International Journal of Information Technology and Web Engineering
Volume 17 • Issue 1

Social Network Analysis for Precise Friend Suggestion for Twitter by Associating Multiple Networks Using ML

Dharmendra Kumar Singh Singh, Dr. C. V. Raman University, India

Nithya N., Sona College of Technology, India

Rahunathan L., Kongu Engineering College, India

Preyal Sanghavi, R. B. Institute of Management Studies Gujarat Technological University, India

Ravirajsinh Sajubha Vaghela, R. B. Institute of Management Studies Gujarat Technological University, India

Poongodi Manoharan, Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar*

https://orcid.org/0000-0001-6801-6138

Mounir Hamdi, Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar

Godwin Brown Tunze, Mbeya University of Science and Technology, Tanzania

https://orcid.org/0000-0002-7991-9962

ABSTRACT

The main aim in this paper is to create a friend suggestion algorithm that can be used to recommend new friends to a user on Twitter when their existing friends and other details are given. The information gathered to make these predictions includes the user's friends, tags, tweets, language spoken, ID, etc. Based on these features, the authors trained their models using supervised learning methods. The machine learning-based approach used for this purpose is the k-nearest neighbor approach. This approach is by and large used to decrease the dimensionality of the information alongside its feature space. K-nearest neighbor classifier is normally utilized in arrangement-based situations to recognize and distinguish between a few parameters. By using this, the features of the central user's non-friends were compared. The friends and communities of a user are likely to be very different from any other user. Due to this, the authors select a single user and compare the results obtained for that user to suggest friends.

KEYWORDS

Friend Suggestion, Tags, Tweet Analysis, Twitter

INTRODUCTION

Twitter is a popular microblogging service that is about 11 years old and is gathering more and more users. On Twitter, users can tweet about any topic they want within the character limit of 140 characters. Users also follow other users to receive the tweets they post online. Kwak et al. (2010) While studying the properties of Twitter's network, it is found that most users use Twitter to discuss

DOI: 10.4018/IJITWE.304050 *Corresponding Author

their daily activities or opinions on current topics. It helps users with similar intentions in a community to connect. Java et al. (2007).

Our main Objective in this paper is to create a friend suggestion algorithm that can be used to recommend new friends to a user on Twitter when their existing friends and other details are given. The information gathered to make these predictions includes the user's friends, tags, tweets, language spoken, id, etc. Based on these features, trained our models using supervised learning methods. The machine learning-based approach used for this purpose is the K-Nearest neighbors approach. This approach is used to decrease the dimensionality of the information alongside its feature space. K-nearest neighbor classifier is normally utilized in arrangement-based situations to recognize and distinguish between a few parameters. By using this, the features of the central user's non-friends were compared. The friends and communities of a user are likely to be very different from any other user. Due to this, in this research contribution it select a single user and compare the results obtained for that user to suggest friends. It will be called this user the "central user". Thus, in this work, it effectively train our model as per the central user itself. Daniyalzade and Lipus (2007).

LITERATURE SURVEY

The author talks about different networks and finds the relationship to recommend friends. It has two major components first being related networks by selecting important features and the second being network structure and preserving most of it. It is based on friend correlation and considers effect in different social roles. Huang et al. (2016).

The author gives idea about a new friend recommendation system using artificial bee colony(ABC) which indicates a link between users. It is based on the structural properties of the social network. Firstly, it finds the relevant parameters for the relationship among users using social topology. The sub- graph of the network is composed of users and all users within the network separated by three degrees of separation, then based on the subgraph new links are suggested thus indicating new friends. Akbari et al. (2013).

The paper gives a idea about a more precise friend recommendation with 2 stages. In the first stage the information of the relationship between text and users, then align the recommended friends. In the second stage, they built the topic model of the relationship between image features and users. Huang et al. (2017).

The paper proposed a novel semantic-based recommendation of friends based on their lifestyles. They take advantage of smartphones, friend books to discover the lifestyle of a user from user-centric data and then measure the similarity of lifestyle to finally recommend from with similar lifestyles. The friend book finally keeps a list of people with the highest score to recommend a friend. Wang et al. (2015).

