive information including passwords, financial data, and intellectual property.
- *Network spoofing and DNS hijacking:* Attackers can spoof public networks by creating fake access points that mimic legitimate networks. Users unknowingly connect to these spoofed networks, allowing attackers to intercept their data or redirect them to malicious websites. Similarly, DNS hijacking can redirect users to fraudulent websites, enabling the theft of credentials or sensitive information.
- *Lack of encryption:* Public networks often lack adequate encryption, exposing transmitted data to potential interception and unauthorized access. This is especially true for unsecured Wi-Fi networks commonly found in public places such as cafes, airports, and hotels.
- *Social engineering attacks:* Public networks provide a fertile ground for social engineering attacks, where attackers manipulate individuals to reveal sensitive information.

Phishing emails, fake websites, and social media scams are common tactics used on public networks to deceive users.

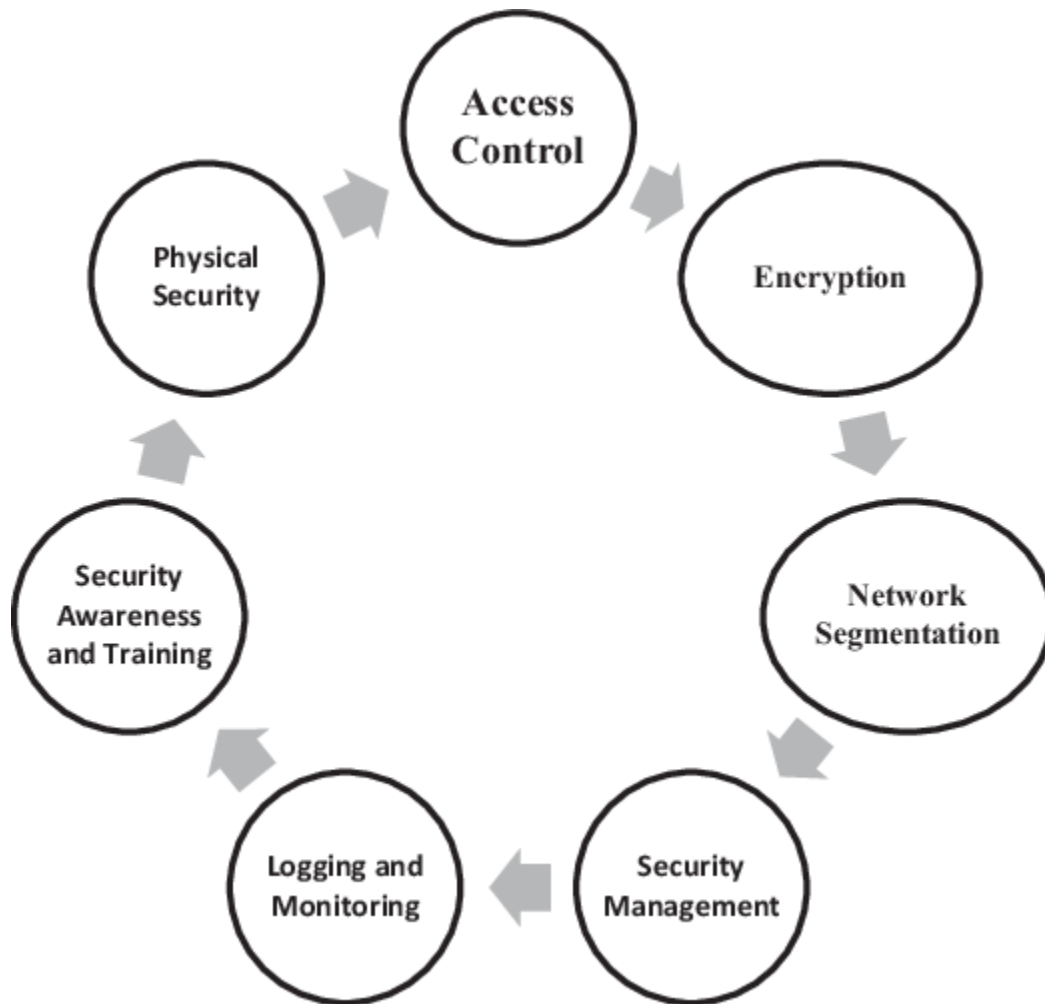
- *Vulnerable applications and software:* Public networks expose connected devices to a wide range of potentially vulnerable applications and software. Unpatched or outdated software can be misused by attackers to gain unlawful access or execute malicious code on devices.
- *Lack of control and trustworthiness:* Organizations have limited control over the security measures implemented on public networks. They must rely on the service providers and network administrators to ensure the network's integrity and security. However, there may be instances where these entities fail to maintain adequate security controls.
- *Compliance and legal issues:* Utilizing public networks introduces compliance challenges, especially in industries with strict data protection regulations. Data traversing public networks may cross international boundaries, necessitating compliance with different data protection laws and regulations.

To mitigate the risks and vulnerabilities associated with both private and public networks, organizations should implement a range of security measures. These include strong encryption protocols, secure authentication mechanisms, regular software patching, employee education on safe internet practices, and

the use of VPN to establish secure connections. Additionally, network monitoring, intrusion detection systems (IDS), and firewalls can help identify and mitigate potential threats arising from public network usage.

By understanding the risks and taking proactive security measures, organizations can protect their sensitive data, systems, and resources when interacting with public networks, ensuring the integrity and confidentiality of their operations.

## **5 Security considerations for private and public networks**



**Figure 2:** Key aspects for security considerations for private and public networks.

Security considerations for both private and public networks are crucial to ensure confidentiality, integrity, and availability of data and resources. While the specific considerations may vary, [→Figure 2](#) shows keys aspects that should be taken into account for both network types:

- *Access control:* Implement strong access controls to authenticate and authorize users and devices. These include

use of secure passwords, multifactor authentication (MFA), and RBAC to limit access privileges based on user roles and responsibilities.

- *Encryption:* Use encryption mechanisms, such as secure sockets layer/transport layer security (SSL/TLS), IPsec, and VPNs, to protect sensitive data in transit over private and public networks. Encryption ensures that data remains secure even if intercepted.
- *Network segmentation:* Segment networks into smaller subnets or VLANs to separate different functional areas or security zones. This helps contain potential breaches and limits lateral movement within the network. Additionally, deploy firewalls and access control lists to enforce segmentation and control traffic flow between segments.
- *Security management:* Frequently apply security analysis and updates to network devices, operating systems, and software applications. Security or patch management minimizes the risk of recognized vulnerabilities being misused by attackers.
- *Logging and monitoring:* Enable comprehensive logging of network activities and implement a centralized monitoring system to detect and respond to security incidents promptly. Analyzing logs can provide valuable insights into potential security breaches and aid in forensic investigations.

- *Security awareness and training:* Educate employees about best security practices including recognizing and reporting phishing attempts, avoiding suspicious websites, and protecting sensitive information. Regular training programs enhance employee awareness and foster a security-conscious culture.
- *Physical security:* Consider physical security measures to protect network infrastructure including server rooms, data centers, and network devices. Implement restricted access controls, video surveillance, and environmental controls to safeguard physical assets.

By considering these security considerations, organizations can establish a robust security framework for both private and public networks. Implementing appropriate controls, protocols, and measures mitigates risks, protects against unauthorized access, and ensures the overall security of network environments.

## **5.1 Network segmentation**

Distribute a computer network into smaller sub networks known as network segmentation or VLANs to improve security, performance, and management. Each segment operates as a

separate network, isolated from other segments, with its own set of rules and controls.

### 5.1.1 Benefits of network segmentation

→[Figure 3](#) shows the benefits of network segmentation:



**Figure 3:** Benefits of network segmentation.

- *Enhanced security:* Network segmentation limits the impact of security breaches by containing them within specific segments. It helps in isolating sensitive data and critical systems from potential threats, reducing the attack surface and minimizing lateral movement by attackers.
- *Improved performance:* By dividing the network into smaller segments, network congestion is reduced. This improves overall network performance, as data traffic is localized to specific segments and does not impact other parts of the network.
- *Compliance and regulatory requirements:* Network segmentation can aid in meeting regulatory and compliance requirements. It allows organizations to apply specific security controls and access policies to sensitive data or

systems, ensuring compliance with industry-specific regulations.

