Computer Science & Information Technology 90

Natarajan Meghanathan
David C. Wyld (Eds)

Computer Science & Information Technology

7th International Conference on Soft Computing, Artificial Intelligence and Applications (SAI 2018), July 14~15, 2018, Chennai, India

AIRCC Publishing Corporation

Volume Editors

Natarajan Meghanathan,
Jackson State University, USA
E-mail: nmeghanathan@jsums.edu

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

ISSN: 2231 - 5403
ISBN: 978-1-921987-88-5
DOI : 10.5121/csit.2018.81001 - 10.5121/csit.2018.81008

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The 7th International Conference on Soft Computing, Artificial Intelligence and Applications (SAI 2018) was held in Chennai, India during July 14~15, 2018. The 7th International Conference on Advanced Information Technologies and Applications (ICAITA 2018), The 4th International Conference on Computer Science, Information Technology and Applications (CSITA 2018), The 4th International Conference on Image and Signal Processing (ISPR 2018) and The 5th International Conference on Signal and Image Processing (Signal 2018) was collocated with The 7th International Conference on Soft Computing, Artificial Intelligence and Applications (SAI 2018). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The SAI-2018, ICAITA-2018, CSITA-2018, ISPR-2018, Signal-2018 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically. All these efforts undertaken by the Organizing and Technical Committees led to an exciting, rich and a high quality technical conference program, which featured high-impact presentations for all attendees to enjoy, appreciate and expand their expertise in the latest developments in computer network and communications research.

In closing, SAI-2018, ICAITA-2018, CSITA-2018, ISPR-2018, Signal-2018 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the SAI-2018, ICAITA-2018, CSITA-2018, ISPR-2018, Signal-2018.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

Natarajan Meghanathan
David C. Wyld

Organization

General Chair

David C. Wyld	Southeastern Louisisna University, USA
Jan Zizka	Mendel University in Brno, Czech Republic

Program Committee Members

Adnan Rawashdeh	Yarmouk University, Jordan
Agoujil Said	University of Moulay Ismail Meknes, Morocco
Ahmad Qawasmeh	The Hashemite University, Jordan
Ahmed Salamh	Zawia University, Libya
Alessio Ishizaka	University of Portsmouth, United Kingdom
Anand Nayyar	Duy Tan University, Vietnam
Arindam Sarkar	University of Kalyani, India
Azam Khalili	University of Malayer, Iran
Benaissa Mohamed	Univ Ctr Of Ain Temouchent, Algeria
Bin Cao	Hebei University of Technology, P.R. China
Bingwen Feng	Jinan University, China
Bouchra Marzak	Hassan II University, Morocco
Burdescu Dumitru Dan	University of Craiova, Romania
Dabin Ding	University of Central Missouri, United States
Emad Al-Shawakfa	Yarmouk University, Jordan
Goran Bidjovski	International Balkan University, Macedonia
Gridaphat Sriharee	King Mongkut's University of Technology, Thailand
Guruprasad Khataniar	Gauhati University, India
Haibo Yi	Shenzhen Polytechnic, China
Hamed Al-Rubaiee	University of Bedfordshire, United Kingdom
Hamid Ali Abed AL-Asadi	Basra University, Iraq
Hani Bani-Salameh	Hashemite University, Jordan
Hongzhi	Harbin Institute of Technology, China
Hyunsung Kim	Kyungil University, Korea
Ireneusz Kubiak	Military Communication Institute, Poland
Israa Shaker Tawfic	Ministry of Science and Technology, Iraq
Issa Atoum	The World Islamic Sciences and Education, Jordan
Iyad alazzam	Yarmouk University, Jordan
Joey s.Aviles	Panpacific Univeristy North Philippines, Philippines
Jose-Luis Verdegay	University of Granada, Spain
Khaled Almakadmeh	Hashemite University, Jordan
Khalilur Rhaman	BRAC University, Bangladesh
Limiao Deng	China University of Petroleum, China .
Longzhi Yang	Northumbria University, UK
Lygpapers	Sichuan University, China
Maciej Kusy	Rzeszow University of Technology, Poland
Manish Kumar	Birla Institute of Technology and Science-Pilani, India