The paper gives idea about users who want to meet friends on social media, they interviewed active users and then developed a friend request acceptance model to refer to various factors that influence it. They found out the major factor that impacts the person who accepts the friend request, mostly person with common hobbies and mutual friend is accepted. Rashtian et al. (2014).

The paper talks about how to find short paths between users in a network which would indicate their closeness and will also result in them being a good recommendation as a friend. These are based on email contact, it is found that the kind of things people talk or find is a huge factor in determining their closeness. Adamic and Adar (2005).

The paper "Ranking Users for Intelligent Message Addressing" uses various Machine Learning algorithms for its task. It also works on the simple k nearest neighbors concept to overtake other algorithms. It finds people who can receive a message from the central user, the person who is looking for friends. It understudies various people to conclude that intelligent messages can be added advantage to email Carvalho and Cohen (2008). The paper "Inferring relevant social networks from interpersonal communication" analyzes the network to look for the region of interest. It sees large social networks to look for unobserved ties (the tie in which i and j are connected) in the event (where i email j). The paper had two conclusions first being prediction task choosing threshold value yields better results and secondly, optimal threshold value seems to be consistent.

PROPOSED WORK

In this proposed work, used the Twitter API to collect the user data in the following way. Data collection steps taken can be listed as follows: choosing the central user, find their set of current friends, finding the set of users that are not friends with the current user (non-friends), and finally, gathering the profile information for every friend and non-friends concerning the central user. Now coming to all the major parts of the work in detail.

Dataset

Twitter enables us to mine the information of any client by utilizing Twitter API or Tweepy libraries in Python. The information extracted will be the tweets done by the users in the given time, along with their other details. The primary activity here involves getting the buyer key. These keys will help the API for confirmation. Data collection was hampered by the API's restrictions and privacy settings. To begin with, some profiles only allow limited access, making it impossible to view any information other than the user's name in certain cases. As a consequence, in this work, limited ourselves to selecting profiles inside particular networks that supplied us with all of the information, the author need, making it impossible to ask for all of a user's friends directly. The dataset can only be used to identify whether or not two people are friends. As a consequence, in this work, needed a large number of user ids to begin with, and it was impossible to find all of a person's friends unless had a large enough starting set. The lack of a straightforward mechanism for creating a list of users based on whatever arbitrary criteria which selected was the issue to be considered.

Choosing Central User

Since suggesting friends for each user would require querying the dataset for every user again and again. This implies that the algorithm would have a very high time complexity to be able to run in real-time. Due to this issue, in this work, fix a central user for which all the quantitative analysis is done once, and the corresponding friends are suggested. This central user can be changed every time the algorithm is tested. Once the algorithm is tested for the central user, it can further be extended to work for all the users independently in a parallel computing platform.

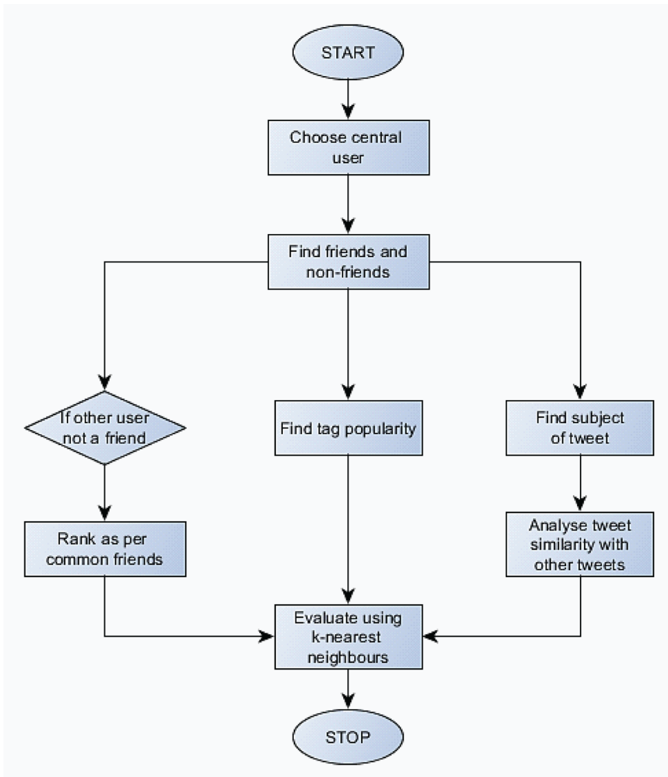
Comparison of Parameters Among Non-Friends

Once the dataset was queried and here in this paper, chose the central user to start with, the parameters obtained in the dataset were compared to decide on the possible factors that can affect the friend suggestion algorithm. While the number of followers can be a factor in determining the possible friends of a user, it is already incorporated by Twitter so other factors like the tag of the tweet, tag popularity, and common friends were explored further. Also, the subject of the user's tweet was obtained by using natural language processing and then compared with tags of other users' tweets to further enhance the suggestions.

Generating Friends List

Figure 1 shows the flow diagram for Friend Suggestion in Twitter. For this purpose, the non-friends of users are ranked as per the number of friends they have in common with the central user. Apart from this, the overall count of the tag of a tweet posted by the central user is compared with all other tags to evaluate the popularity of the user's tweet's tag. The tweet posted by the user is then analyzed for its subject and this subject obtained from the text is then compared with tags of other user's tweets

Figure 1. Flow Diagram of Friend Suggestion Algorithm



to find if the context of the central user's tweets matches with the tweets of other non-friends. The algorithm for calculating the tag popularity of the central user's tag is discussed below.

Algorithm 1: Algorithm for calculating tag popularity tags[] = list of tags of all users curr_tag = tag used by current user tp = tag popularity

```
i = 0
for tag in tags:
if (tag == curr_tag) then i = i + 1 End if end for
tp = i / length(tags)
```

In algorithm 1, the dataset is read and the tags of all the users are stored in an array independently. Once the central user is chosen arbitrarily, their tag becomes the current tag. For calculating tag popularity, the count of the tags of other user's tweets that are the same as the current tag is evaluated concerning the length of the array in which tags are stored. So:

$$tp = \begin{cases} 0, \ if\left(tag != curr\_tag\right) \\ \dfrac{i}{length\left(tags\right)}, otherwise \end{cases} \tag{1}$$

where:

- tp: tag popularity
- tag: tag of central user curr_tag: tag of the current user
- i: count of tags same as tag of central user tags: array of tags of all users

The algorithm for finding common friends among the non-friends of the central user is described as follows.

Algorithm 2: Algorithm for finding common friends friends[]= list of friends of all users curr_friends[] = list of friends of current user users[] = list of ids of all users

```
curr_user = id of current user
for i in users:
if (i == curr_user) then users[i] = -2
else if (i in curr_friends) then users[i] = -1
else users[i] = 0 end if End for
for i in curr_friends:
if (users[i] >= 0) then
if (i in friends(curr_friends)) then users[i] += 1 End if end if
end for
```

The other parameters included are the language spoken by the user, tweet similarity, and common friends. The formulas used for this purpose are:

$$sim = \begin{cases} 0, & if \;\; sub != curr\_sub \\ 1, & otherwise \end{cases} \tag{2}$$

- sim: similarity of the subject of tweets
- sub: subject of a tweet of central user
- curr_sub: subject of current tweet

$$k = \sqrt{tp^2 + l^2 + cf^2 + sim^2} \tag{3}$$

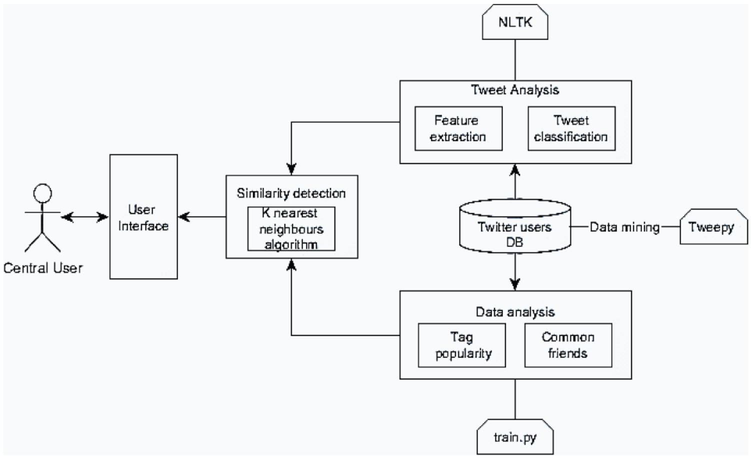
- k: distance of a non-friend from central user
- l: language spoken by the current user
- cf: number of common friends

Here, the non-friends with more similarity among corresponding attributes will have a higher value of k, as per the selected criteria.

SYSTEM ARCHITECTURE

The objective of this section is to discuss a novel system architecture for friend suggestions in a Twitter- like framework. This is done by working with the proposed methodology discussed in the previous section. The corresponding system architecture is illustrated in Figure 2.

Figure 2. System architecture for friend suggestion on Twitter



While designing the system, in the first phase, that is, the data mining phase, the dataset is extracted from Twitter. In the analysis and computation phase, one part deals with the data analysis while another module deals with the tweet analysis part of the system. The component of the system that analyzes the connection among users is included in the data analysis part. As shown in the figure, the system architecture has four major independent modules, which are described in detail.

Data Extraction Module

In this module, libraries such as Tweepy are used to extract user data from Twitter. Due to security reasons, only a few required details for each user are extracted. Also, the user-selected belong to a common social group so that their friends are also present in the dataset. Once data is extracted, the friends of the users that are not a part of the dataset are excluded during the preprocessing stage. The output of this component is the processed CSV file containing data for each user.

Data Analysis Module

Given the tags of tweets available for each of the users, this module does important computations for detecting similarities among various non-friends of the central user. This module further has two parts, that is, calculating tag popularity and finding common friends.

In this phase, the mathematical computations take place and all the quantitative results are obtained. Other smaller attributes that are considered for calculations are the language spoken by the current user, their number of followers, etc. These attributes are already a part of the Twitter friend suggestion algorithm. However, they also play a role in detecting similarities among two users.

Tweet Analysis Module

This component deals with extracting the subject of the tweet from the content of the tweet. This subject is then compared with the tags of all the other tweets to analyze the similarity between them. This is done by using chunking, then feature extraction is done on the obtained chunks of sentences in the tweet. Once this is done, the features and the probability of that feature being the subject of the tweet are compared to finally arrive at the resultant subject.

Similarity Detection Module

Once all the results from the data analysis component and the tweet analysis component of the system are obtained, these results are then further combined to arrive at the final similarity metric for the non- friend. This is done by calculating the distances between the data for the given central user and the current user.

IMPLEMENTATION

Description of Dataset

The dataset used is directly extracted from Twitter using the Tweepy library in Python. It consists of a network of 4000 users. The attributes included in the dataset are username, user id, user avatar, tweet, tweet id, tweet tag, language, number of followers, and id of all the friends of the user.

The dataset description is discussed in Table 1. The table shows the overall analysis of the data extracted from Twitter. Figure 3 illustrates the graphical distribution of various languages used by all the users in their tweets. The Figure shows that most of the users in the dataset communicate in English.

EXPERIMENTAL SETUP

The execution of the code required the installation of Python 3 on the system. Along with this, feature extraction and subject determination of tweet require NLTK module in Python, which enables natural language processing in Python. The central user is randomly selected and given as input while running the code.

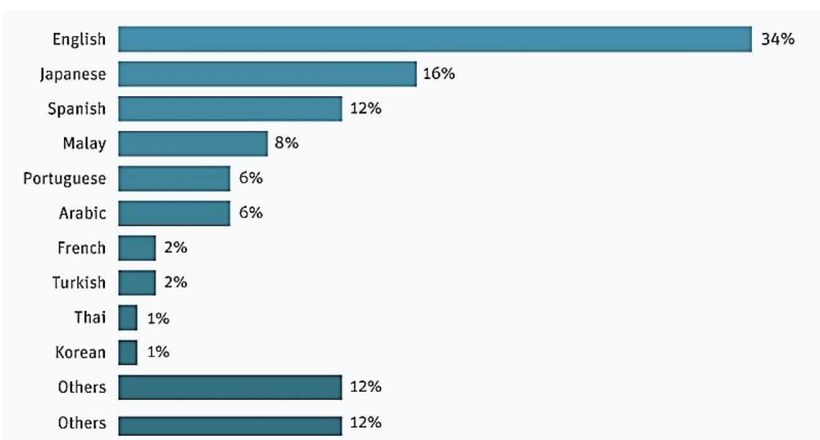
RESULTS AND ANALYSIS

The dataset used here has a network of 4000 users, where all the friends of the user are in the network itself and even if a user has friends that are not a part of the dataset, they are excluded from the

Table 1. Dataset

Description Number of users	4000
Number of distinct tags	483
Avg. number of followers per user	~390 Avg. number of friends per user

Figure 3. Language distribution in the dataset



analysis part. The algorithm suggests the top 10 friend suggestions that a user can relate with as per the quantitative results obtained from the factors studied. However, if there is a clash in a ranking of several users, that is, if multiple non-friends have the same final score, then the users are displayed in alphabetical order. This means that there is a chance that a user who may be a possible friend of the central user is excluded from the suggestions because of the clash in the rank.

Figure 4 and Figure 5 show the top 10 friend suggestions for two different users having the same tags. Here, it can show that have similar friend suggestions as they posted tweets with the same tags.

Figure 4. Friend recommendations for the username “cooperjellybean”

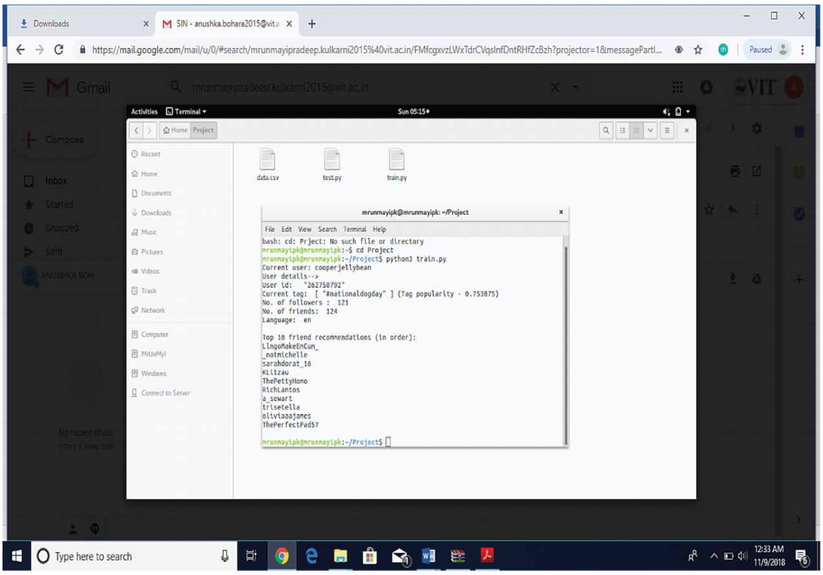
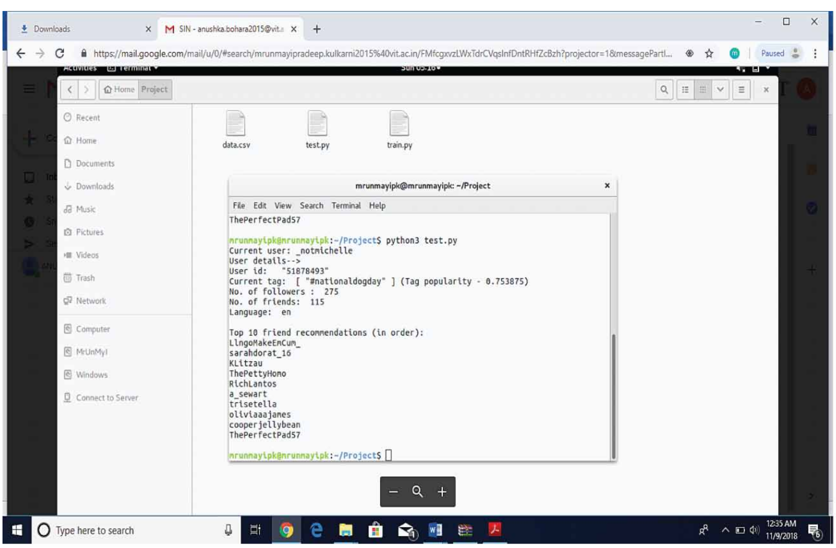


Figure 5. Friend recommendations for the user “_notmichelle”



Although the results that have been discussed above were obtained by examining the dataset ex- tracked from Twitter, the approach and the algorithms described in the paper can be applied to any social networking based framework, such as Facebook or even e-mail based framework, such as Gmail (where it can be used to suggest email groups).

CONCLUSION

The idea behind our friend suggestion method is that people have a greater affinity for their friends than for strangers. As a result, non-friends who have a common friend have greater clout than non-friends who are disconnected from the core user. Having a high number of common friends does not necessarily mean the user's interests are aligned with the central user's. Non-friends must be ranked according to a set of criteria that takes into consideration the users' common interests. This would help to separate those with different interests from those who have similar ones, boosting the algorithm's efficiency. In this research work, proposed a friend recommendation algorithm based on Tweet similarity and Tag popularity, in addition to the existing criterion of mutual friends. Non-friends may be ranked based on these factors by our recommendation engine. Using natural language processing, provided techniques for assessing tag popularity, finding common friends, and measuring tweet similarity. Our technique may improve Twitter's existing friend suggestion algorithm by offering a more efficient and complete approach to friend recommendation, according to the study of the small network dataset gathered. In simple words, instead of recommending friends based on common friends, our algorithm takes into account extra platform-relevant factors.

Finally, in this research contribution, have tried to improve the problem of identifying a user's offline (that is, real-life) social community, purely from examining the Twitter network structure of the central user. Based on observations from our Twitter data and results from previous works, it propose an algorithm involving three factors to improve the Twitter friend suggestion algorithm. Incorporating these factors, developed a novel algorithm to iteratively discover the user's possible friends based on a new way of measuring user closeness.

FUTURE WORKS

The current algorithm being used in the friend suggestion is the K-nearest neighbors algorithm. While this algorithm helps derive useful interpretations from the dataset, comparing quantitative results from other algorithms can help improve the suggestions further. Other training and classification algorithms such as Convolutional Neural Networks and Support Vector Machines can be used and differences in accuracy rate can be analyzed and compared among the various algorithms to arrive at a model that gives the best results for friend recommendation in Twitter. Other factors such as the popularity of user's tweets can also be analyzed to make interpretations and use them to make suggestions better. There can also be amendments that can be made to the algorithm to make it faster and more time- efficient.

Another very meaningful similarity metric that can be used to derive results is cosine similarity-based comparison. It measures the similarity between two non-zero vectors by measuring the cosine of the angle between them. The lesser the angle, the more similar/related are the corresponding users.

REFERENCES

Adamic, L., & Adar, E. (2005). How to search a social network. *Social Networks*, *27*(3), 187–203. doi:10.1016/j.socnet.2005.01.007

Akbari, F., Tajfar, A. H., & Nejad, A. F. (2013). *Graph-based friend recommendation in social networks using artificial bee colony. In Dependable, Autonomic and Secure Computing*. DASC.

Baccour, E., Foufou, S., Hamila, R., & Hamdi, M. (2015). A survey of wireless data center networks. *49th Annual Conference on Information Sciences and Systems (CISS)*, 1–6. doi:10.1109/CISS.2015.7086853

Bourouis, S., Band, S. S., Mosavi, A., Agrawal, S., & Hamdi, M. (2022). Meta-Heuristic Algorithm-Tuned Neural Network for Breast Cancer Diagnosis Using Ultrasound Images. *Frontiers in Oncology*, *12*, 834028.