- *Simplified network management*: Segmenting the network makes it easier to manage and troubleshoot issues. It simplifies network administration tasks and reduces complexity.

## 5.2 Segmentation techniques

- *VLANs*: VLANs are logical partitions within a physical network. VLANs separate devices into different broadcast domains, even if they are physically connected to the same network switch. VLANs enable isolation and provide control over network traffic by segregating devices based on logical groupings.
- *DMZ (demilitarized zone)*: A DMZ is a separate network segment that acts as a buffer zone between the internal private network and the untrusted public network, typically the internet. The DMZ hosts publicly accessible services such as web servers or email servers while keeping them isolated from the internal network.
- *Virtualization*: Virtualization technologies, such as virtual machines (VMs) or containers, can create virtual network segments. This enables multiple isolated virtual networks to coexist on the same physical infrastructure, allowing

organizations to achieve network segmentation within virtualized environments.

### **5.3 Use of firewalls and intrusion prevention systems (IPS)**

Firewalls and IPS play a key role in network segmentation:

- *Firewalls:* Firewalls are security devices that monitor and control incoming and outgoing network traffic based on predetermined security policies [[→9](#)]. They act as a barrier between different network segments, inspecting traffic and allowing or blocking specific communication based on defined rules.
- *IPS:* IPS monitors network traffic for suspicious activities or known attack patterns. It can detect and prevent malicious traffic from entering or spreading within network segments, enhancing the security of each segment.

Firewalls and IPS are commonly deployed at the boundaries between network segments to enforce security policies and provide an additional layer of protection against unauthorized access, malware, and other network-based threats.

By employing network segmentation techniques such as VLANs, DMZs, and virtualization, along with the use of firewalls and

IPS, organizations can enhance their network security, reduce the attack surface, and improve overall network performance and management.

## **5.4 Access control mechanisms**

Access control mechanisms play a critical role in ensuring the security of networks and resources. Here are some key access control mechanisms and their benefits:

### **5.4.1 Authentication and authorization**

- *Authentication:* Authentication verifies the identity of users or devices trying to access a network or resource. It classically involves the use of identifications such as usernames and passwords, biometric data, or digital certificates.  
Authentication ensures that only legalized persons or devices gain access.
- *Authorization:* Authorization determines the level of access approved to authenticated users or devices. It defines what resources or actions an authenticated entity is allowed to access or perform based on their assigned privileges or permissions.

### **5.4.2 Role-based access control (RBAC)**

RBAC is a widely used access control model that assigns permissions to users based on their roles within an organization [[→10](#)]. Each role has a set of associated permissions, and users are granted access based on their assigned role. RBAC simplifies access management by providing a structured approach to authorization, improving security and reducing administrative overhead.

### **5.4.3 Two-factor authentication (2FA) and multifactor authentication (MFA)**

2FA and MFA provide an additional layer of security by requiring users to provide multiple forms of authentication to access a network or resource. 2FA and MFA significantly enhance security by reducing the likelihood of unauthorized access in case one authentication factor is compromised.

## **5.5 Network access control (NAC) solutions**

NAC solutions focus on enforcing security policies and controlling access to a network [[→11](#)]. Here are some additional concepts related to NAC and intrusion prevention:

### **5.5.1 Intrusion detection versus intrusion prevention**

- *IDS*: IDS monitors network traffic and systems for signs of potential security breaches or malicious activities [[→12](#)]. It identifies and raises alerts when suspicious activity is detected but does not take immediate action to prevent or stop the attack.
- *IPS*: IPS, on the other hand, not only detects malicious activities but also actively takes measures to prevent them. It can automatically block or drop suspicious network traffic, modify firewall rules, or perform other actions to mitigate the attack in real time.

### **5.5.2 Network-based intrusion prevention systems (NIPS)**

NIPS are dedicated devices or software solutions deployed at strategic points within a network to monitor and prevent unauthorized access and malicious activities. They inspect network traffic in real time, comparing it against known attack signatures or behavior patterns to detect and block potential threats. NIPS can actively respond to detected threats by blocking or redirecting network traffic to mitigate attacks.

### **5.5.3 Host-based intrusion prevention systems (HIPS)**

HIPS are software installed on individual hosts or endpoints to monitor and prevent intrusions at the host level. It operates by

analyzing system logs, examining file integrity, monitoring network connections, and implementing intrusion prevention measures directly on the host. HIPS can detect and block suspicious activities on the host such as unauthorized system modifications or attempts to exploit vulnerabilities.

#### **5.5.4 Signature-based and anomaly-based detection**

- *Signature-based detection:* Signature-based detection relies on predefined signatures or patterns of known threats. It compares network traffic or system behavior against a database of known attack signatures to identify and block malicious activities. However, signature-based detection may struggle with detecting new or unknown threats that do not match existing signatures [[→13](#)].
- *Anomaly-based detection:* Anomaly-based detection identifies deviations from normal network or system behavior. It establishes a baseline of normal activities and alerts or takes action when deviations or anomalies are detected. Anomaly-based detection is effective in identifying previously unknown or zero-day attacks but can also generate false positives if the baseline is not properly established [[→14](#)].

#### **5.5.5 Continuous monitoring and incident response**

Continuous monitoring is crucial to identify and respond to security incidents promptly. It involves 24-h care of network traffic, system logs, and security events to detect anomalies or signs of potential intrusions. By continuously monitoring the network and systems, organizations can respond swiftly to security incidents, contain the threats, and minimize damage.

Incorporating intrusion prevention mechanisms, such as NIPS and HIPS, along with signature-based and anomaly-based detection, provides organizations with proactive measures to prevent and detect unauthorized access and malicious activities. Continuous monitoring and incident response capabilities enable organizations to respond swiftly to security incidents, reducing the impact and ensuring the ongoing security of their networks.

## **5.6 Physical security measures**

Physical security measures are essential for protecting the network boundary where private and public networks intersect. Here are some key concepts related to physical security:

### **5.6.1 Importance of physical security at the network boundary**

Physical security is the foundation for overall network security. It prevents unauthorized access, tampering, theft, and physical damage to network infrastructure and resources. Without robust physical security measures, other cybersecurity measures can be compromised.

### **5.6.2 Restricted access areas and visitor management**

Restricting access to critical areas ensures that only authorized personnel can enter. This can be achieved through measures such as secure access control systems, badges or access cards, biometric authentication, and physical barriers like locks or gates. Visitor management processes, including registration, identification verification, and temporary access authorization, help control access for visitors and contractors.

### **5.6.3 Surveillance systems and alarms**

Surveillance systems, including CCTV cameras, video monitoring, and intrusion detection alarms, provide real-time monitoring and recording of activities at the network boundary. They deter potential intruders, enable the identification of security incidents, and facilitate investigations.

### **5.6.4 Environmental controls and power protection**

Maintaining appropriate environmental conditions, such as temperature and humidity control, is vital for the proper functioning of network infrastructure and equipment. Power protection mechanisms, such as uninterruptible power supply systems and backup generators, ensure continuous power supply to critical network components, minimizing the risk of disruptions and data loss.

#### **5.6.5 Integration of physical and digital security measures**

Effective security requires the integration of physical and digital security measures. Physical security controls, such as access control systems, should be integrated with network authentication mechanisms to ensure that only authorized individuals can access the network. Similarly, security incidents detected through digital systems should trigger appropriate physical security responses such as alarms or lockdown procedures.

#### **5.6.6 Defense-in-depth approach**

The defense-in-depth approach involves implementing multiple layers of security measures to provide overlapping protection. This includes a combination of physical security controls (locks, barriers, surveillance) and cybersecurity controls (firewalls,

IPS, and encryption). A defense-in-depth strategy ensures that if one layer is breached, other layers are still in place to prevent or mitigate the impact of an attack.

By implementing robust physical security measures, organizations can strengthen the overall security of their network boundary. Restricted access areas, visitor management, surveillance systems, environmental controls, and power protection contribute to a secure physical environment. Integrating physical and digital security measures and adopting a defense-in-depth approach helps organizations create a layered security posture that enhances the protection of their network infrastructure and sensitive information.