Mirosław Kwiatkowski	AGH University of Science and Technology, Poland
Mohamed Anis Bach Tobji	University of Manouba, Tunisia
Mohamed B. El_Mashade	Al_Azhar University, Egypt
Mohamedmaher Benismail	King Saud University, Saudi Arabia
Mohammad Alshraideh	The University of Jordan, Jordan
Mohammed A. Akour	Yarmouk University, Jordan
Mohammed Nabil El Korso	Paris Nanterre University, France
Mohd Hafiz Fazalul Rahiman	Universiti Malaysia Perlis, Malaysia
Morteza Alinia Ahandani	University of Tabriz, Tabriz, Iran
N V Subba Reddy	Manipal University, India
Nadhir Ben Halima	Taibah University, Saudi Arabia
Nadjia Benblidia	Saad Dahlab University, Algeria
Nahlah Shatnawi	Yarmouk University, Jordan
Naresh Doni Jayavelu	University of Washington, Seattle, WA
Nishant Doshi	MEFGI, India
Oscar Mortagua Pereira	University of Aveiro, Portugal
Padma Shri	Manipal Institute of Technology, India
Pawel Karczmarek	The John Paul II Catholic University ,Poland
Pengfei Wu	Sichuan University, China
Pietro Ducange	SMARTEST Research Centre eCampus University, Italy
Poonam Tanwar	Manav Rachna International University, India
Prabukumar	Vellore Institute of Technology (VIT),India
Pranjal S. Bogawar	Priyadarshini College of Engineering, India
Rafael Stubs Parpinelli	State University of Santa Catarina, Brazil
Raghav Prasad Parouha	Indira Gandhi National Tribal University, India
Ramesh R. Galigekere	Manipal University, India
Ravishankar H. Kamath	Manipal Institute of Technology, India
Rhattoy	Moulay Ismail University, Morocco
Saad Al-Janabi	Al-Turath College University, Iraq
Saban Gulcu	Necmettin Erbakan University, Turkey
Said Agoujil	University of Moulay Ismail, Morocco
Saman Babaie-Kafaki	Semnan University, Semnan, Iran
Samia Nefti-Meziani	University of Salford, UK.
Shadi R . Masadeh	Isra University , Jordan
Shameem SSS	Manipal International University, Malaysia
Son Nguyen Thai	Tra Vinh University, Vietnam
Stefano Michieletto	University of Padova, Italy
Tanzila Saba	Prince Sultan University, Riyadh
Tonghan Wang	East China University of Technology, China
V M Thakare	SGB Amravati University, India
Vinay Rishiwal	MJP Rohilkhand University, India
Vivekananda Bhat	Manipal University, India
Wai Lok Woo	Newcastle University, United Kingdom
Wan Shuai	Northwestern Polytechnical University, China
Weili Zhang	eBay Inc, San Jose, CA, US
Y.K. Sundara Krishna	Krishna University, India
Yunliang JIANG	Huzhou University, P.R.China
Zhongsheng Hou	Beijing Jiaotong University, China

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Networks & Communications Community (NCC)

Soft Computing Community (SCC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

7th International Conference on Soft Computing, Artificial Intelligence and Applications (SAI 2018)

7th International Conference on Advanced Information Technologies and Applications (ICAITA 2018)

4th International Conference on Computer Science, Engineering and Information Technology (CSITY-2018)

4th International Conference on Image and Signal Processing (ISPR 2018)

5th International Conference on Signal and Image Processing (Signal 2018)

A STUDY OF DEEP LEARNING TECHNIQUES FOR CULTURAL EVENTS RECOGNITION

Aman Swaraj[1], Harshita Sahni[2], Archit Agarwal[3], Neeraj Kumar Pandey[4], Supriya Shukla[5]

[1,2,4,5]Department of Computer Science and Engineering,
College of Engineering Roorkee, Roorkee, