Tweet Recommender Model using Adaptive Neuro-Fuzzy Inference System

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF DEGREE

OF

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By:

VIBHUTI SHARMA

2K17/CSE/19

Under the supervision of

Dr. AKSHI KUMAR

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

JUNE, 2019

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

DECLARATION

I, Vibhuti Sharma, Roll No. 2K17/CSE/19 student of M.Tech (Compter Science & Engineering), hereby declare that the Project Dissertation titled "**Tweet Recommender Model using Adaptive Neuro-Fuzzy Inference System**" which is submitted by me to the Department of Computer Science & Engineering , Delhi Technological University, Delhi Report of the Major II which is being submitted to Delhi Technological University, Delhi, in partial fulfillment for the requirement of the award of degree of Master of Technology for the requirements of the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: DTU, Delhi Vibhuti Sharma

Date: 25-05-2019 (2K17/CSE/19)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College Of engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**Tweet Recommender Model using Adaptive Neuro-Fuzzy Inference System**" which is submitted by Vibhuti Sharma, Roll No. 2K17/CSE/19, Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfilment for the requirement of the award of degree of Master of Technology (Computer Science and Engineering) is a record of a project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date: 25-05-2019

(Dr. Akshi Kumar)

SUPERVISOR

Assistant Professor

Department of Computer Engineering

Delhi Technological University

ABSTRACT

Twitter is a ubiquitous, socially engaging and rapid communication medium. To filter the relevant information (news/hashtags/links/follow/tweet) for better user experience recommender systems have been extensively used on Twitter. Uncertainty in user preference, fuzziness in the rating process and the imprecision associated with the voluminous and varied twitter data are some of the difficulties associated which impede enhanced recommendations. This research put forwards an adaptive neuro-fuzzy inference system based tweet recommender model to handle the uncertainty, impreciseness and vagueness in item features and user's behaviour. The proposed hybrid (content-based and collaborative filtering based) model learns the interests of users (source tweet user and target tweet user) to categorize tweets. The users are characterized as source user (the user who posted the original tweet) and target user (to whom the tweet is to be recommended). The interests of the source and target user are extracted and the correlation between user interests is established which along with the category of the target tweet are then used to build the neuro-fuzzy model. The results show that the proposed model predicts the recommendation score correctly most of the time with the root mean square error of 0.93. -parameters.

ACKNOWLEDGEMENT

I am most thankful to my family for constantly encouraging me and giving me unconditional support while pursuing this research.

I am extremely grateful to **Dr. Akshi Kumar** Asst. Professor, Department of Computer Science Engineering, Delhi Technological University, Delhi for providing invaluable guidance and being a constant source of inspiration throughout my research. I will always be indebted to her for the extensive support and encouragement she provided.

I also convey my heartfelt gratitude to all the research scholars of the web Research Group at Delhi Technological University, for their valuable suggestions and helpful discussions throughout the course of this research work.

Vibhuti Sharma

Roll No. 2K17/CSE/19

TABLE OF CONTENTS

List of figures

List of Tables

LIST OF ACRONYMS

P	Precision
R	Recall
BEP	Precision-Recall breakeven point
ML	Machine Learning
SVM	Support Vector Machine
URL	Uniform Resource Locator
MF	Membership Function
Trimf	Triangular MF
Trapmf	Trapezoidal MF
Bellmf	Generalized bell MF
Gaussmf	Gaussian curve MF
Gauss2mf	Gaussian combination MF
Pimf	π-shaped MF
Dsigmf	Difference between two sigmoidal MF
Psigmf	Product of two sigmoidal MF

Chapter 1

Introduction

The mounting global interest of users in social media portals has reinforced research in analytics and sensing-based domains to discover knowledge from the publicly available user-generated big data [1-3]. Recommender systems are one of the most proverbial and easily understandable applications of big data. As a specialized information filtering system, a recommender system tries to make predictions on the basis of user preferences and interests [4, 5]. Their use has been pervasive with interesting use-cases within variety of application domains that range from recommending products, movies, music, books, research articles, search queries, social tags, experts, persons, jokes, restaurants, financial services and even twitter followers.

1.1 Overview

Concurrently, twitter has been the top choice communication and socially engaging media owing to its escalating global presence and accessibility [6]. Its ease of use, socializing and activism paradigm, easy and fast follower generation and quick condensed news, further stimulates the popularity. A typical tweet configuration on twitter includes various components. The following fig.1.1 depicts the structural elements associated with a generic tweet.

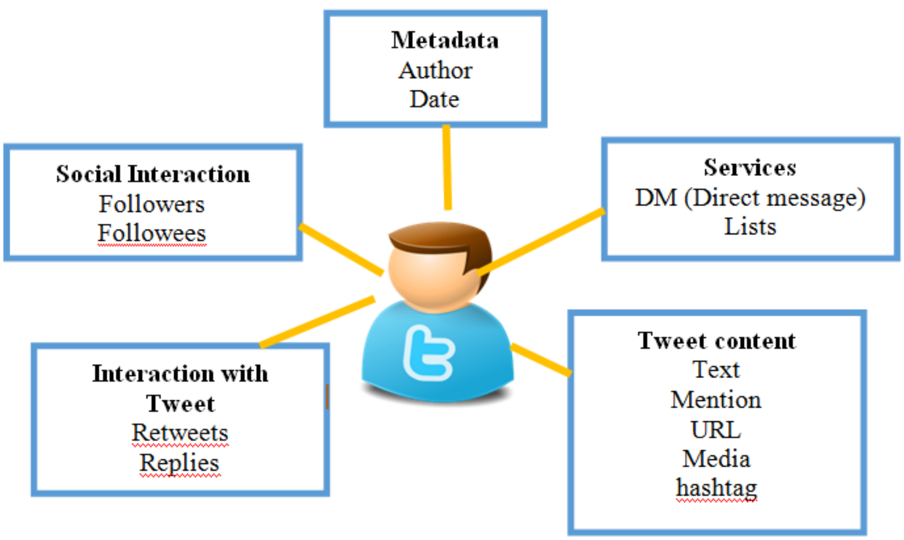

Fig.1.1. Structural elements of Tweet

As per the twitter usage statistics available[1] every second, on an average, around 6,000 tweets are tweeted on Twitter, which corresponds to over 350,000 tweets sent per minute, 500 million tweets per day. Apparently, in 2019, the social network's audience size is projected to reach 275 million monthly active users worldwide[2]. As a prominent source of user-generated big data, twitter data is representative of all the characteristic V's, namely, velocity, volume, value, variety, and veracity in data. Having this access to information is significant, and so more recently the point of focus has shifted from the problem of ‘information overload’ to the hitches of ‘filter failure’. Thus, it has now become crucial to build clever, intelligent and semantic filters which can guide users, personalize preferences and give interest-based options. Recommendation on twitter can be used to suggest social interactions (followers, followees), tweet content (mentions, URLs, tweets, hashtags) and retweets.

When the user is following many active users, there are chances that the user might miss out reading some interesting tweets [7]. Motivated by the need to build superior filters which can handle the massive amount of information available, this research proffers an adaptive neuro-fuzzy inference system based tweet recommender model.

As compared to conventional logic which has absolute truth values characterized as true or false, fuzzy logic demonstrates degrees of truth. For example, while sensing cold weather, conventional logic will link values to either no (0) or yes (1) whereas fuzzy logic will define a scale that is, very much (0.9), little (0.25) or negligible (0.1). Fuzzy logic is capable of representations similar to generalized human cognitive abilities for handling problems with imprecise, incomplete data and modeling nonlinear functions of arbitrary complexity. It is readily customizable to model the natural language semantics capturing the intricacy and vagueness associated with linguistic use. However, it lacks a delineated method that can transform these representations into a rule based fuzzy inference system (FIS). Adjustment of membership functions (MFs) further add to the time complexity [8].

Recently, deep learning neural architectures have been popularly used in natural language processing tasks owing to their hierarchical learning and generalization

[1]internetlivestats.com
[2]*https://www.statista.com*

capabilities. More specifically, artificial neural networks (ANN) work on the basis of the structure and functions of a human brain and can be employed for automatically adjusting the membership functions and diminish the rate of errors in the rule determination of fuzzy logic[9]. The benefits that this unification of neural network and fuzzy logic provides encouraged us to use an adaptive neuro-fuzzy inference system (ANFIS)to model a tweet recommender system. The ANFIS combines the learning capabilities of neural networks with the abilities of fuzzy logic to model uncertainty in expressiveness. It incorporates the benefits of adaptive control technique, artificial neural network, and the fuzzy inference system.

1.2 Research Objective

The primary objectives of this thesis are:

- To develop an adaptive neuro-fuzzy inference system (ANFIS) based tweet recommender model which provides an efficient and personalized tweet recommendation to users.
- To build a hybrid recommender model(content filtering and collaborative filtering based), where the content history of users is analyzed to uncover their interests implicitly using supervised learning, and the information from similar interest users is used to predict affinity to a given tweet.

1.3 Proposed Model

The problem statement can be defined as, "*What are the chances (recommendation score) that a tweet,'t' that has been tweeted/retweeted by a user, 'y', will be of interest to a user, 'x'?"* To solve this problem, the proposed system firstly learns users' interests by analyzing tweets, retweets and likes. An input tweet is classified into six pre-defined categories, namely, politics, business, technology, entertainment, health and sports. The users are characterized as source user (the user who posted the original tweet) and target user (to whom the tweet is to be recommended). The users' interests are implicitly derived by selecting the top three categories. Next the inference rule base is built to find the correlation between the category of target tweet, interest of target user and the interest of source user. Thus, it is the recommendation score calculated using this fuzzy inference rule base which tells about the probability that a tweet, t tweeted/retweeted by a user, y, belonging to category t_c will be of user x's interest. The following fig.1.2 abstracts the model graphically.