#### **5.6.7 Examples of organizations successfully protecting their network boundaries**

Several organizations have successfully implemented measures to protect their network boundaries and mitigate potential threats. Here are a few examples:

- *Google:* Google is known for its robust security practices. The company implements a multilayered security approach, combining encryption, access controls, IDS, and continuous monitoring. Google's BeyondCorp model eliminates the

traditional perimeter-based security approach and focuses on securing individual devices and user identities, regardless of their location or network used.

- *Amazon Web Services (AWS):* AWS provides cloud services and has established a strong reputation for security. AWS employs various security measures including network firewalls, virtual private clouds, access controls, encryption, and identity and access management mechanisms. AWS also offers comprehensive security services such as AWS Shield for DDoS protection and AWS WAF (web application firewall) for web application security [[→15](#)].
- *Cisco systems:* Cisco, a leading network infrastructure and security solutions provider, protects its network boundaries using its own security products. Cisco's ASA Firewall and Next-Generation Firewall solutions help enforce access controls, monitor network traffic, and detect and prevent security threats. Cisco also emphasizes network segmentation, employing VLANs and DMZs to separate internal networks from external ones [[→16](#)].
- *Microsoft:* Microsoft employs a range of security measures to protect its network boundaries. This includes strong access controls, such as Azure Active Directory, for authentication and RBAC for authorization. Microsoft Azure, their cloud platform, incorporates multiple security features including

virtual networks, network security groups, and application gateways to enforce secure network boundaries [[→17](#)].

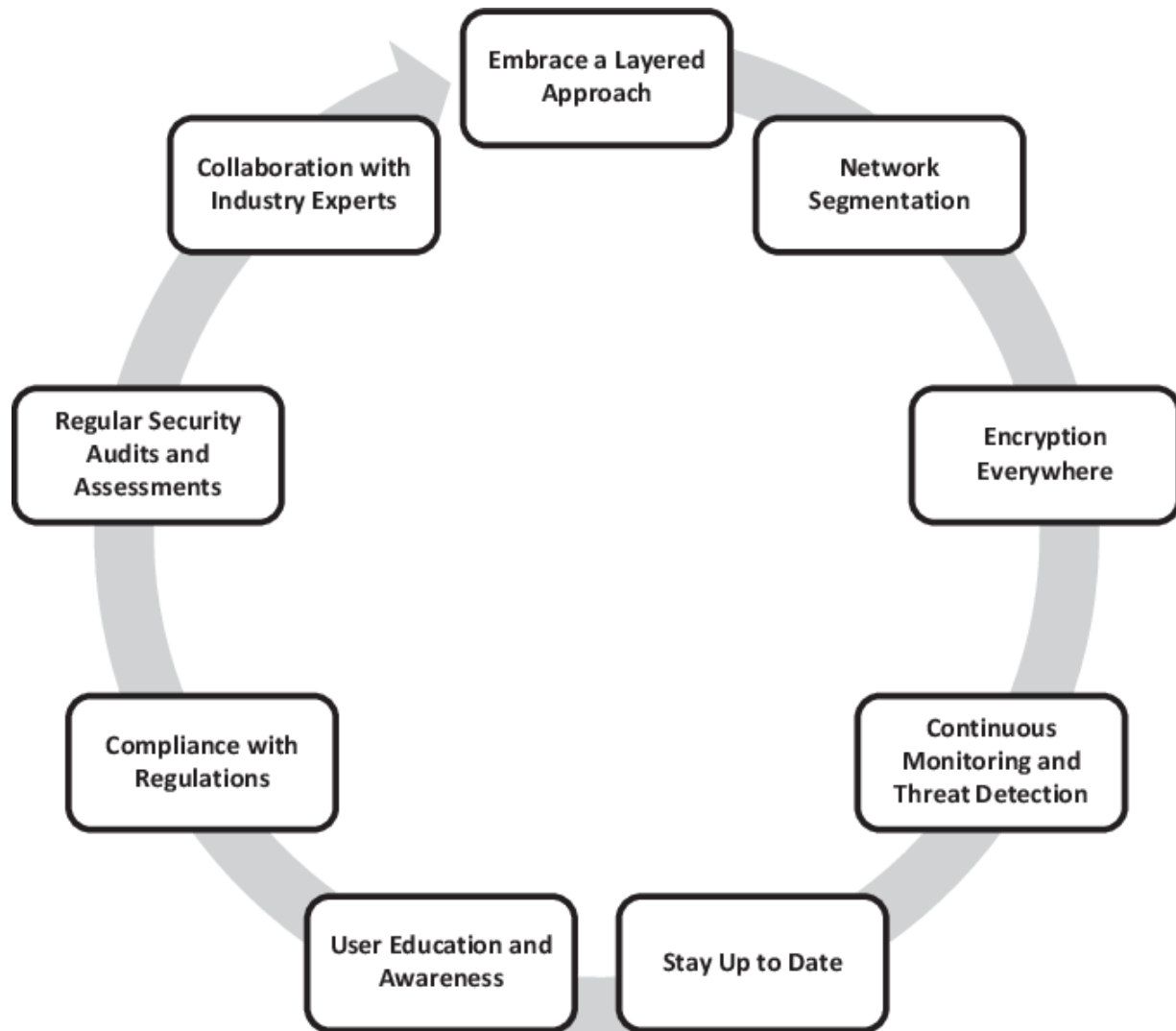
- *Financial institutions:* Banks and financial institutions place a high priority on network security due to the sensitive nature of their data. They employ various security measures including robust access controls, encryption for data in transit and at rest, secure communication channels (e.g., VPNs), intrusion detection and prevention systems, and rigorous compliance with industry regulations such as the Payment Card Industry Data Security Standard (PCI DSS) [[→18](#)].

It is important to note that each organization's security strategy should be tailored to their specific needs, risk profile, and regulatory requirements. These examples illustrate the adoption of best practices, including access controls, network segmentation, encryption, and intrusion detection/prevention, to protect network boundaries effectively.

#### **5.6.8 Recommendations and best practices for organizations to protect their physical boundaries**

The examples of organizations successfully protecting their network boundaries offer valuable lessons learned and best practices that can be applied to enhance network security. Some

key takeaways and best practices for organizations are shown in [→Figure 4](#):



**Figure 4:** Best practices for organizations.

- *Embrace a layered approach:* This ensures that security measures are implemented at various points and provides defense-in-depth against potential threats.

- *Implement strong access controls:* To control and manage user access to the network. Implementing strong access controls helps prevent unauthorized access and strengthens overall security.
- *Network segmentation:* Employ network segmentation to separate different parts of the network and create security zones. This helps contain potential breaches, limit lateral movement within the network, and restrict access to sensitive resources. Use technologies like VLANs, DMZs, and VPN to enforce segmentation.
- *Encryption everywhere:* Apply encryption to protect data from unlawful access, even if intercepted or accessed by malicious actors.
- *Continuous monitoring and threat detection:* Implement robust monitoring systems, identify anomalies, and detect possible security threats.
- *Stay up to date:* Frequently apply security analysis and updates to network devices, operating systems, and software applications. Keeping systems up to date helps to identify possible security threats.
- *User education and awareness:* Educating users about potential security risks and their role in maintaining network security is essential.

- *Compliance with regulations:* Stay compliant with relevant industry-specific regulations and data protection laws. Ensure network architectures and security measures align with compliance requirements, such as GDPR, HIPAA, or PCI DSS, depending on the industry.
- *Regular security audits and assessments:* Regular assessments help ensure that security controls remain effective and up to date.
- *Collaboration with industry experts:* Collaborate with industry experts, security vendors, and participate in security communities to stay informed about emerging threats, best practices, and evolving security technologies.

By incorporating these lessons learned and best practices into network security strategies, organizations can improve their ability to protect their network boundaries, mitigate potential risks, and safeguard their sensitive data and resources.

## **6 Future trends and considerations**

Protecting the physical boundary where the private and public networks encounter is an evolving challenge due to the dynamic nature of technology and emerging threats.

Organizations should consider the following future trends and

considerations to enhance the protection of their physical boundaries:

- *Zero trust architecture:* The adoption of zero trust principles is gaining momentum. Organizations are moving away from traditional perimeter-based security and adopting a zero-trust approach that assumes no trust within or outside the network. This approach emphasizes strict identity verification, granular access controls, continuous monitoring, and authentication for every user and device attempting to access resources at the boundary.
- *Software-defined perimeter (SDP):* SDP is an emerging concept that provides secure access to resources based on user identity and device posture, regardless of network location. SDP solutions dynamically create encrypted microsegments between users and resources, reducing the exposure of assets at the physical boundary and mitigating the risk of unauthorized access.
- *Cloud-based security services:* With the increasing adoption of cloud services, organizations are leveraging cloud-based security services to protect their physical boundaries. Cloud-delivered firewalls, web application firewalls, and secure web gateways provide scalable and flexible security controls that can be deployed at the intersection of private and public

networks, ensuring consistent protection across diverse environments.

- *Artificial intelligence and machine learning:* AI and ML technologies are being utilized to spot and reply to security threats in actual time. These technologies can analyze vast volumes of data, recognize patterns, and detect irregularities that may indicate possible security breaches or unauthorized access attempts at the physical boundary.
- *Internet of things (IoT) security:* As IoT devices become more prevalent in organizations, securing the physical boundary requires consideration of IoT security. Organizations should implement robust authentication mechanisms, monitor and manage IoT devices, and ensure that proper security controls are in place to prevent IoT devices from becoming entry points for attackers.
- *Enhanced physical security technologies:* Organizations should consider adopting advanced physical security technologies to protect the physical boundary. This includes biometric access controls, video analytics for real-time threat detection, and physical sensors that integrate with the overall security ecosystem to provide a holistic view of the security posture.
- *Threat intelligence and information sharing:* Collaboration and sharing of threat intelligence among organizations, industry groups, and security communities can provide valuable

insights into emerging threats and vulnerabilities at the intersection of private and public networks. Organizations should actively participate in information sharing initiatives and leverage threat intelligence platforms to enhance their security defenses.

- *Privacy and data protection regulations:* Organizations must stay updated with evolving privacy and data protection regulations. Organizations should implement privacy-enhancing technologies and adopt privacy-by-design principles.

## **7 Conclusion**

The protection of the physical boundary where private and public networks converge is a critical aspect of ensuring the security and integrity of organizational assets and information. This convergence introduces potential security risks and vulnerabilities that can be mitigated through the implementation of robust mechanisms and practices. Throughout this study, we have examined various mechanisms employed by organizations to safeguard their physical boundaries in the context of private-public network convergence. These mechanisms include the use of network security technologies such as firewalls, intrusion detection

systems, VPN, and network segmentation. Additionally, organizational policies and guidelines, as well as employee awareness and training programs, play significant roles in establishing a security-conscious culture and ensuring compliance with security protocols.

The findings of this study highlight the importance of carefully considering factors such as scalability, performance, compatibility, and cost-effectiveness when implementing mechanisms to protect the physical boundary. Organizations need to strike a balance between security requirements and operational efficiency to effectively address the challenges posed by private-public network convergence. In conclusion, organizations must recognize the significance of protecting the physical boundary where private and public networks converge and take proactive steps to implement comprehensive security mechanisms. The convergence of these networks should be viewed as an opportunity to enhance communication and collaboration while upholding the highest standards of security to safeguard organizational assets in an ever-evolving digital landscape.

## **References**

**[1]** A. Saklani and S. C. Dimri, "Technical Comparison between IPv4 & IPv6 and Migration from IPv4 to IPv6," International Journal of Science and Research, vol. 2, no. 7, pp. 231–235, 2013.

⇒

**[2]** D. C. Dobhal and S. C. Dimri, "Design and Implementation of TCP Friendly end-to-end Layered Solution for Mobile Ad hoc Networks (MANET)," International Journal of Applied Engineering Research, vol. 9, pp. 10249–10262, 2014. ⇒

**[3]** P. N. Goki, "Network Authentication, Identification, and Secure Communication through Optical Physical Unclonable Function," in European Conference on Optical Communication (ECOC), pp. 1–4, 2022. ⇒

**[4]** P. Singhal, "Survey on Security Issues in Mobile Cloud Computing and Preventive Measures," Asian Journal of Multidimensional Research, vol. 10, no. 10, pp. 1103–1109, 2021.

⇒

**[5]** A. Bahuguna, R. K. Bisht and J. Pande, "Country-level Cyber Security Posture Assessment: Study and Analysis of Practices," Information Security Journal: A Global Perspective, vol. 29, no. 5, pp. 250–266, 2020. ⇒

**[6]** A. Bahuguna, R. K. Bisht and J. Pande, "Assessing Cyber Security Maturity of Organizations: An Empirical Investigation in Indian Context," Information Security Journal: A Global Perspective, vol. 28, no. 6, pp. 164–177, 2019. ➔

**[7]** G. A. Marin, "Network security basics, Security & Privacy," IEEE, vol. 3, no. 6, pp. 68–72, 2005. ➔

**[8]** C. Systems, "Cisco Adaptive Security Virtual Appliance (ASAv) Getting Started Guide, 9.13," 2019. ➔

**[9]** J. Tyson, C. Pollette and S. Crawford, "How a VPN (Virtual Private Network) Works," [Online]. Available:  
➔<https://computer.howstuffworks.com/vpn.htm> [a](#), [b](#)

**[10]** Cisco Systems Inc, "Cisco Secure Firewall ASA Virtual Getting Started Guide, 9.18," 2022. [a](#), [b](#)

**[11]** Cisco Systems, Inc, "Cisco 850 Series and Cisco 870 Series Access Routers Software Configuration Guide," 2005. [a](#), [b](#)

**[12]** Cisco Systems Inc, "IP Addressing: NAT Configuration Guide, Cisco IOS Release 12.4T," 2011. [a](#), [b](#)

**[13]** C. Natalino, "Autonomous Security Management in Optical Networks," in Optical Fiber Communications Conference and

Exhibition (OFC), pp. 1–3, 2021. [a](#), [b](#)

**[14]** S. Shin, H. Wang and G. Gu, “A First Step Toward Network Security Virtualization: From Concept To Prototype,” IEEE Transactions on Information Forensics and Security, vol. 10, no. 10, pp. 2236–2249, Oct. 2015. doi: 10.1109/TIFS.2015.2453936. [a](#), [b](#)

**[15]** A. Lara and B. Ramamurthy, “OpenSec: Policy-Based Security Using Software-Defined Networking,” IEEE Transactions on Network and Service Management, vol. 13, no. 1, pp. 30–42, Mar. 2016. doi: 10.1109/TNSM.2016.2517407. [a](#), [b](#)

**[16]** L. Hu, H. Wen, B. Wu, J. Tang and F. Pan, “Adaptive Secure Transmission for Physical Layer Security in Cooperative Wireless Networks,” IEEE Communications Letters, vol. 21, no. 3, pp. 524–527, Mar. 2017. doi: 10.1109/LCOMM.2016.2633425. [a](#), [b](#)

**[17]** A. Bahuguna, R. K. Bisht and J. Pande, “Country-level Cybersecurity Posture Assessment: Study and Analysis of Practices,” Information Security Journal: A Global Perspective, vol. 29, no. 5, pp. 250–266, 2020. [a](#), [b](#)

**[18]** D. Anand, J. Pande and U. Maheshwari, “Identity-based Encryption Algorithm Using Hybrid Encryption and MAC Address for Key Generation,” International Journal of Innovative

Technology and Exploring Engineering, vol. 8, no. 12, pp. 2467–2474, 2019. [a](#), [b](#)

By combining binary search and insertion sort, a sorting method for small input size

**Bhawmesh Kumar**

**Aditya Bhatt**

**Neeraj Panwar**

**Abstract**

Insertion sort, one of the popular sorting methods having average time complexity  $O(n^2)$ , is based on the decrease and conquer strategy. We insert elements one by one in the sorted part and decrease the size of the unsorted portion, instead of a simple comparison method for insertion of elements in the sorted portion if we apply the binary search approach to locate the appropriate position for the inserting element in the sorted part. This will reduce the number of key comparisons; the only problem with this approach is to shift the elements one place to its right which are greater than  $A[i]$ . But the convergence of this approach toward the result is fast than the simple insertion sort. As far as time complexity is concerned both have almost similar performance, but when the cost of comparison is high this method is well suited.