Carvalho, V. R., & Cohen, W. W. (2008). Ranking users for intelligent message addressing. In *European Conference on Information Retrieval*, (pp. 321–333). Springer.

Daniyalzade, E., & Lipus, T. (2007). Facebook friend suggestion. *CS, 229*.

Dhanaraj, R. K., Ramakrishnan, V., Poongodi, M., Krishnasamy, L., Hamdi, M., Kotecha, K., & Vijayakumar, V. (2021). Random Forest Bagging and X-Means Clustered Antipattern Detection from SQL Query Log for Accessing Secure Mobile Data. *Wireless Communications and Mobile Computing*, 2021.

Dhiman, P., Kukreja, V., Manoharan, P., Kaur, A., Kamruzzaman, M. M., Dhaou, I. B., & Iwendi, C. (2022). A Novel Deep Learning Model for Detection of Severity Level of the Disease in Citrus Fruits. *Electronics (Basel)*, *11*(3), 495.

Hamdi, M., Bourouis, S., Rastislav, K., & Mohmed, F. (2022). Evaluation of Neuro Image for the Diagnosis of Alzhimer's Disease Using Deep Learning Neural Network. *Frontiers in Public Health*, 35.

Huang, S., Zhang, J., Wang, L., & Hua, X. S. (2016). Social friend recommendation based on multiple network correlation. *IEEE Transactions on Multimedia*, *18*(2), 287–299. doi:10.1109/TMM.2015.2510333

Java, A., Song, X., Finin, T., & Tseng, B. (2007). Why we twitter: understanding microblogging usage and communities. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, 56–65. doi:10.1145/1348549.1348556

Lilhore, U. K., Poongodi, M., Kaur, A., Simaiya, S., Algarni, A. D., Elmannai, H., & Hamdi, M. et al. (2022). Hybrid Model for Detection of Cervical Cancer Using Causal Analysis and Machine Learning Techniques. *Computational and Mathematical Methods in Medicine*, 2022.

Maurya, S., Joseph, S., Asokan, A., Algethami, A. A., Hamdi, M., & Rauf, H. T. (2021). Federated transfer learning for authentication and privacy preservation using novel supportive twin delayed DDPG (S-TD3) algorithm for IIoT. *Sensors (Basel)*, *21*(23), 7793.

Muniyappan, A., Sundarappan, B., Manoharan, P., Hamdi, M., Raahemifar, K., Bourouis, S., & Varadarajan, V. (2022). Stability and numerical solutions of second wave mathematical modeling on covid-19 and omicron outbreak strategy of pandemic: Analytical and error analysis of approximate series solutions by using hpm. Mathematics, 10(3), 343. doi:10.1145/1772690.1772751

Poongodi, M., Hamdi, M., Gao, J., & Rauf, H. T. (2021, December). A Novel Security Mechanism of 6G for IMD using Authentication and Key Agreement Scheme. In 2021 IEEE Globecom Workshops (GC Wkshps) (pp. 1-6). IEEE.

Poongodi, M., Hamdi, M., Sharma, A., Ma, M., & Singh, P. K. (2019). DDoS detection mechanism using trust-based evaluation system in VANET. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 183532–183544.

Poongodi, M., Hamdi, M., Varadarajan, V., Rawal, B. S., & Maode, M. (2020, July). Building an authentic and ethical keyword search by applying decentralised (Blockchain) verification. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 746-753). IEEE.

Poongodi, M., Hamdi, M., & Wang, H. (2022). Image and audio caps: Automated captioning of background sounds and images using deep learning. *Multimedia Systems*, 1–9.

Poongodi, M., Malviya, M., Hamdi, M., Rauf, H. T., Kadry, S., & Thinnukool, O. (2021). The recent technologies to curb the second-wave of COVID-19 pandemic. *IEEE Access : Practical Innovations, Open Solutions*, *9*, 97906–97928.