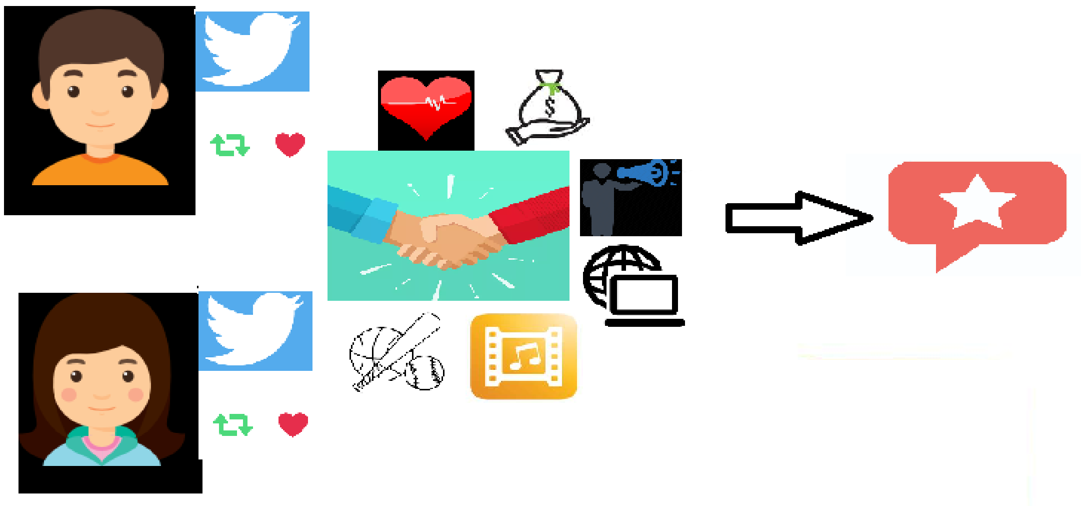

Fig.1.2. Graphical abstract of the proposed model

1.4 Organization of Thesis

The thesis is reported as follows: Chapter 2 presents a review of related literature work and a brief description of the concepts used in the proposed model. This helps in building a better understanding of the proposed model. Chapter 3 presents the architecture and working details of proposed ANFIS based tweet recommender model followed by its empirical analysis in chapter 4. Finally, the conclusion and future scope for the research is presented in chapter 5.

Chapter 2

Literature Review

Twitter has become a key social media for rapidly disseminating news and opinions. Many primary as well as secondary studies have been done within the domain of Twitter data related recommendation. These broadly include: news recommendation, followee recommendation, tweet recommendation, link recommendation and hashtag recommendation, amongst others.

2.1 Related Work

This section presents a summary of the work done in the field of recommender models especially on Twitter. Broadly, the recommender models on Twitter can be classified in the following categories:

a) News recommendation
b) Followee recommendation
c) URL recommendation
d) Hashtags recommendation
e) Tweet recommendation

News recommendation

Many studies have reported news recommendation models. In [7] authors suggest that user based attributes are the features which indicate the social relation that exists among the users. It helps in predicting similarities among users. In [10], authors studied and observed how user's profiles change with time. Also, they analyzed the methods for news recommendation to the users based on their interests. A hybrid recommendation model based on the similarity between a user's profile and news articles has also been developed [11]. Here, authors ranked the news articles on the basis of similarity to user profile. In [12], authors use tweets to develop user profiles and recommend the news articles which they are most likely to find interesting. Supervised learning method has been used in this study. Spectrum entity extraction system has been used in the paper [13]. Entity concept has been used for finding the similarity among tweets and news articles.

Followee recommendation

Much work has also been done in the field of followee recommendation. Armentano et al. proposed various topology-based models for recommending followees [14-16]. In [17], authors listed features which could be useful for followee recommendation. Location, popularity, tweet content, common friends and activity were said to be relevant but they evaluated only two of them i.e., activity and popularity. Basically, this study suggested that if a user has many active and popular followees then while recommending followees to him, activity and popularity will play the most important role. Golder et al. introduced a structural approach for followee recommendation [18]. This approach is based on filtered people, shared interests and reciprocity in order to recommend followees. The basic assumption behind reciprocity is that many a times users follow back his/her followers just to reciprocate. Homophily is the assumption behind shared audience and shared interests. Homophily means people tend to make relations with like-minded people. Based on the followers and followees, the similarity among users is computed.

URL recommendation

Another important aspect of recommender systems on Twitter is URL recommendation. Work based on URL recommendation has been done [19-21]. In [19] authors demonstrated the importance of social graph information for recommendation. They also used the tweet content. Galuba*et al.* proposed an architecture which predicts the users who are most likely to tweet about a URL on the basis of past user behavior [22]. URL has proved to be a key unit of information in Twitter. The most common methods for URL ranking in order to recommend to a user include 1) similarity among tweets which contain the current URL, and 2) the voting peer of the user who tweeted a tweet containing the current URL.

Hashtag recommendation

Due to the growing use of hashtags and trends, recommender systems based on hashtags are becoming popular among researchers nowadays. The authors in [23] proposed a recommendation system based on tag correlation. Some of the researches focused on the growth and decay of the trending subjects [22, 24]. Asur*et al.* analized the trends on Twitter and came up with the factors affecting the setup, continuance and decay of trends [22]. Significant amount of work has also been done in the field of hashtag recommendation. ‘#’ followed by a word is known as a hashtag. Chances are high that this word tells the label of a tweet. A hashtag is capable of spreading a tweet all over the world within seconds. [25] Although, it is not necessary that users are interested in all the information which appears on their timeline. Hashtags can contribute in filtering out the relevant information [1]. In [26], authors introduced an algorithm, 5WTAG for determining topic hashtags in microblogs. This algorithm is also capable of clustering the hashtags and measure the similarity between pair of hashtags. Jun et al. proposed a LSTM based method which incorporated the temporal information into the sentence-level attention model to enhance hashtag recommendation [27]. Hashtag recommendation has been modeled as a multi-class classification problem by deep neural network [28] and attention mechanism has also been used in some works for the same[29-32] to enhance the performance. In [33], authors introduced a semantic based method for finding similarities among short texts.

Tweet recommendation

On Twitter, it is the content that a tweet carries which attracts users. Therefore, the recommender systems based on tweets can prove to be the most efficient ones. Many studies have come reported for the same. Chen et al. [34] proposed an architecture based on collaborative ranking model to recommend tweets to users. They used content of tweet and user relations to detect the user interests. Due to the presence of humungous amount of content present on Twitter, it becomes extremely important to extract the relevant information for better user experience. Here, relevant information refers to the tweets in which a user is most likely to be interested in. A limited work has been done on recommender systems to extract and recommend relevant tweets to users. In [35], authors proposed an unsupervised learning method for extracting information out of voluminous text collection. The authors in [36] proposed a dynamic personalized recommender system for recommending tweets to users. In this work, authors considered the trends that are popular locally and using this information they developed a recommender system.

2.2 Background Concepts

In this section a discussion on the basic concepts, namely, the taxonomy of twitter, recommender systems, fuzzy inference system, artificial neural networks and ANFIS is presented. The related work follows the background concepts.

2.2.1 Taxonomy of Twitter

Twitter is an online social networking service on which people post short texts known as tweets. It is all about rapid communication and is aptly termed as microblog. The tweets are by default public but a user can restrict the visibility to his/her followers. The twitter taxonomy is presented in table 2.1.

Table 2.1: Taxonomy of Twitter

Element	Description
Tweet	A 280 character twitter post by a user with a choice to select from 40 languages. The tweet can be retweeted and/or liked and/or commented on.
Retweet RT	Re-posting of a tweet to share it with all of your followers. It is sharing the same exact tweet with your followers as it was published by the other person/business handle. Retweets generally intend to spread content virally.
Quote tweet	Sharing someone's tweet with your comment/mention/thought about the tweet, here you still have 280 characters to share besides the tweet content of the other person/business.
Reply/Comment	Replying to a tweet by mentioning the person who’s tweeted it and not reposting. Subsequently, a threaded conversation is the observed phenomenon.

#hashtag	# is used to reference a quote/topic/subject from any discussion thread where the hashtag acts as a directory to the quoted text. Basically, hashtags organize content around keywords.
Username/ Twitter Handle	Twitter handle or username are your identity on Twitter. These terms are used interchangeably. It's unique to the account and comes after the "@" sign and creates a link to your profile.
Follower	Follower is someone who has clicked the 'Follow' button your profile. It simply refers to the act of subscribing to the feed of another account. Followers can: • Read tweets from the accounts they follow. • Interact with tweets – like, comment and share. • If the account allows, followers can send direct messages too.
Following	These are people you choose to follow. Every time a user tweets, their content will show up in your personal feed.
Mention @	@ is used when you want to mention or tag a specific user and the person usually gets notified for the same.
Like	Like button is a way of giving someone thumbs up. It is represented using a red heart icon.
Media	A media object represents a photo, video or animated GIF. Twitter currently supports 4 photos, 1 animated GIF or 1 video in a Tweet.
URL/Link	Every Tweet has its own URL that you can bookmark or share with friends.
Direct Message	It facilitates a private communication system on Twitter. These private messages don't show up on anyone's Twitter home page or public feed.
Feed	Ongoing stream of tweets in reverse chronological order listed on your homepage.
Twitter Moment	Moments are curated stories stitching together multiple tweets showcasing the very best of what's happening on Twitter.

Timestamp	The date and time of posting

2.2.2 Recommender Systems

Recommender systems provide personalization of information and facilitates decision making. Improved user experience and increased conversions (content consumed, product bought, engagement, etc.) characterize the application of recommender systems which are services that track and analyze individual persons' (e.g. website or app users) behavior to make inferences and predictions for better decisions. The following figure 2.1 depicts the types of recommender systems described across pertinent literature.