**Keywords:** Binary search, insertion sort, time complexity, input size, decrease and conquer,

## **1 Introduction**

In computer science, sorting and searching are basic processes that are essential to a broad variety of applications. In this age of information, effective data organization and retrieval are crucial for maximizing efficiency and enhancing user experiences. To address these issues, a number of algorithms have been created, each with unique strengths and limitations. Here, we will examine the ideas behind binary insertion sort fusion of sorting and searching algorithms, concentrating on the simplicity of insertion sort, and the effectiveness of binary search.

Sorting algorithms are designed to arrange elements in a specific order, often in ascending or descending order. One of the simplest sorting algorithms is insertion sort, where the array splits into sorted and unsorted parts [[→1](#)]. In this sorting first take one element and iterate through the sorted array [[→2](#)]. Insertion sort removes one element from the given input array and later finds the right position as per the sorted array, finally inserted there. This step repeats itself till no element is left in the input array [[→3](#)]. It is much less efficient on large lists than more advanced algorithms such as merge sort, quick sort, and heap sort. The complexity of the worst and average case is  $O(n^2)$  whereas in the best case  $O(n)$ . As per space complexity, the worst case is  $O(n)$  total and  $O(1)$  auxiliary.

On the other hand, searching algorithms are used to locate a specific element within a collection of sorted data. Binary search is a widely used algorithm for searching in sorted arrays. Binary search is used to find out the location of an element in a sorted array [[→4](#)]. It is also

known as half interval or binary chop. It is applicable for sorted arrays to find the element location. Initially, it calculates the position of the middle of an array and then splits it into two arrays; it also considers the beginning and end position. Then check whether a searched element is less than mid element or greater than proceed with as per the condition. An element can be left side array or in a right side array or maybe at mid position. If at the mid position then the element is found otherwise it may be at the left and right array. Finally, with repeated steps, this search is completed with the element location which we need to find out. If the search ends with the remaining half being empty, the target is not in the array. This algorithm used logarithmic time in the worst case, then  $O(\log n)$  comparison. This search is faster than the linear search.

Numerous real-world applications, including database systems, e-commerce platforms, and search engines, employ sorting and searching algorithms. While e-commerce systems offer items based on factors like price, popularity, or relevancy, sorting algorithms increase query efficiency and allow effective indexing. Search engines are powered by search algorithms, which help consumers discover relevant information rapidly. Recognizing the potential benefits of combining sorting and searching algorithms, the concept of binary insertion sort [→5] emerged. Binary insertion sort harnesses the power of binary search to optimize the insertion process in insertion sort. It is faster in terms of working and stable filter algorithms. This algorithm is best for the lower size of an array.

When elements of an array are less, then binary insertion sort is best. The performance of algorithms is based on the time taken, how fast, memory space, and the kind of data structure used. Instead of blindly traversing the sorted part of the array to find the correct position for each element, binary insertion sort uses binary search to efficiently locate the insertion point. By doing so, it reduces the number of comparisons needed, leading to a more efficient sorting process. Other types of sorting algorithms are also available such as heap, quick, and merge sort. Various approaches and comparative studies have been done to sort the large and small arrays [[→6](#), [→7](#), [→8](#)]. These searching and sorting can also be helpful in the field of machine learning-based applications [[→9](#)].

Sorting and searching algorithms are essential tools in computer science. Insertion sort provides a basic understanding of how sorting works, while binary search showcases the power of decrease and conquer strategies. Binary insertion sort combines the strengths of both algorithms, optimizing the insertion process in insertion sort through the application of binary search. By understanding these algorithms and their characteristics, we can make informed decisions when it comes to selecting the most appropriate algorithm for our specific needs.

## **2 Related works**

Bender et al. [[→10](#)] showed the gapped insertion sort technique. Library sort is far better than traditional insertion sort because

library sort is based on a priority queue. This algorithm has a space one-third overhead than the library. Ahmad et al. [[→11](#)] suggested a modified efficient approach to sort an array to reduce the complexity. It showed the comparison with insertion, binary with insertion, and shell sort. Patel et al. [[→12](#)] proposed a novel approach to sort an array for the worst case and take more space. Where double space is taken to implement this approach if  $n$  is the number of elements, then  $2^n$  is taken as double-size array space. As a comparison point of view, the complexity of the Best Case for ascending order, the worst case for descending order, and the average case for random orders is shown through the graph.

Choudaiah et al. [[→13](#)] compared the performance of distinguishing types of sorting algorithms. Prepared groups of two types of sorting algorithms where one for  $n \cdot \log(n)$  and the second for  $n^2$ . Under the first group quick, shell, and heap are considered and for the group second bubble, insertion, and selection are considered. Show the comparison graphs for the time taken and number of swaps. Ray and Ghosh [[→14](#)] described the sorting technique based on binary gapped with support of binary insertion sort. It has been implemented on data elements arrived for being sorted in random access memory which is real-time sorting. It calculates the correct relative position and gapped absolute position. Goel and Kumar [[→15](#)] proposed a novel approach for sorting on the basis of Brownian Motus and clustered binary insertion sort. As comparison

shows 25%, 50%, 75%, and 100% are the levels of randomness in the initial dataset.

Shahbaz and Kumar [[→16](#)] proposed a sorting technique that measured execution time along with memory space required to perform sorting. It gives an improvement in large dataset consideration. This approach is compared with quick, merge, heap, insertion, selection, and shell. Buradagunta et al. [[→17](#)] suggested a study on various algorithms for positive and negative numbers. Algorithms considered for the comparative study are UNH, selection, insertion, bubble, merge, and quick sort. Quick sort takes less time, as per less input size than UNH sort. When increasing the size, then bubble takes more time. Iwama and Teruyama [[→18](#)] examined the average complexity of sorting algorithms. With the gap being reduced by 25% using binary insertion, which is still straightforward for a thorough mathematical analysis and performs well for several T length ranges.

Zutshi and Goswami [[→19](#)] aimed to propose a quadratic sorting algorithm to reduce the limitation of existing algorithms and worked with a disjoint set of unsorted elements. It is best for time complexity as comparison with existing algorithms. Schrier et al. [[→20](#)] determine the activity series using sorting algorithms. Relevant algorithms like binary insertion sort have been discussed, along with how they might be used to solve activity series problems. The number of experiments necessary to ascertain the relative reactivities is significantly reduced by using a suitable algorithm.

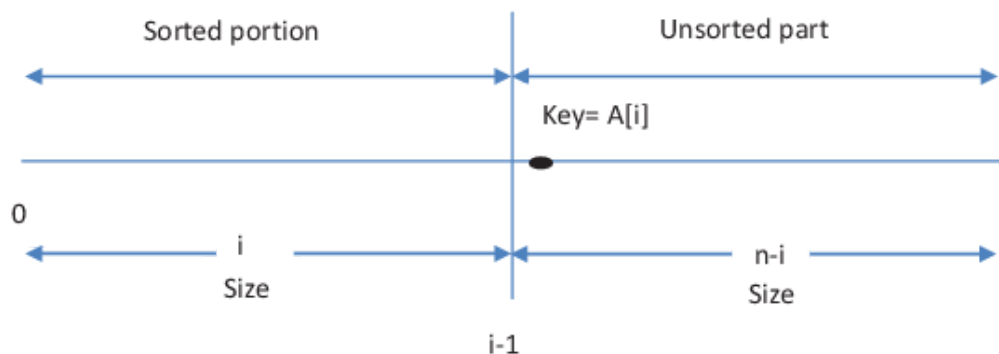
Rabiu et al. [[→21](#)] compare and review studies given on the different types of sorting algorithms using concurrency and shared data mechanisms. This is implemented in the Java development kit environment on the octa-core machine. As discussed quick sort developed using atomicity is the best for large size arrays. Find out the running time quick sort sequential and parallel with the support of acyclic barrier.

Ida [[→22](#)] proposed a system that can be used to book doctor's appointment. Here, they sorted the physicians according to the ratings provided by the patients using binary insertion sort. The physicians are arranged in decreasing order, starting with the one with the highest rating. Mell [[→23](#)] suggest using human expert opinion to gauge the relative security effect of software security features. A binary insertion sort is performed on a collection of security components by a knowledge encoding tool, and experts are requested to compare pairs of security elements to assess their relative relevance. To design and assess methods for handling huge datasets with constrained memory resources, further research is required. Certain input sizes or data distributions are best suited for modern sorting algorithms. An exciting research area is creating adaptive sorting algorithms that dynamically change strategies depending on incoming data attributes.

### **3 Proposed work**

Insertion sort is a sorting method based on the concept of decrease and conquer; stepwise we decrease the unsorted portion. Initially, the size of the sorted portion would be 1, an array (sub-array) containing a single element is a sorted array. Then we take the first element of the unsorted portion as key, that is,  $\text{key} = A[i]$ , for  $i = 1, 2, 3 - (n - 1)$  and then insert it at its appropriate position; in the sorted portion this will increase the size of the sorted portion by 1. In insertion sort to insert key at its appropriate position, there are lot many comparisons, which increase the complexity of the algorithm, rather than this if we use the strategy of binary search, this will be helpful to reduce the number of comparisons significantly.

For the sorted portion we compute the  $\text{mid} = \frac{0+i}{2}$  then compare ( $A[\text{mid}]$  and key) if key is smaller than  $A[\text{mid}]$ , shown in [→Figure 1](#), then use same binary search for  $A[0: \text{mid} - 1]$  else in  $A[\text{mid} + 1: i]$ , if ( $A[\text{mid}] = \text{key}$ ) then put key at the position next to  $A[\text{mid}]$ 's right this way we can insert key at its appropriate position.



**Figure 1:** The insertion sort.

And all the elements from the right where key is inserted to position  $i$  has shifted one position to its right.

### 3.1 Algorithm

Input: A [0: N-1] N element array, elements are in random order.

Output: A [0: N-1] elements are in ascending order

Binary\_Insertion\_Search (A)

N = length (A)

for i=1 to N-1 do

j=i, key=A[i]

low=0, high=i

while(low < high)

mid = (low+ high)/2

if (key < A[mid])

high = mid

else:

low = mid +1

end of if

end of while loop

while ( $j > \text{low}$ )

$A[j] = A[j-1]$

$j = j-1$

end of while loop

$A[\text{low}] = \text{key}$

end of for loop

end of function

### 3.2 Time complexity

The best case of binary insertion sort occurs when the input array is already sorted, thus  $\text{left} = i = j$ , the element remains in its initial position and so condition ( $j > \text{left}$ ) is false and so no right shifting of elements thus time complexity in this case.

To locate the appropriate position for the key element the time requirement is of order  $\theta(n \log_2 i)$  for  $i$ th iteration since there is no right shifting of element and only binary search works to locate the position and so the time complexity in this case:

(1)

$$\begin{aligned}
T_B(n) &= \sum_{i=1}^{n-1} \log_2 i = \log_2 1 + \log_2 2 + \log_2 3 + \cdots + \log_2 n - 1 \\
&= \log_2 1 + \log_2 2 + \log_2 3 + \cdots + \log_2 n - \log_2 n \\
&= \log_2 (1, 2, 3, \dots, n) - \log_2 n
\end{aligned}$$

$$T_B(n) = \log_2 \lfloor n - \log_2 n \rfloor$$

We know that  $\lfloor n - \log_2 n \rfloor = 1, 2, 3, \dots, (n-1)$

(2)

$$1 \leq n$$

$$2 \leq n$$

$$3 \leq n$$

$$n - 1 \leq n \Rightarrow 1, 2, 3, \dots, (n-1). n \leftrightarrow \leq n^n$$

$$n = n \Rightarrow \lfloor n \leq n^n$$

$$\text{And so } \log_2 \lfloor n \leq \log_2 n^n$$

$$\Rightarrow \log_2 \lfloor n \leq \log_2 n^n$$

From eqs. (1) and (2), we have

$$\begin{aligned}
T_B(n) &\leq n \log_2 n - \log_2 n \\
&= (n-1) \log_2 n \\
&\Rightarrow T_B(n) \in (n \log_2 n) \\
&\because (n-1) \in \theta(n) \\
&\log_2 n \in \theta(\log_2 n)
\end{aligned}$$

The best-case time complexity of binary insertion sort

$(n-1) \log_2 n \in \theta(n \log_2 n)$  is  $\theta(n \log_2 n)$ .

### 3.3 In the worst case of binary insertion sort

Elements are in decreasing order and thus condition ( $j > \text{low}$ ) will be valued for all  $i$  value; thus, the worst case of insertion sort occurs when the input elements are in nonincreasing order, and this is the condition ( $j > \text{left}$ ) would be valid for all possible value of  $i$  thus the worst-case time complexity will be given by

$$T_W(n) = \sum_{i=1}^{n-1} \{\log_2 i + i\}$$

$$i = 1$$

$\theta(\log_2 i)$  is time required to locate exact appropriate position for key element and ( $j > \text{low}$ ) will be valid for all  $i$  times thus

$$\begin{aligned} T_W(n) &= \sum_{i=1}^{n-1} \{\log_2 i + i\} \\ &= \sum_{i=1}^{n-1} \log_2 i + \sum_{i=1}^{n-1} i \\ &= \theta(n \log_2 n) + (1 + 2 + \cdots + (n - 1)) \\ &= \theta(n \log_2 n) + \frac{n-1}{2} (1 + 2 + \cdots + (n - 1)) \\ &= \theta(n \log_2 n) + \frac{n-1}{2} \cdot n \\ &= \theta(n \log_2 n) + \theta(n^2) \end{aligned}$$

And so,

$$T_W(n) \in \theta(n^2)$$

The worst-case time complexity of binary insertion sort is square.

### 3.4 For average case of binary insertion sort

It may be considered that the condition ( $j > \text{low}$ ) would be valid at most  $i/2$  times and thus, in this case time complexity would be

$$\begin{aligned}
 T_A(n) &= \sum_{i=1}^{n-1} (\log_2 i + i/2) \in \theta(n^2) \\
 &= \sum_{(i=1)}^{(n-1)} \log_2 i + \frac{1}{2} \sum_{(i=1)}^{(n-1)} i \\
 &= \theta(n \log_2 n) + \frac{1}{2} \cdot \frac{n(n-1)}{2} \\
 &= \theta(n \log_2 n) + \frac{1}{4} \cdot n(n-1) \\
 &= \theta(n \log_2 n) + \theta(n^2)
 \end{aligned}$$

And thus,

$$T_A(n) \in \theta(n^2)$$

Thus, it has been observed that the average case and worst case have some order of time complexity.

### 3.5 Time complexity analysis

It has been observed that the combination of binary search and insertion sort does not record much better time complexity. Through time complexity is some but binary insertion sort reduces the number of key comparisons to locate the position of the element which we want to insert in the sorted portion. The only problem with this new approach is the shifting of elements from one place to its right; otherwise this algorithm will be in this line of quicksort or merge sort. This shifting of elements detones the performance of the algorithm even though it is much better for low values of input  $n$ .

## 4 Numerical illustration

Consider the instance shown in [→Figure 2](#).

0	1	2	3	4	5	6
2	5	7	4	4	9	1

**Figure 2:** Array.

$A[i]=A[3]=4=key$

$j=3, i=3$

$low=0, high=3$

$(0<3)$

$mid=(0+3)/2=1$

$(key<A[mid])$  i.e  $(4<A[1]=5)$

$high=mid=1$

$mid= (low+high)/2 =0+1/2=0$

$key=4<A[mid]$  i.e  $(4<A[0]=2)$  false

So,  $low=0+1=1$

$(j=3>1)$

$A[3]=A[3-1]$  i.e.  $A[3]=A[2]=7$

$j=2$  as shown in [→Figure 3.](#)

0	1	2	3	4	5	6
2	4	5	7	4	9	1

**Figure 3:** Array after iteration.

$(j=2>1)$

$a[2]=a[2-1]$  i.e.  $a[2]=a[1]$

$j=1$

$(1>1)$  false

$a[\text{low}]=\text{key}$  i.e.  $a[1] = \text{key}=4,$

The array is shown in [→Figure 4.](#)

0	1	2	3	4	5	6
2	4	5	7	4	9	1

**Figure 4:** Array after iteration again.

This way this approach works fine and sorts the given array.

## 5 Conclusion

This chapter combines the binary search and insertion sorting methods and presents an efficient way to sort an array. Both binary search and insertion sort are based on decrease and conquer; thus this new method is also based on the same strategy. In insertion sort we insert the element one by one on the sorted portion; to locate the appropriate position it compares  $A[i]$  one by one with  $A[i - 1]$  and so on till the exact position does not confirm. Instead of this, if we use binary search to locate the position this drastically reduces the number of comparisons but a significant improvement in time complexity does not achieve since we need to shift those elements which are greater than  $A[i]$  to one place right. But this approach is fruitful for small-size arrays and when the cost of comparison is very high. Since the number of comparisons has been reduced this method is an improved version of insertion sort.

## References

**[1]** T. SinghSodhi, S. Kaur and S. Kaur, "Enhanced Insertion Sort Algorithm," International Journal of Computer Applications, vol. 64, no. 21, pp. 35–39, 2013. doi: 10.5120/10761-5724. ➡

**[2]** D. Jiang and M. Zhou, "A Comparative Study of Insertion Sorting Algorithm Verification," 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2017. doi: 10.1109/itnec.2017.8284998. ➡

**[3]** W. Min, "Analysis on 2-element Insertion Sort Algorithm," 2010 International Conference on Computer Design and Applications, 2010. doi: 10.1109/icdda.2010.5541165. ➡

**[4]** M. Ahmad, A. A. Ikram, I. Wahid and A. Salam, "Efficient Sort Using Modified Binary Search – A New Way to Sort," World Appl Sci J, vol. 28, no. 10, pp. 1375–1378, 2013. doi: 10.5829/idosi.wasj.2013.28.10.1715. ➡

**[5]** M. Hasan, S. Hossain, S. N. Datta and A. Yousuf, "Binary Insertion Sort: A Modified Way of Sorting," Asian Journal of Information Technology, vol. 5, no. 7, pp. 678–60, 2006. ➡

**[6]** I. S. Rajput, B. Kumar and T. Singh, "Performance Comparison of Sequential Quick Sort and Parallel Quick Sort Algorithms," International Journal of Computer Applications, vol. 57, no. 9, pp. 14–22, 2012. doi: 10.5120/9142-3363. ➡

**[7]** D. R. Aremu, O. O. Adesina, O. E. Makinde, O. Ajibola and O. O. Agbo-Ajala, "A Comparative Study of Sorting Algorithms," Afr J Comput Ict, vol. 5, no. 6, pp. 199–206, 2013. ➡

**[8]** B. Subbarayudu, L. Lalitha Gayatri, P. Sai Nidhi, P. Ramesh, R. Gangadhar Reddy and K. C. Kumar Reddy, "Comparative Analysis on Sorting and Searching Algorithms," International Journal of Civil Engineering and Technology (IJCIET), vol. 8, no. 8, pp. 955–978, 2017. ➡

**[9]** P. Rawat, P. Singh and V. Tripathi, "Load Balancing in Cloud Computing Leading Us Towards Green Cloud Computing," SSRN Electronic Journal, 2022. doi: 10.2139/ssrn.4031990. ➡

**[10]** M. A. Bender, M. Farach-Colton and M. A. Mosteiro, "Insertion Sort is  $O(n \log n)$ ," Theory of Computing Systems, vol. 39, no. 3, pp. 391–397, 2006. doi: 10.1007/s00224-005-1237-z. ➡

**[11]** M. Ahmad, A. A. Ikram, I. Wahid and A. Salam, "Efficient Sort Using Modified Binary Search-a New Way to Sort," World Appl Sci J, vol. 28, no. 10, pp. 1375–1378, 2013. doi: 10.5829/idosi.wasj.2013.28.10.1715. ➡

**[12]** S. Patel, M. Dennis Singh and C. Sharma, "Increasing Time Efficiency of Insertion Sort for the Worst-Case Scenario," Int J Comput Appl, pp. 975–8887, 2014,  
➡<http://www.princeton.edu/~achaney/tmve/wiki100k/> ➡

**[13]** S. Choudaiah, C. Chowdary and M. Kavitha, "Evaluation of Sorting Algorithms, Mathematical and Empirical Analysis of Sorting Algorithms," Int J Sci Eng Res, vol. 8, no. 5, 2017.

→<http://www.ijser.org>. →

**[14]** S. K. Ray and S. Ghosh, "Binarily Gapped Binary Insertion Sorting Technique," IETE J Res, vol. 64, no. 3, pp. 337–346, 2018. →

**[15]** S. Goel and R. Kumar, "Brownian Motus and Clustered Binary Insertion Sort Methods: An Efficient Progress over Traditional Methods," Future Generation Computer Systems, vol. 86, pp. 266–280, 2018. doi: 10.1016/j.future.2018.04.038. →

**[16]** M. Shabaz and A. Kumar, "SA Sorting: A Novel Sorting Technique for Large-scale Data," Journal of Computer Networks and Communications, vol. 2019, 2019, doi: 10.1155/2019/3027578. →

**[17]** S. Buradagunta, J. D. Bodapati, N. B. Mundukur and S. Salma, "Performance Comparison of Sorting Algorithms with Random Numbers as Inputs," Ingénierie Des Systèmes D Information, vol. 25, no. 1, pp. 113–117, 2020. doi: 10.18280/isi.250115. →

**[18]** K. Iwama and J. Teruyama, "Improved Average Complexity for Comparison-based Sorting," Theoretical Computer Science, vol. 807, pp. 201–219, 2020. doi: →<https://doi.org/10.1016/j.tcs.2019.06.032>. →

**[19]** A. Zutshi and D. Goswami, "Systematic Review and Exploration of New Avenues for Sorting Algorithm," International Journal of

Information Management Data Insights, vol. 1, no. 2, 2021. doi: 10.1016/j.jjime.2021.100042. ➡

[20] J. Schrier, M. F. Tynes and L. Cain, "Determining the Activity Series with the Fewest Experiments Using Sorting Algorithms," Journal of Chemical Education, vol. 98, no. 5, pp. 1653–1658, 2021. doi: 10.1021/acs.jchemed.1c00043. ➡

[21] A. M. Rabi, E. J. Garba, B. Y. Baha, Y. M. Malgw and M. Dauda, "Performance Comparison of Three Sorting Algorithms Using Shared Data and Concurrency Mechanisms in Java," Arid-zone Journal of Basic & Applied Research, vol. 1, no. 1, 2022. doi: 10.55639/607fox. ➡

[22] S. J. Ida. "Doctor Appointment Booking System Using Content Based Filtering Recommendation." ISSN: 0974-5823, vol. 7, no. 1, 2022. ➡

[23] P. Mell, "The Generation of Security Scoring Systems Leveraging Human Expert Opinion," arxiv, vol. 2, 2021, doi: ➡  
[➡https://doi.org/10.48550/arXiv.2105.13755](https://doi.org/10.48550/arXiv.2105.13755). ➡

---

**De Gruyter Series on the Applications of Mathematics in  
Engineering and Information Sciences**

**Already published in the series**

**Volume 16: Distributed Transfer Function Method. One-  
Dimensional Problems in Engineering**

Bingen Yang, Kyoungrae Noh

ISBN 978-3-11-075854-2, e-ISBN (PDF) 978-3-11-075893-1, e-ISBN (EPUB) 978-3-11-075900-6

### **Volume 15: Machine Learning for Cyber Security**

Malik Preeti, Nautiyal Lata, Ram Mangey (Eds.)

ISBN 978-3-11-076673-8, e-ISBN (PDF) 978-3-11-076674-5, e-ISBN (EPUB) 978-3-11-076676-9

### **Volume 14: Multiple Criteria Decision-Making Methods. Applications for Managerial Discretion**

Mohini Agarwal, Adarsh Anand, Deepti Aggrawal

ISBN 978-3-11-074356-2, e-ISBN (PDF) 978-3-11-074363-0, e-ISBN (EPUB) 978-3-11-074374-6

### **Volume 13: Integral Transforms and Applications**

Nita H. Shah, Monika K. Naik

ISBN 978-3-11-079282-9, e-ISBN (PDF) 978-3-11-079285-0, e-ISBN (EPUB) 978-3-11-079292-8

### **Volume 12: Noise Filtering for Big Data Analytics**

Souvik Bhattacharyya, Koushik Ghosh (Eds.)