Poongodi, M., Nguyen, T. N., Hamdi, M., & Cengiz, K. (2021). Global cryptocurrency trend prediction using social media. *Information Processing & Management*, *58*(6), 102708.

Poongodi, M., Sharma, A., Hamdi, M., Maode, M., & Chilamkurti, N. (2021). Smart healthcare in smart cities: Wireless patient monitoring system using IoT. *The Journal of Supercomputing*, *77*(11), 12230–12255.

Poongodi, M., Vijayakumar, V., Al-Turjman, F., Hamdi, M., & Ma, M. (2019). Intrusion prevention system for DDoS attack on VANET with reCAPTCHA controller using information based metrics. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 158481–158491.

Poongodi, M., Bourouis, S., Ahmed, A. N., Vijayaragavan, M., Venkatesan, K. G. S., Alhakami, W., & Hamdi, M. (2022). A Novel Secured Multi-Access Edge Computing based VANET with Neuro fuzzy systems based Blockchain Framework. Computer Communications. doi:10.1145/1772690.1772722

Poongodi, M., Malviya, M., Hamdi, M., Vijayakumar, V., Mohammed, M. A., Rauf, H. T., & Al-Dhlan, K. A. (2022). 5G based Blockchain network for authentic and ethical keyword search engine. IET Commun., 16(5), 442-448.

Poongodi, M., Malviya, M., Kumar, C., Hamdi, M., Vijayakumar, V., Nebhen, J., & Alyamani, H. (2022). New York City taxi trip duration prediction using MLP and XGBoost. International Journal of System Assurance Engineering and Management, 13(1), 16-27. doi:10.1109/TMM.2017.2652074

Ramesh, T. R., Vijayaragavan, M., Poongodi, M., Hamdi, M., Wang, H., & Bourouis, S. (2022). *Peer-to-peer trust management in intelligent transportation system: An Aumann's agreement theorem based approach*. ICT Express.

Ramesh, T. R., Lilhore, U. K., Poongodi, M., Simaiya, S., Kaur, A., & Hamdi, M. (2022). Predictive analysis of heart diseases with machine learning approaches. Malaysian Journal of Computer Science, 132-148. doi:10.1016/S0167-8191(96)00054-3

Rashtian, H., Boshmaf, Y., Jaferian, P., & Beznosov, K. (2014). To befriend or not? a model of friend request acceptance on facebook. *Symposium on Usable Privacy and Security (SOUPS)*, 285–300.

Rathore, M. S., Poongodi, M., Saurabh, P., Lilhore, U. K., Bourouis, S., Alhakami, W., ... & Hamdi, M. (2022). A novel trust-based security and privacy model for Internet of Vehicles using encryption and steganography. Computers and Electrical Engineering, 102, 108205. doi:10.1109/JSAC.2003.810499

Rawal, B. S., Manogaran, G., & Hamdi, M. (2021). Multi-Tier Stack of Block Chain with Proxy Re-Encryption Method Scheme on the Internet of Things Platform. *ACM Transactions on Internet Technology*, *22*(2), 1–20.

Rawal, B. S., Manogaran, G., Singh, R., Poongodi, M., & Hamdi, M. (2021, June). Network augmentation by dynamically splitting the switching function in SDN. In 2021 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.

Rawal, B. S., Manogaran, R. G., Singh, M., & Poongodi, M. (2021). Network Augmentation by Dynamically Splitting the Switching Function in SDN. *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 1–6. doi:10.1109/ICCWorkshops50388.2021.9473709

Sahoo, S. K., Mudligiriyappa, N., Algethami, A. A., Manoharan, P., Hamdi, M., & Raahemifar, K. (2022). Intelligent Trust-Based Utility and Reusability Model: Enhanced Security Using Unmanned Aerial Vehicles on Sensor Nodes. Applied Sciences, 12(3), 1317.

Zedini, E., Oubei, H. M., Kammoun, A., Hamdi, M., Ooi, B. S., & Alouini, M. (2017). A New Simple Model for Underwater Wireless Optical Channels in the Presence of Air Bubbles. *GLOBECOM 2017- 2017 IEEE Global Communications Conference*, 1–6.