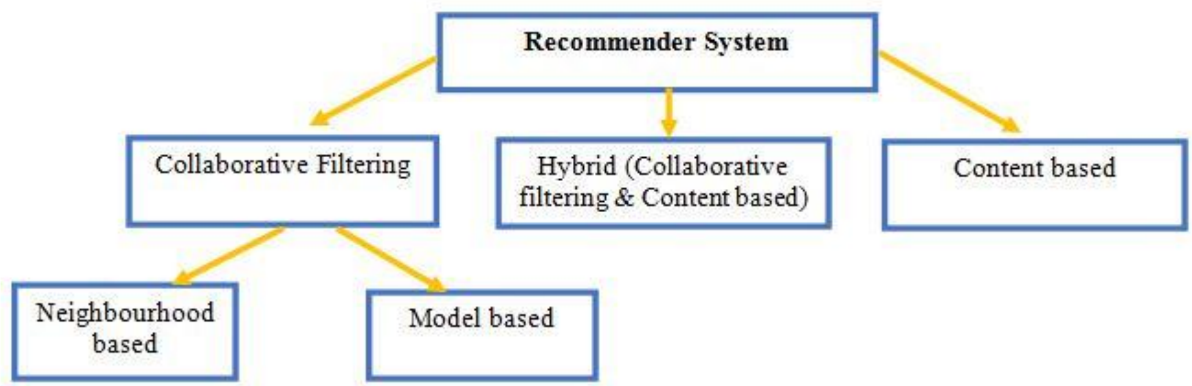

Fig.2.1. Types of Recommender systems

The following Table 2.2 briefly describes these recommender system (RS) types.

Table 2.2: Prominent Recommender system types

Type of RS	Definition	Pros	Cons
Content based filtering	It depends on similarity among items for prediction. Recommendations are based on the basis of features of the items that a user has used in the past [37][4]. It uses various kinds of	• Able to recommend new items (no existing rating) • Quickly	• Require in-depth description of items • Only those items are

	models for finding similarity among items for making recommendations such as Vector Space model (eg. TF/IDF) or probabilistic models (eg. Naïve Bayes classifier, neural networks). Further, these techniques learn the model witheither statistical analysis or ML techniques. Examples: News Dude [38], LIBRA [39]	adjust recommendations if user preference changes	recommended which are similar to the ones which are in their profiles
Collaborative filtering	It relies upon how different users responded to same products and not on the object itself. It matches similar users for making recommendations Examples: [40].GroupLens [43], Ringo [44].	• A good performer in fields where there is not much information associated with items • It can recommend relevant content which might not be similar to the one present in user's profile	• Data sparsity [37][42] • Cold start [41] • Scalability[42]
Neighbourhood based	It is a type of collaborative filtering based RS. It searches	Same as collaborative	Same as collaborative

	for a similar neighbor of the user on the basis of already rated items by him [45]. Based on such neighbours, recommendations are made using various algorithms. It can either be item based or user based.	filtering	filtering
Model based	It is a type of collaborative filtering based RS. It involves learning through previous ratings for increasing the efficiency of collaborative filtering based RS. Model based collaborative filtering technique can use machine learning or matrix factorization techniques [46].	• Quick recommendation • Solve the data sparsity problem of collaborative filtering • Same as collaborative filtering	Same as collaborative filtering except the data sparsity problem
Hybrid filtering	It is a combination of traditional recommendation techniques used for overcoming the limitations of these systems when used individually [47]. Example: P-tango [48], DailyLearner [49]	• More accurate recommendations • Limitations of one technique are overcome by another • Overall weaknesses	

		suppressed	

2.2.3 Artificial Neural Network (ANN)

ANNs imitates the human nervous system where number of nodes in ANN represents the neurons in brain. The nodes are connected to each other by links which have weights associated with them. The nodes take data as input to perform basic operations on it and then pass the output to other nodes through connecting links. This output is known as activation value. The learning in ANNs takes place by updating weights of links whenever needed. Whether to alter the weight or not depends on the output predictions. In order to get accurate outputs, activation functions are applied on inputs. The most commonly used activation function in ANN is sigmoid function. The way of modification of connection weights can be categorized as follows:

- *Supervised Learning:* The ANN is given a dataset. The output of ANN is compared with the desired output present in the dataset. Based on the difference between predicted and actual values, the weights are updated. This process repeats until ANN gives the desired output.
- *Unsupervised Learning:* This way of training is independent. During training, ANN clusters the same kind of input data into clusters. For a new input data, ANN categorizes it into one of the clusters. No feedback on results is provided in this case.

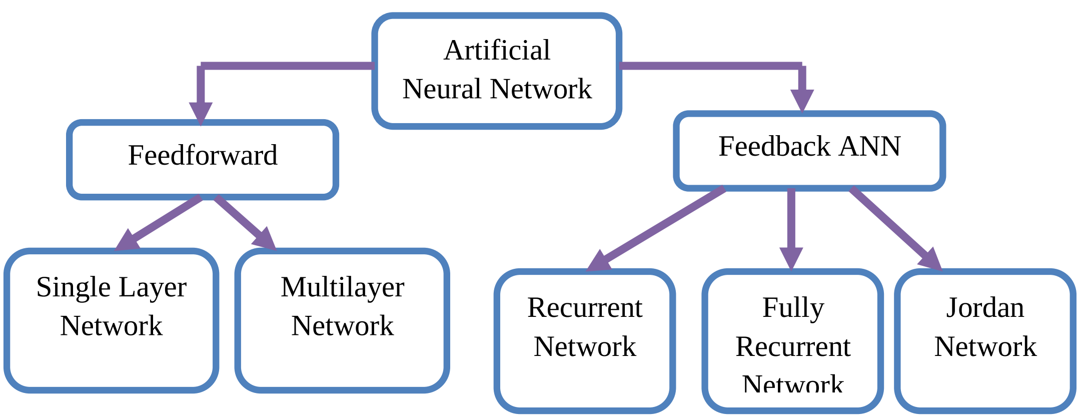

Fig. 2.2. Types of ANN

- *Reinforcement Learning:* It is similar to supervised learning as in this case ANN gets feedback for its outputs. The difference is that here the feedback is evaluative instead of instructive. The weights are adjusted on the basis of these feedbacks.

There are two types of ANN topologies: Feedback (feedback loops present) and Feed-forward (unidirectional information flow, no feedback loops). Further classification is shown in Fig. 2.2. Many practical applications of ANN have been reported in the areas of speech recognition, signature application, face recognition and pattern recognition.

2.2.4 Fuzzy Inference System (FIS)

While dealing with real life problems, most of the times, one doesn't have solutions in terms of either yes or no. One has a vague idea which also changes continuously. Instead of customary true and false, fuzzy logic describes the outcomes in a range from true to false or 0 to 1. The most important unit of fuzzy logic vital for decision making is FIS. For drawing decision rules, it uses if-then and AND-OR operators. Input to FIS can either be fuzzy or crisp but the output is always fuzzy.

The following fig. 2.3 depicts the working of a FIS.

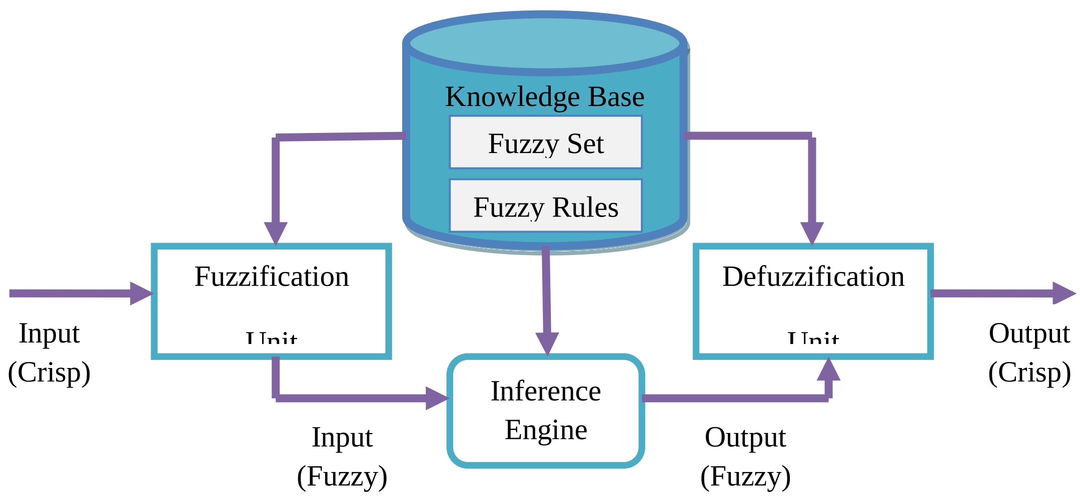

Fig.2.3. Fuzzy Inference System

There are a wide range of applications of fuzzy logic within various domains such as automotive, aerospace, defense, finance, electronics, medical, transportation, security, pattern mining and classification, psychology and business. Fuzzy logic is used in neural networks because:

- Fuzzy logic provides an intended effect when used for defining weights in neural networks.
- For parallel processing because when fuzzy logic is used in neural networks then the values mist be fuzzy.
- When it is not possible to have crisp values, fuzzy values solve the problem.
- In unexpected situations, neural networks should perform better. Fuzzy values are better than crisp values to do the same.

Vice versa, neural networks are used to train fuzzy systems because of the following advantages:

- Preprocessing can be made better because it becomes very easy to learn new data patterns when neural networks are used.
- Fuzzy rules can be refined when neural network is used because neural networks are very efficient in learning relationships with new data.

2.2.5 Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS combines the best features of neural network (ANN) and fuzzy inference system (FIS)[50]. It is defined as a multilayer feed-forward architecture where each neuron executes its function for arriving signals.

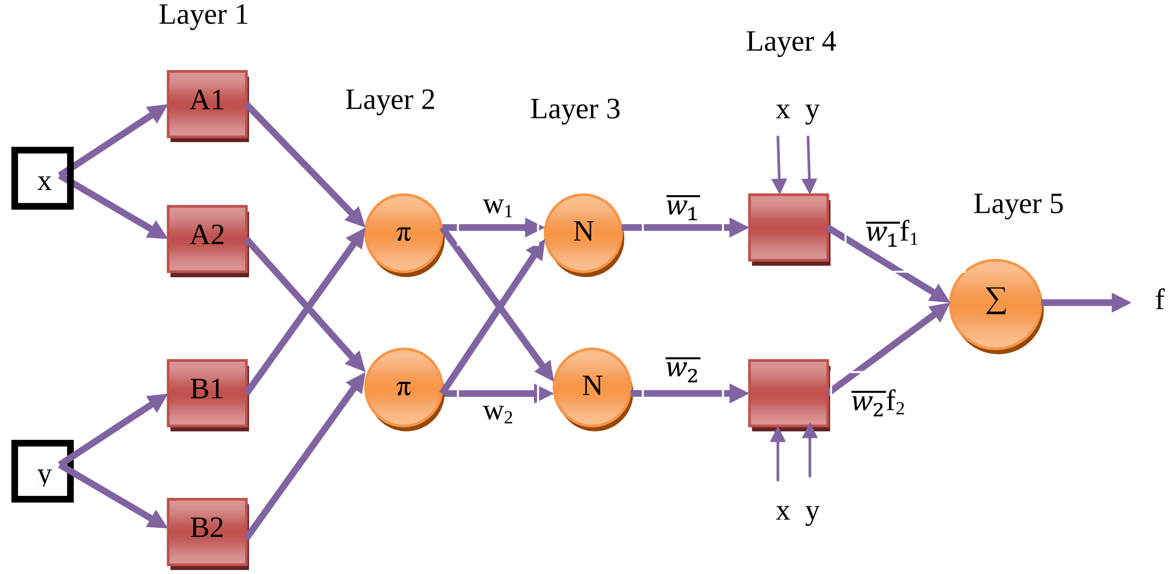

Fig. 2.4.ANFIS Architecture [8]

It maps input features to input membership functions (MF). Input membership functions are then mapped to the set of if-then rules and then the rules are mapped to a set of output features. Finally, the output features are mapped to output membership functions and the output membership functions to the decision which is associated with the output. ANFIS is an effective soft computing technique with easy implementation and easy incorporation of both numeric and linguistic knowledge for solving a problem. Fig. 2.4 shows the architecture of an ANFIS. Each layer in ANFIS contributes in calculating the parameters with the help of the respective functions present in them. The working of each layer is briefly described as follows:

- *Layer 1:* Every node in layer 1 is an adaptive node. In this layer, the output of each node is fuzzy membership grade of the inputs. Here, the parameters are known as premise parameters. During the process of learning, different parameters are used for the specification and tuning of membership functions. The membership function can be represented in equation 1:

$$\mu_{Ai}(x) = \frac{1}{1+\left|\frac{x-c_i}{a_i}\right|^{2b}} \quad (1)$$

where, $\mu_{Ai}(x)$ lies between 0 and 1;

a_i, b, c_i are parameter set

- *Layer 2:* Each node is a fixed node in this layer. Product of input signals is the output of the nodes in this layer which can be written as given in equation 2:

$$O_{2,i} = w = \mu_{Ai}(x).\mu_{Bi}(x)(2)$$

- *Layer 3:* All nodes in this layer are fixed nodes. They calculate the ratio of the i^{th} rule's firing strength relative to the sum of all rule's firing strengths. The normalized firing strength is the output which can be written as given in equation 3:

$$O_{2,i} = \overline{w} \times w_1 / (w_1 + w_2) \quad (3)$$

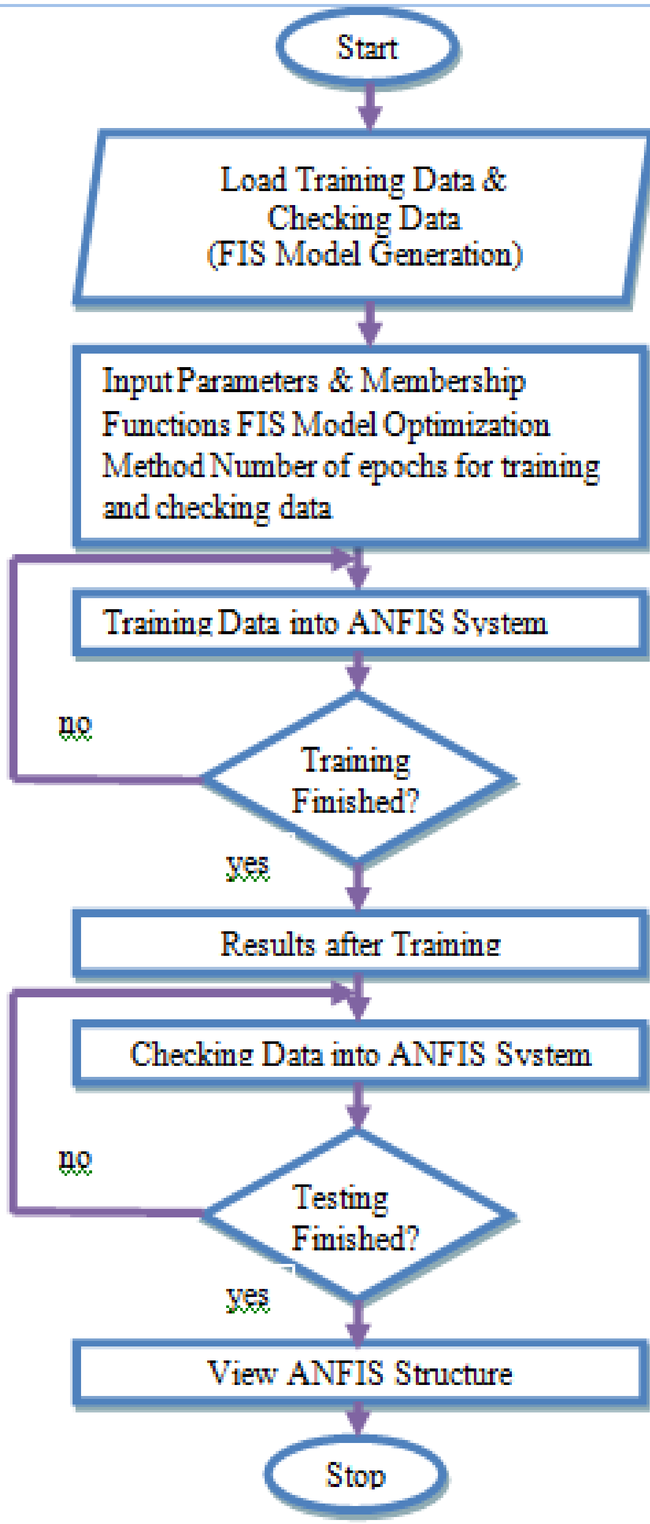

Fig. 2.5. ANFIS Training Process

- *Layer 4:* Each node in layer 4 is an adaptive node. The output of these nodes can be defined as given in equation 4:

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) (4)$$

where, $\{p_i, q_i, r_i\}$ is the parameter referred to as consequent parameters

- *Layer 5:* In this layer, all nodes are fixed nodes. Their job is to sum all arriving signals and output that as total output. The total output can be defined as given in equation 5:

$$O_{5,i} = \sum \overline{w_i} f_i (5)$$

ANFIS identifies the rules on its own and tunes the membership function parameters accordingly. In order to develop an efficient model, some choices should be made carefully. For example, setting the number and type of membership functions, optimization methods, types of output MFs and the number of epochs. Fig.2.5 depicts the ANFIS training process.

Chapter 3

Proposed Method

The proposed ANFIS based tweet recommender model provides efficient and personalized tweet recommendation to users using content filtering and collaborative filtering techniques. The content history of users is analyzed to uncover their interests implicitly into pre-defined interest categories using a multinomial Naïve Bayes classifier, and the information from similar interest users is used to predict affinity to a given tweet. A complete process flow of the proposed model is shown in the following Fig.3.1.

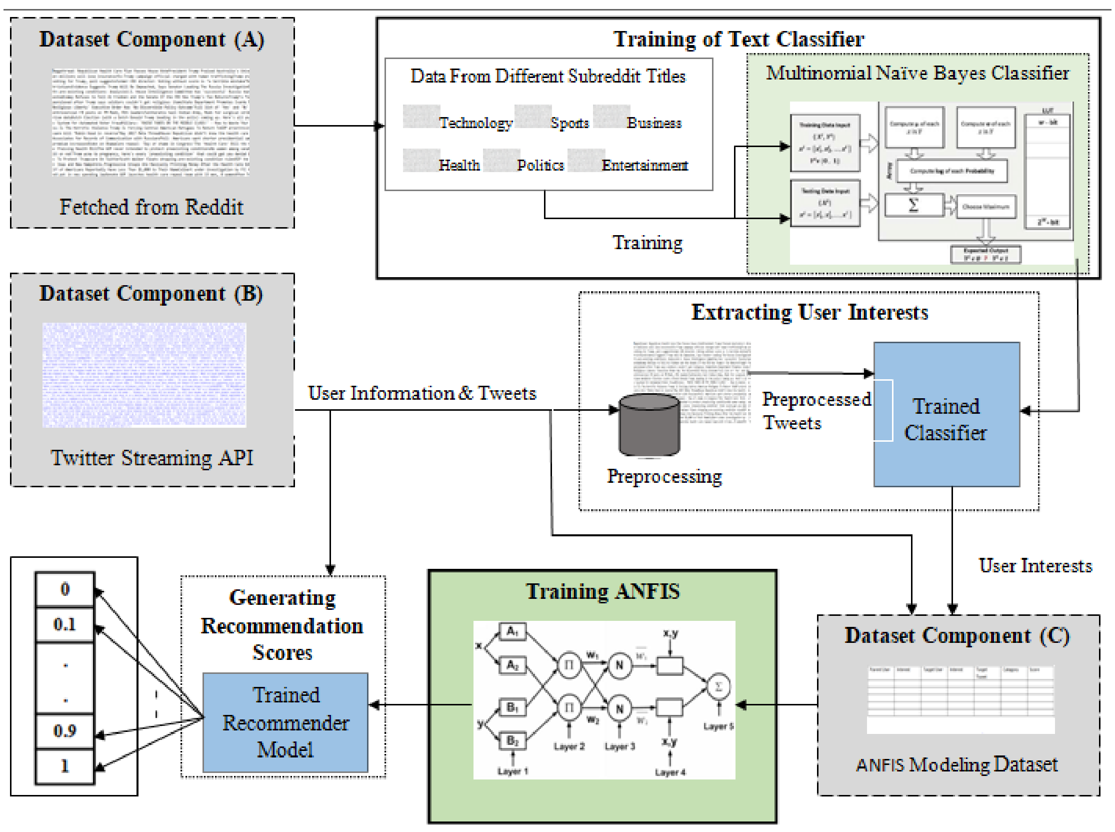

Fig. 3.1. ANFIS based tweet recommender model

3.1 Tweet Categorization

It is imperative to analyze the user tweet content , i.e., tweets/retweets/likes in order to detect the top interests of a users. Sixinterest categories for users have been identified, namely, technology,business,entertainment; politics,health&sports. Multinomial Naïve Bayes classifier has been trained to classify tweets into the six pre-defined interest categories.Multinomial Naïve Bayes Classifier is a specifictype of Naïve Bayes Classifier in which multinomial distribution is used for every feature. For training, the dataset is built by fetching textual data for various categories from the reddit website.[3] Here, the textual data refers to the headlines present on the website under each category. Using this dataset, the classifier has been trained to classify input text in the above mentioned categories. Further, the trained classifier is used to classify tweets into different categories. Thus, the interest categories are learned and the classifier is used to derive users' implicit interest by analyzing his tweets, retweets and likes.

3.2 User Interests' Extraction

The second step was to extract areas in which a user is most interested in. We used the classifier trained in 4.1 for this task. Using Twitter API, the tweets/retweets/likes of users were fetched. For all the basic preprocessing of text,NLTK[4] was used.

Preprocessing refers to the basic handling of data in order to transform it into feature vectors. It includes removal of unwanted data, stop words, tokenization, stemming, spell correction etc. [51]. After preprocessing we got feature vectors for input text (tweets). Then, these were passed to the trained text classifier to classify all the tweets into the six pre-defined categories. We noted the topmost interests of a user and built a dataset of users and their areas of interest.

---

[3]www.reddit.com.
[4]https://www.nltk.org/

3.3 Fuzzy Rule Base

Using the user data which is extracted in 4.2 and Twitter API, a training data set for ANFIS was created. The training data for ANFIS must have an output vector. In the proposed framework, the output is computed based on fuzzy inference system. So the first step is to ascertain the fuzzy rule base.The fuzzy rule base contains the basic if-then rules which play an important rule in decision making. The rule base used in this model consists of 14 if-else rules as follows:

1. $t_c = x_1$ AND ($x_1 \in \{ y_1, y_2, y_3\}$)
   then score = 1
2. if $t_c = x_1$ AND ($x_2 \in \{ y_1, y_2, y_3\}$ OR $x_3 \in \{ y_1, y_2, y_3\}$)
   then score = 0.9
3. if $t_c = x_2$ AND ($x_1 \in \{ y_1, y_2, y_3\}$ OR $x_2 \in \{ y_1, y_2, y_3\}$ OR $x_3 \in \{ y_1, y_2, y_3\}$)
   then score = 0.8
4. if $t_c = x_3$ AND ($x_1 \in \{ y_1, y_2, y_3\}$ OR $x_2 \in \{ y_1, y_2, y_3\}$ OR $x_3 \in \{ y_1, y_2, y_3\}$)
   then score = 0.7
5. if $t_c = x_1$ AND ($x_1 \notin \{ y_1, y_2, y_3\}$ AND $x_2 \notin \{ y_1, y_2, y_3\}$ AND $x_3 \notin \{ y_1, y_2, y_3\}$)
   then score = 0.6
6. if $t_c = x_2$ AND ($x_1 \notin \{ y_1, y_2, y_3\}$ AND $x_2 \notin \{ y_1, y_2, y_3\}$ AND $x_3 \notin \{ y_1, y_2, y_3\}$)
   then score = 0.5
7. if ($x_1 = y_1$ OR $x_1 = y_2$ OR $x_1 = y_3$)AND $t_c \notin \{ x_1, x_2, x_3\}$
   then score = 0.5
8. if $t_c = x_3$ AND ($x_1 \notin \{ y_1, y_2, y_3\}$ AND $x_2 \notin \{ y_1, y_2, y_3\}$ AND $x_3 \notin \{ y_1, y_2, y_3\}$)
   then score = 0.4
9. if ($x_1 = y_1$ OR $x_2 = y_1$ OR $x_3 = y_1$)AND $t_c \notin \{ x_1, x_2, x_3\}$
   then score = 0.5
10. if $t_c = x_3$ AND ($x_1 \notin \{ y_1, y_2, y_3\}$ AND $x_2 \notin \{ y_1, y_2, y_3\}$ AND $x_3 \notin \{ y_1, y_2, y_3\}$)
   then score = 0.4
11. if ($x_1 = y_2$ OR $x_1 = y_3$)AND $t_c \notin \{ x_1, x_2, x_3\}$
   then score = 0.4
12. if ($x_2 = y_2$ OR $x_2 = y_3$)AND $t_c \notin \{ x_1, x_2, x_3\}$
   then score = 0.2
13. if ($x_3 = y_2$ OR $x_3 = y_3$)AND $t_c \notin \{ x_1, x_2, x_3\}$
   then score = 0.1
14. if ($t_c \notin \{ x_1, x_2, x_3\}$ AND $\{ x_1, x_2, x_3\} \cap \{ y_1, y_2, y_3\} = \emptyset$
   then score = 0

where,

x_i: i^{th} interest of user x and $i \in \{1, 2, 3\}$

where, i = 1 is the highest priority (topmost area of interest),

i = 2 is the second priority,

i = 3 is the third priority

x : Target user, the user to whom we want to recommend tweets

y: The source user who tweeted/retweeted a tweet, t that is now being considered for recommendation

t : Target tweet, which is being considered for recommendation

t_c : Category of tweet, t and $t_c \in$ {Health, Entertainment, Business, Technology, Sports, Politics}

score: It is the recommendation score which tells about the probability that a tweet, t tweeted/retweeted by a user, y and which comes under category, t_c will be liked by a user x. The probabilities are as follows:

if score = 1 then probability = very high

if $0.7 \leq$ score ≤ 0.9 then probability = high

if $0.5 \leq$ score ≤ 0.6 then probability = moderate

if $0.1 \leq$ score ≤ 0.4 then probability = low

if score = 0 then probability = very low

Based on these fuzzy rules, we developed the training dataset with seven inputs (top three interests of x, top three interests of y, category of target tweet) and one output (score).

3.4 ANFIS System Training Process

A typical ANFIS training process was depicted in fig. 7 [52]. The first step is to obtain training and checking data sets. Training data set must have an output vector which contains the output for every set of inputs. The premise parameters for MFs are set by using training data set. A threshold error value is determined. Least squares method is used to find consequent parameters. The premise parameters are updated using the gradient decent method if error is greater than threshold. Once the error comes out to be lesser than the threshold value, the process terminates. The role of checking data set is to compare the predicted values to actual values [52]. Hybrid learning is used by ANFIS for learning while training. Hybrid learning is an integration of the gradient descent and the least squares method. The first step of ANFIS training in MATLAB is to create training data set. The data set must be in the form of a matrix where the output should be in the last column. The number of columns depend on the numner of inputs and the rows denote the existing data points. The initial set of MFs can be set with the help of the command genfis1 in MATLAB or it can also be done manually. In this study, we used the genfis1 command for the same. After setting some other options (number of epochs etc.), system training begins.

Once the training process terminates, we get the final membership functions and training. For more accuracy the checking data set can be used with the training dataset. ANFIS functions well with one training data set but, if checking data set is also provided then the chances are more that system will understand the model more accurately. Using evalfis function, system performance can be evaluated.That is, the trained model is testedfordifferent sets of inputs in order check its performance.Here, weused the dataset created in the previous section.

3.5 Working of the proposed model

In this section,astep by step approach is used to explain the whole process of the proposed ANFIS based tweet recommender, using an example.

4.5.1. Working of text classifier

The text classifier which is trained by multinomial Naïve Bayes classifier takes a tweet as an input and gives its category as output as shown in Table 4.1.

Table 4.1: Text Categorization

Input (Tweet)	**Output (Category)**
Trying to cut sugar out of your diet? Freeze bananas. They're much sweeter that way and you'll have a tasty treat.	Health
Reserve Money for the week ended May 17, 2019 and Money Supply for the fortnight ended May 10, 2019	Business
Making of the democracy! Are you ready for May 23? Get the real time results of #LokSabhaElections2019 on the website:	Politics

http://www.results.eci.gov.in	

4.5.2. Extracting User Interests

Using the text classifier, we extract the top interests of users. The process has been illustrated for a user with user id: narendramodi. First we fetched 1000 tweets of Mr. Narendra Modi username: narendramodi) and passed them through the text classifier. Based on that we found out the top three interests of the user. In this case the topmost interest is 'Politics' (661 tweets), second is 'Technology' and third 'Business'. The classification results that ware used to extract user interests of the user, *narendramodi* is shown in Table 4.2.

Table 4.2: Extracting User Interests

Category	Number of tweets (1000)
Technology	143
Business	125
Politics	661
Entertainment	37
Sports	22
Health	12

The result clearly shows that the user whose username on Twitter is 'narendramodi' has interests in the following order: Politics→Technology→Business→Entertainment→Sports→Health

Similarily the interests of all the users are extracted. The top three interests of every user are used to build the dataset.

4.5.3. Data set for ANFIS training

The database created for training ANFIS has 7 input vectors and one output vector. Here, it is explained with the example where the score is computed for the user whose username is 'narendramodi'. Some instances are shown below:

Target Username, x: narendramodi

Area of interests (in order):

x_1 = Politics

x_2 = Technology

x_3 = Business

'y' are the users who tweeted/retweeted the tweetswhich are under consideration for recommendation to x. So the top areas of interests for users 'y' are also extracted.

Recommendation score for recommending a tweet 't' that comes under category 't_c' and is tweeted/retweeted by user y (who has interests y_1, y_2, y_3) to the user x (narendramodi, having interests: x_1, x_2, x_3) is shown in Table 5. Here, the score is computed using the fuzzy rules explained in section 4.3. For example, the tweet 'tweet$_1$' has been tweeted by a user 'y' whose areas of interest are in the order as follows: Entertainment→Politics→Business.

Therefore, areas of interest of source user 'y' (in order):

y_1 = Entertainment

y_2 = Politics

y_3 = Business

The tweet 'Tweet$_1$' comes under the category 'Entertainment'.

Therefore, t_c=Entertainment

Here, $x_1 = y_2$=Politics;

$t_c \notin \{ x_1, x_2, x_3\}$ as Entertainment ∉ { Politics, Technology, Business}

This case comes under rule number 11 in fuzzy rule base mentioned in section 4.3 i.e., **if** ($x_1 = y_2$OR $x_1 = y_3$)AND $t_c \notin \{ x_1, x_2, x_3\}$ **then** score = 0.4.

Therefore, recommendation score is 0.4 for tweet$_1$ which means that the probability that tweet$_1$ will be liked by the user x (narendramodi) is low. Similarly the score is computed for all the tweets as shown in Table 4.3.

Table 4.3: Predicting Recommendation Score

T	**t_c**	**y_1**	**y_2**	**y_3**	**Score**	**Probability**
Tweet$_1$	Entertainment	Entertainment	Politics	Business	0.4	Low
Tweet$_2$	Technology	Politics	Sports	Technology	0.8	High

Tweet_3	Entertainment	Entertainment	Health	Sports	0	Very Low
Tweet_4	Sports	Sports	Entertainment	Business	0.1	Low
Tweet_5	Health	Health	Entertainment	Politics	0.4	Low
Tweet_6	Politics	Politics	Entertainment	Sports	1	Very high
Tweet_7	Politics	Technology	Business	Health	0.9	High

where,

Tweet_1: "*Disaster movies (as tragic as they are) have always been a genre I like to watch! #kedarnath is a strong leap into that genre! The film moves you in the last act and the romance is beautifully captured throughout! For all those thirsting an honest love story !This is the one*!"

Tweet_2: "*I just used this awesome web app to compress a video, https://clipchamp.com via @clipchamp*"

Tweet_3: "*Saw @SuiDhaagaFilm for the second time last night and I loved it more than the first time. What an emotional rollercoaster with brilliant performances by the entire cast. #SuiDhaagaMadeInIndia*"

Tweet_4: "*First Indian Female Wrestler to win Gold at Commonwealth Games, Geeta Phogat , speaking at #IBMYoutsav Thank You Geeta for sharing your most awe inspiring story of grit and passion.*"

Tweet_5: "*In our news digest this week: why a 'universal cancer test' is not yet ready for patients: http://po.st/fC50gu*"

Tweet$_6$: "*It's in India's interest to seek early resolution of its territorial dispute with China. But this does not suit the Chinese game plan.*"

Tweet$_7$: "*Big victory for @JKPC_ in First phase of Panchayat Polls. JKPC won 20 out of 22 Panchayat's and 130 out of 150 Panch in upper & Lower Ramhall handwara constituency. Congratulations to @JKPC_ Family. It's the result of their hard work and dedication.*"

Chapter 4

Results and Discussion

After the identification of inputs and output, the next step is the validation of the quality of the framework. The evaluation of the performance of the ANFIS model was carried out. The best set for training and checking was chosen on the basis of root-mean-square-error (RMSE). The accuracy was measured by comparing the predicted and actual values. We set various factors like data set sample, number of epochs, membership function type and number andnumber of inputs very carefully to achieve the most efficient results.The details of validation results is described in the following sub-sections.

4.1 Text Categorization Performance Analysis

This section describes the performance analysis of multinomial Naïve Bayes classifier for text categorization. There are a number of metrics which are used for the evaluation of text classification algorithms. The most popular ones are accuracy, precision, recall and F-measure [53]. So, here we have used these two for evaluating the performance of text categorizer used in the proposed framework.

- Precision: The ratio of words/sentences/documents which are correctly assigned the category A to the total number of words/sentences/documents assigned category A is known as precision.
- Recall: The ratio of words/sentences/documents which are correctly assigned the category A to the total number of words/sentences/documents which are actually in category A is known as recall.

  Recall and precision can also be computed as:

$$Precision = {}^{a}/_{(a+b)} \qquad Recall = {}^{a}/_{(a+c)}$$

where

a: number of category A documents which are classified into A

b: number of non-A documents which are classified into A c: number of Category A documents which are classified as non-A

Performance of a classifier is also measured using the combination of precision and recall. For example, F-measure, precision-recall breakeven point (BEP). Sometimes, such measures prove to be more effective. F-measure can be computed as:

$$F - measure = 2PR/(P + R)$$

where

P: Precision; R: Recall

BEP is the value whereprecision and recall are equal to recall. Often, it is computed bycalculating the arithmetic mean of precision and recall. In this study, accuracy is also been used for the performance evaluation. Accuracy is measured as the ratio of correctly classified words/sentences/documents to the total number of words/sentences/documents.

The training data set used in this work was built using the headlines fetched from www.reddit.com under the titles related to six categories. This dataset was used to evaluate Naïve Bayes classifier, SVM and multinomial Naïve Bayes classifier on same number of training instances. Figure 4.1 displays the graphical representation of accuracy results of all these classifiers. These results show that multinomial Naïve Bayes classifier performs better that the other two classifiers on most training data sizes. Therefore, the text categorizer is trained using multinomial Naïve Bayes classifier.

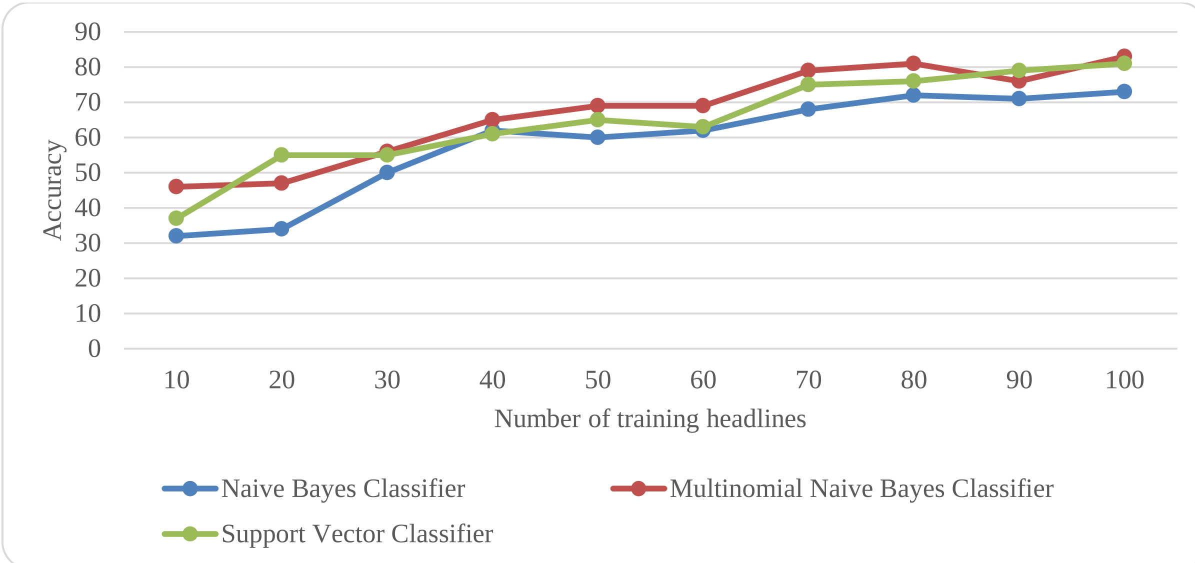

Fig. 4.1. Comaprison of Tweet category classifiers

After training, we computed all the above mentioned performance metrics for each category. This has been done using the confusion matrix which is a result of training and testiong of text categorizer. A confusion matrix is a table/matrix which displays the summary of prediction results. The confusion matrix that we got when we used multinomial Naïve Bayes classifier to categorize text into six categories is shown in Table 4.4.

Table 4.4: Confusion Matrix

Predicted→ Actual↓	Politics	Health	Entertainment	Business	Technology	Sports
Politics	**105**	0	16	7	3	0
Health	0	**61**	0	3	11	0
Entertainment	13	0	**120**	4	0	16
Business	0	0	0	**50**	7	3
Technology	0	11	0	8	**66**	1
Sports	0	0	6	9	3	**77**

Based on the confusion matrix we computed, the values of true positive, false positive, true negative and false negative. Using these precision and recall were computed. Further, BEP and F-measure is also computed. The result in terms of Accuracy, BEP and F-measure is shown is Table 4.5.

Table 4.5: Performance of multinomial Naïve Bayes classifier for text categorization

Category	**Accuracy**	**BEP**	**F-measure**
Politics	80.2	84.6	84.3
Health	81.3	83.0	82.9
Entertainment	78.4	81.5	81.3
Business	83.3	72.5	70.9
Technology	76.7	75.0	74.9
Sports	81.1	80.2	80.2
Average	**80.2**	**79.5**	**79.1**

4.2 ANFIS Model Performance Analysis

The ANFIS framework has been modeled using 950 train data pairs. For achieving the most appropriate recommendation score predictions, various modifications have been done. The MATLAB R2016a was used to train the ANFIS and obtain results[54]. There are different kinds of membership functions (MF) in ANFIS as mentioned below:

- Triangular MF (Trimf)
- Trapezoidal MF (Trapmf)
- Generalized bell MF (Bellmf)
- Gaussian curve MF (Gaussmf)
- Gaussian combination MF(Gauss2mf)
- π-shaped MF (Pimf)
- Difference between two sigmoidal MF (dsigmf)
- Product of two sigmoidal MF (psigmf)

We evaluated all these MFs for a small dataset (250 train data pairs) to get to know the best performer. The results of the same are displayed in Table 4.6.

Table 4.6: Performance of different MFs

Membership Function	Root Mean Square Error	
	Training	**Testing**
Trimf	0.0009	1.734
Trapmf	0.076	7.713
Bellmf	0.0005	1.5369
Gaussmf	0.0024	0.9257
Gauss2mf	0.06	5.6853
Pimf	0.3562	8.368
Dsigmf	0.0035	3.752
Psigmf	0.0035	3.648

As shown in Table 8, Gaussmf provided the best results. So for full dataset (950 train data pairs), we chose Gaussian MF for the training and testing of ANFIS. Gaussmf is a function which uses Gaussian membership function to calculate the fuzzy membership values. The performance (training and testing error) largely depends on the size of training data set. Therefore, we increased the size of dataset for actual training.

The number of epochs was set to 70. Setting of the number of epochs depends on the MF. It adjusts the MF parameters for optimization. Also, the number of iterations plays a crucial role in reducing the error. We used the hybrid learning algorithm method (an integration of the gradient descent and the least squares method). In hybrid learning algorithm, there is one forward pass and one backward pass in each epoch [8]. The input data and functional signals are forwarded in order to calculate the output of each node in a forward pass. The error rates get propagated from the output side toward the input side in a backward pass. In order to avoid over fitting in the process of model development, ANFIS automatically sets the MF parameters. For generating the FIS, the grid partitioning technique was implemented. Grid partitioning is the default partitioning method which is used to generate fuzzy inference system (FIS) for the supplied data set [54]. It clusters all the data points and generates the rules. The basic inference rules that were used for model development are mentioned in detail in Section 3. The total number of parameters and fuzzy rules for ANFIS were 2229 and 2187 respectively, as shown in table 4.7.

Table 4.7: ANFIS Information

Number of nodes	4426
Number of linear parameters	2187
Number of non-linear parameters	42
Total number of parameters	2229
Number of training data pairs	950
Number of testing data pairs	250
Number of fuzzy rules	2187

The error tolerance was set to zero. After training of ANFIS, the training data errors were plotted against the number of epochs as shown in Fig. 4.2.

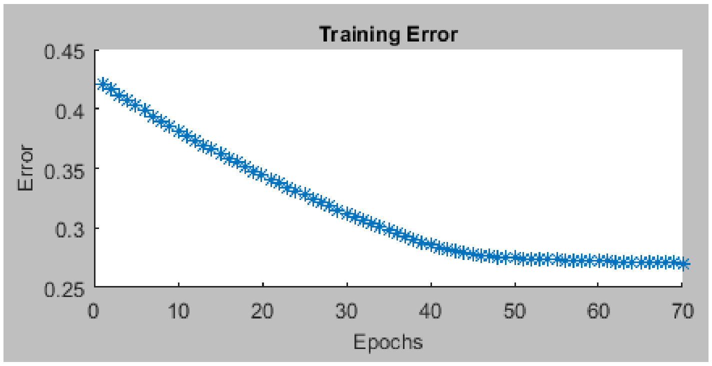

Fig. 4.2. Training Error Plot

Fig. 4.3 shows the result of testing the model against training data itself. The average testing error in this case came out to be 0.027.

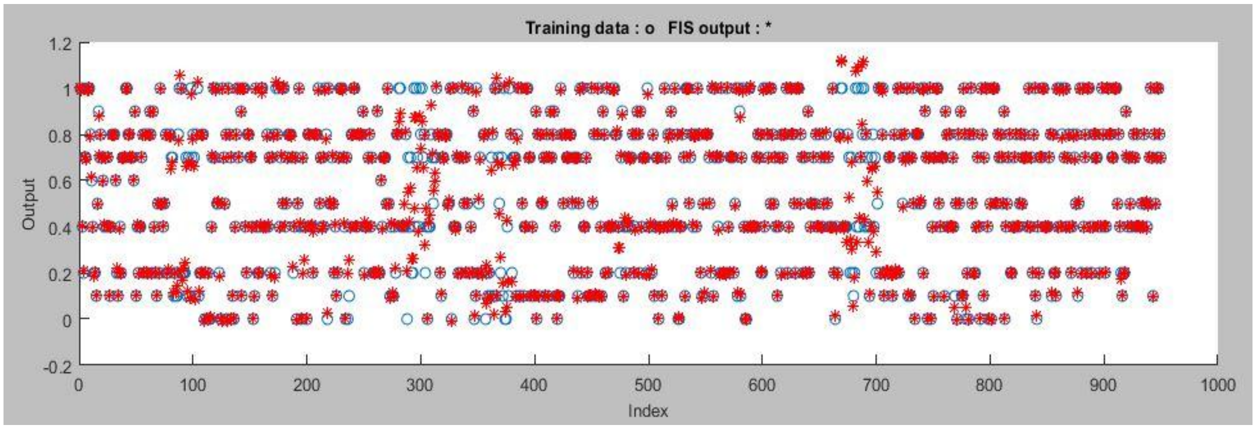

Fig. 4.3. Training data and FIS output

The performance of the proposed model by using Gaussian MF is shown in Fig. 4.4 which shows the plot of actual and predicted values of recommendation score for testing data. The average testing error came out to be 0.93.

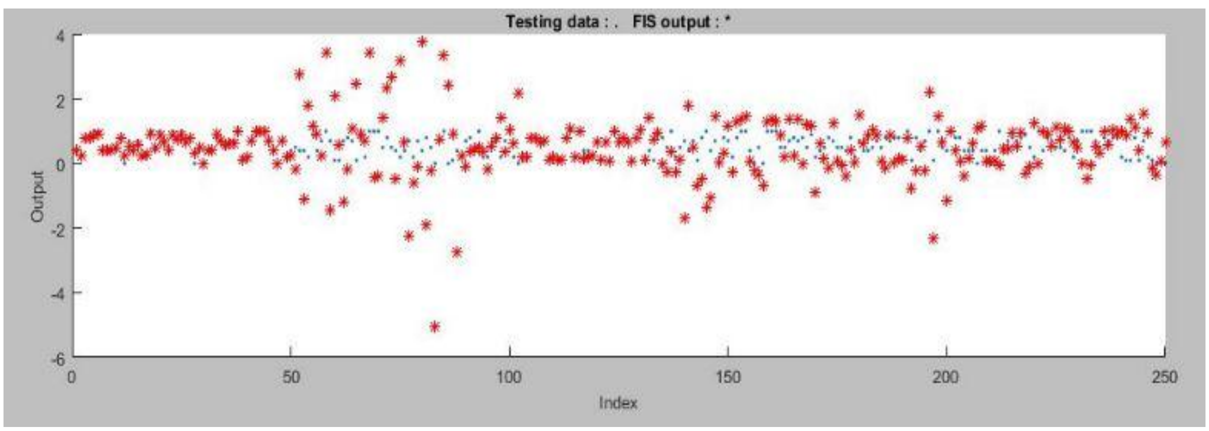

Fig. 4.4. Testing data and FIS output

The results quite clearly indicate that the proposed model predicts the recommendation score correctly most of the time.

Chapter 5

Conclusion and Future Scope

Intelligent semantic filters are required to deal with the information overload on the chaotic and complex social media portals. Recommender systems as information filtering models are decision making tools for personalizing user experience in these large and complex information environments. This research proffered a novel ANFIS based tweet recommender model to recommend worthy user interest-based tweets reducing the browsing effort to locate information. An inference rule base was built to find the correlation between the category of target tweet, interest of target user andthe interest of sourceuser. Using this hybrid recommender model, the final recommendation score tells about the probability that a tweet, tweeted/retweeted by a user, y, belonging to category $t_c$will be of user x's interest. The root mean square error (RMSE) is used to calculate the error in predicted and actual values and demonstrates the effectiveness of adaptive neuro-fuzzy based tweet recommender system in practice. Future research will aim to improve the proposed framework by considering images, videos, emoticons and hyperlinks for generating recommendation score. Moreover, instead of only six categories, a more detailed set of categories of user interests would provide more specific and accurate areas in which a particular user is interested which would improve the recommendation model.

References

1. Yakushev, A.V., Boukhanovsky, A.V. and Sloot, P.M. (2013). October. “Topic crawler for social networks monitoring”. In International Conference on Knowledge Engineering and the Semantic Web (pp. 214-227). Springer, Berlin, Heidelberg.
2. Sloot, P.M.,Janusz, H., George, K. and Harold, L.M. (2018). Supercomputer Simulation of Critical Phenomena In Complex Social Systems. Journal Scientific and Technical of Information Technologies, Mechanics and Optics, 118(6), p.967.
3. Zarrabi, N., Prosperi, M.C., Belleman, R.G., Di Giambenedetto, S., Fabbiani, M., De Luca, A. and Sloot, P.M. (2013). Combining social and genetic networks to study HIV transmission in mixing risk groups. The European Physical Journal Special Topics, 222(6), pp.1377-1387.
4. Kumar, A., Arora, H., Singh, N K, Gupta, D. Khanna, A. (2018). “Supported Matrix Factorization using Distributed Representations for Personalized Recommendations on Twitter”, Computers and Electrical, Elsevier, Volume 71, October 2018, Pages 569-577.
5. Bobadilla J., Ortega F., Hernando A., Gutiérrez A. (2013). “Recommender systems survey”, Knowledge-Based Systems, Volume 46, July 2013, Pages 109-132.
6. Kumar A., Jaiswal A. (2019). “Systematic literature review of sentiment analysis on Twitter using soft computing techniques”, Concurrency Computation PractExper. https://doi.org/10.1002/cpe.5107
7. Kywe S.M., Lim EP., Zhu F. (2012). "A Survey of Recommender Systems in Twitter", In: Aberer K., Flache A., Jager W., Liu L., Tang J., Guéret C. (eds) Social Informatics. SocInfo 2012. Lecture Notes in Computer Science, vol 7710. Springer, Berlin, Heidelberg
8. J. -. R. Jang, (1993). "ANFIS: adaptive-network-based fuzzy inference system," in IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 3, pp. 665-685, May-June 1993. doi: 10.1109/21.256541

9. Suparta W, Alhasa, KM, (2016). “Adaptive Neuro-Fuzzy Inference System” in Modeling of Tropospheric Delays using ANFIS 2016, XVI 109p. 51 ilus
10. F. Abel, Q. Gao, G. J. Houben, and K. Tao, Analyzing user modeling on Twitter for personalized news recommendations, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics),vol. 6787 LNCS, pp. 1–12 (2011).
11. Jonnalagedda, Nirmal &Gauch, Susan, Personalized News Recommendation Using Twitter, 21-25. 10.1109/WI-IAT.2013.144 (2013).
12. De Francisci Morales, G., Gionis, A., Lucchese, C., From Chatter to Headlines: Harnessing the Real-Time Web for Personalized News Recommendation, In: The 5th ACM International Conference on Web Search and Data Mining (2012).
13. Uysal, I., Croft, B.W., User Oriented Tweet Ranking: a Filtering Approach to Microblog, In: The 20th ACM International Conference on Information and Knowledge Management (2011).
14. Armentano, M.G., Godoy, D.L., Amandi, A.A.:, A Topology-Based Approach for Followees Recommendation in Twitter, In: 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems. Barcelona, Spain (2011).
15. Armentano, M.G., Godoy, D.L., Amandi, A.A., Recommending Information Sources to Information Seekers in Twitter, In: International Workshop on Social Web Mining
16. Armentano, M.G., Godoy, D.L., Amandi, A.A., Towards a Followee Recommender System for Information Seeking Users in Twitter, In: The 2nd International Workshop on Semantic Adaptive Social Web.
17. Garcia, R., Amatriain, X., Weighted Content Based Methods for Recommending Connections in Online Social Networks, In: The 2nd ACM Workshop on Recommendation Systems and the Social Web. Barcelona, Spain (2010).
18. Golder, S.A., Marwick, A., Yardi, S., Boyd, D., A structural approach to contact recommendations in online social networks, In: Workshop on Search in Social Media, In conjunction with ACM SIGIR Conference on Information Retrieval.
19. J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. H. Chi, Short and tweet: Experiments on recommending content from information streams, in Proc. SIGCHI Conf. Human Factors Comput. Syst., New York, NY, USA, pp. 1185–1194 (2010).

20. I. Uysal and W. B. Croft, User Oriented Tweet Ranking_: A Filtering Approach to Microblogs, in Proc. 20th ACM Int.Conf. Inf. Knowl. Manage., New York, NY, USA, pp. 2261–2264 (2011).
21. N. Yazdanfar and A. Thomo, Link recommender: Collaborative-filtering for recommending URLs to twitter users, Procedia Comput. Sci., vol. 19, no. Ant, pp. 412–419 (2013).
22. S. Asur, B. A. Huberman, G. Szabo, and C. Wang, Trends in social media: Persistence and decay, in Proc. 5th Int. AAAI Conf. Web SocialMedia, pp. 434–437 (2011).
23. H. Ma, M. Jia, X. Lin and F. Zhuang, Tag correlation and user social relation based microblog recommendation, International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 2424-2430. doi: 10.1109/IJCNN.2016.7727500 (2016).
24. C. Wang and B. A. Huberman, Long trend dynamics in social media, EPJ Data Sci., vol. 1, no. 2, p. 2, (2012).
25. E. Zangerle, W. Gassler, and G. Specht, Recommending #- tags in Twitter, CEUR Workshop Proc., vol. 730, (2011)
26. Z. Zhao et al., Modeling Chinese microblogs with five Ws for topic hashtags extraction, in Tsinghua Science and Technology, vol. 22, no. 2, pp. 135-148, doi: 10.23919/TST.2017.7889636 (2017).
27. J. Ma, C. Feng, G. Shi, X. Shi and H. Huang, Temporal enhanced sentence-level attention model for hashtag recommendation, in CAAI Transactions on Intelligence Technology, vol. 3, no. 2, pp. 95-100, 6 2018. doi: 10.1049/trit.2018.0012 (2018).
28. Li, J., Xu, H., He, X., et al., Tweet modeling with LSTM recurrent neural networks for hashtag recommendation. IEEE Int. Joint Conf. on Neural Networks (IJCNN), Vancouver, Canada, , pp. 1570–1577 (2016).
29. Gong, Y., Zhang, Q.: ‘Hashtag recommendation using attention-based convolutional neural network’. Int. Joint Conf. on Artificial Intelligence, New York, NY, USA, (2016).
30. Li, Y., Liu, T., Jiang, J., et al., Hashtag recommendation with topical attention-based LSTM. Proc. COLING 2016, the 26th Int. Conf. on Computational Linguistics: Technical Papers, Osaka, Japan, 11–17 December 2016, pp. 3019–3029 (2016).

31. Huang, H., Zhang, Q., Gong, Y., et al., Hashtag recommendation using end-to-end memory networks with hierarchical attention. Proc. COLING 2016, the 26th Int. Conf. on Computational Linguistics: Technical Papers, Osaka, Japan, 11–17 December 2016, pp. 943–952 (2016).
32. Zhang, Q., Wang, J., Huang, H., et al., Hashtag recommendation for multimodal microblog using Co-attention network. Proc. 26th Int. Joint Conf. on Artificial Intelligence, Melbourne, Australia, (2017).
33. M. S. Tajbakhsh and J. Bagherzadeh, Microblogging Hash Tag Recommendation System Based on Semantic TF-IDF: Twitter Use Case, 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Vienna, 2016, pp. 252-257. doi: 10.1109/W-FiCloud.2016.59 (2016).
34. K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu, Collaborative personalized tweet recommendation, Proc.35th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '12, no. 800, p. 661, (2012).
35. M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, Probabilistic author-topic models for information discovery, Proc. 2004 ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '04, no. 1990, p. 306, (2004).
36. S. Khater, A. L. Kavanaugh, and J. D. Ivory, Personalized Recommendation for Online Social Networks Information_: Personal Preferences and Location Based Community Trends, IEEE Transactions on Computational SocialSystems vol. 4, no. 3, pp. 104–120, (2017).
37. Burke R., Hybrid recommender systems: survey and experiments. User Model User-adapted Interact; 12(4):331–70 (2002).
38. D. Billsus, M.J. PazzaniUser modeling for adaptive news access User Model User-adapted Interact, 10 (2–3), pp. 147-180 (2000).
39. R.J. Mooney, L. Roy, Content-based book recommending using learning for text categorization Proceedings of the fifth ACM conference on digital libraries, ACM , pp. 195-204 (2000).
40. J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems ACM Trans Inform Syst, 22 (1) pp. 5-53 (2004).

41. R. BurkeWeb recommender systems P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), The Adaptive Web, LNCS 4321, Springer, Berlin Heidelberg (Germany) , pp. 377-408, 10.1007/978-3-540-72079-9_12 (2007)
42. D.H. Park, H.K. Kim, I.Y. Choi, J.K. Kim, A literature review and classification of recommender systems research Expert Syst Appl, 39 (11) , pp. 10059-10072 (2012).
43. J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, J.Riedl, Applying collaborative filtering to usenet newsCommun ACM, 40 (3), pp. 77-87 (1997),
44. U. Shardanand, P. Maes, Social information filtering: algorithms for automating “word of mouth” Proceedings of the SIGCHI conference on human factors in computing systems, ACM Press/Addison-Wesley Publishing Co., pp. 210-217 (1995).
45. Zhao ZD, Shang MS., User-based collaborative filtering recommendation algorithms on Hadoop. In: Proceedings of 3rd international conference on knowledge discovering and data mining, IEEE Computer Society, Washington DC, USA; (2010). p. 478–81. doi: 10.1109/WKDD.2010.54.
46. Yehuda Koren, Robert Bell, and Chris Volinsky, Matrix Factorization Techniques for Recommender Systems. Computer 4, 30-37. (2009). DOI=10.1109/MC.2009.263 http://dx.doi.org/10.1109/MC.2009.263.
47. G. Adomavicius, J. Zhang, Impact of data characteristics on recommender systems performance ACM Trans Manage Inform Syst, 3 (1) (2012)
48. Claypool M, Gokhale A, Miranda T, Murnikov P, Netes D, Sartin M. Combining content- based and collaborative filters in an online newspaper. In: Proceedings of ACM SIGIR workshop on recommender systems: algorithms and evaluation, Berkeley, California; 1999.
49. Billsus D, Pazzani MJ. A hybrid user model for news story classification. In: Kay J, editor. In: Proceedings of the seventh international conference on user modeling, Banff, Canada. Springer-Verlag, New York; 1999. p. 99–108.
50. M. Sugeno, An introductory survey of fuzzy control, Inform Sciences, 36 (1-2) 59-83 (1985).
51. Kumar A, Jaiswal A. Empirical Study of Twitter and Tumblr for Sentiment Analysis using Soft Computing Techniques. Proceedings of the World Congress on Engineering and Computer Science 1: 1-5 (2017).

52. Al-Hmouz, A., Shen, J., Al-Hmouz, R., & Yan, J., Modeling and Simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for Mobile Learning. IEEE Transactions on Learning Technologies, 5, 226-237. (2012).
53. Bhatia M P S, Kumar A. A primer on the web information retrieval paradigm, Journal of Theoretical & Applied Information Technology 4 (7). (2008).
54. MathWorks, Fuzzy logic toolbox user's guide, Natick: Inc, 3 Apple Hill Drive 137-179 (2016).

Appendix A

List of Publications

Journal (Communicated)

1. Jain D., Kumar A., Sharma V., Tweet Recommender Model using Adaptive Neuro-Fuzzy Inference System, Future Generation Computer Systems (ISSN: 0167-739X).