ISBN 978-3-11-069709-4, e-ISBN (PDF) 978-3-11-069721-6, e-ISBN (EPUB) 978-3-11-069726-1

**Volume 11: Artificial Intelligence for Signal Processing and Wireless Communication**

Abhinav Sharma, Arpit Jain, Ashwini Kumar Arya, Mangey Ram (Eds.)

ISBN 978-3-11-073882-7, e-ISBN (PDF) 978-3-11-073465-2, e-ISBN (EPUB) 978-3-11-073472-0

**Volume 10: Meta-heuristic Optimization Techniques. Applications in Engineering**

Anuj Kumar, Sangeeta Pant, Mangey Ram, Om Yadav (Eds.)

ISBN 978-3-11-071617-7, e-ISBN (PDF) 978-3-11-071621-4, e-ISBN (EPUB) 978-3-11-071625-2

**Volume 9: Linear Integer Programming. Theory, Applications, Recent Developments**

Elias Munapo, Santosh Kumar

ISBN 978-3-11-070292-7, e-ISBN (PDF) 978-3-11-070302-3, e-ISBN (EPUB) 978-3-11-070311-5

**Volume 8: Mathematics for Reliability Engineering. Modern Concepts and Applications**

Mangey Ram, Liudong Xing (Eds.)

ISBN 978-3-11-072556-8, e-ISBN (PDF) 978-3-11-072563-6, e-ISBN (EPUB) 978-3-11-072559-9

**Volume 7: Mathematical Fluid Mechanics. Advances on Convection Instabilities and Incompressible Fluid Flow**

B. Mahanthesh (Ed.)

ISBN 978-3-11-069603-5, e-ISBN (PDF) 978-3-11-069608-0, e-ISBN (EPUB) 978-3-11-069612-7

[→www.degruyter.com](http://www.degruyter.com)

**Volume 6: Distributed Denial of Service Attacks. Concepts, Mathematical and Cryptographic Solutions**

Rajeev Singh, Mangey Ram (Eds.)

ISBN 978-3-11-061675-0, e-ISBN (PDF) 978-3-11-061975-1, e-ISBN (EPUB) 978-3-11-061985-0

**Volume 5: Systems Reliability Engineering. Modeling and Performance Improvement**

Amit Kumar, Mangey Ram (Eds.)

ISBN 978-3-11-060454-2, e-ISBN (PDF) 978-3-11-061737-5, e-ISBN (EPUB) 978-3-11-061754-2

### **Volume 4: Systems Performance Modeling**

Adarsh Anand, Mangey Ram (Eds.)

ISBN 978-3-11-060450-4, e-ISBN (PDF) 978-3-11-061905-8, e-ISBN (EPUB) 978-3-11-060763-5

### **Volume 3: Computational Intelligence. Theoretical Advances and Advanced Applications**

Dinesh C. S. Bisht, Mangey Ram (Eds.)

ISBN 978-3-11-065524-7, e-ISBN (PDF) 978-3-11-067135-3, e-ISBN (EPUB) 978-3-11-066833-9

### **Volume 2: Supply Chain Sustainability. Modeling and Innovative Research Frameworks**

Sachin Kumar Mangla, Mangey Ram (Eds.)

ISBN 978-3-11-062556-1, e-ISBN (PDF) 978-3-11-062859-3, e-ISBN (EPUB) 978-3-11-062568-4

### **Volume 1: Soft Computing. Techniques in Engineering Sciences**

Mangey Ram, Suraj B. Singh (Eds.)

ISBN 978-3-11-062560-8, e-ISBN (PDF) 978-3-11-062861-6, e-ISBN  
(EPUB) 978-3-11-062571-4

---

# Index

## A

active [1](#)  
active language [1](#)  
iteration [1](#)  
access matrix [1](#)  
advanced persistent threats [1](#)  
analyzer [1](#)  
anomaly-based detection [1](#)  
ant colony optimization [1](#)  
ending order [1](#)  
assessment [1](#)  
authentication [1](#)  
autonomous [1](#)

## B

baseline [1](#)  
directional [1](#)  
array [1](#)  
array search [1](#)  
blacklist [1](#)  
brackets [1](#)

/cott 1

## C

allenges 1

ematograph 1

her 1

ne 1

ud security 1

stering 1

nparison 1

nplexity 1, 2

nplex relationship 1

nputationally 1

nfidentiality 1

nfusion matrix 1

nstraint 1

nsumption 1

ntemporary 1

nventional 1

nvergence 1

nvolutional 1

neffective 1

nintermeasures 1

ntVectorizer 1

VID era 1

dit card 1

ical 1

ersecurity 1

erwarfare 1

## D

abreaches 1

ision tree 1

rease and conquer 1

ryption 1

amation 1

ense-in-depth 1

militarized zones 1

moralization 1

ital world 1

astrous 1

criminatory speech 1

harmony 1

semination 1

tributed denial-of-service 1

inant 1

rmancy mechanism 1

istically reduces 1

ne-customer 1

## E

esdrops 1

ommerce platforms 1

ctronic soldiers 1

bedded systems 1

otions 1

ryption 1

ryption 1

forcement 1

raction 1

## F

score 1

ricated 1

ebook 1

e-news 1

sible 1

ture extraction 1

walls 1

od 1

network 1

quency 1

ctionality 1

## G

ident 1

ivitational 1

edy 1

## H

ker 1

iltonian 1

althcare 1

uristic 1

st-based intrusion prevention systems 1

oid 1

erparameters 1

## I

onsonment 1

lian Penal Code (IPC) 1

egration 1

egrity 1

erpretability 1

usion prevention systems 1

ation 1

## K

uggle 1

nel 1

## L

kage 1

at 1

itimate 1

ical 1

arithmic 1

juistic 1

balancing 1

search 1

ging 1

istic regression 1

## M

licious 1

lware 1

mory resources 1

taheuristic 1

imize damage 1

igate 1

obile cloud computing 1

ability management 1

onte Carlo 1

ropheme 1

rphology 1

ltivariate 1

## N

atural language processing 1

work spoofing 1

ural network 1

ncreasing 1

## O

imizations 1

anizations and corporations 1

erfitting 1

## P

allel processing 1

ameter 1

sing 1

nalizing 1

formance metrics 1

mutation [1](#)

shing [1](#)

omology [1](#)

ysical boundary [1](#)

arization [1](#)

icies [1](#)

itical Comments, and Religious Comments [1](#)

ential benefits [1](#)

gnomatics [1](#)

cautionary [1](#)

ision-recall trade-off [1](#)

vacy [1](#)

rogrammable [1](#)

paganda [1](#)

tection [1](#)

ocols [1](#)

vocating [1](#)

## Q

adratic sorting [1](#)

ck sort [1](#)

## R

dom forest [1](#)

domized variable neighborhood 1

omware 1, 2

all 1

ister 1

ression 1

ative relevance 1, 2

ability 1

ant 1

stricted 1, 2

rieval 1

ringyas 1

iting 1

## S

eguard 1

ling 1

ure key generation 1

urity audits 1

urity issues 1

urity virtual appliance 1

nautics 1

isational 1

isor 1

erity 1

nature-based detection 1

hole 1

dging 1

in 1

ambase 1

inning 1

holders 1

ervised 1

port vector machine 1

spicious 1

tainable 1

ip 1

tax 1

thesized 1

tematic 1

## T

rorism 1

eat 1

eelapsed 1

ologies 1

ditiional 1

nsmission scheduling 1

versing 1

ger 1

jan 1

ling 1

stworthiness 1

tter 1

## U

authorized 1

predictable 1

ization 1

## V

tex 1

lence-provoking 1

ual private network 1

tages 1

nerability 1

## W

ight 1

rst 1

## Y

Tube 1

Z

o•trust architecture 1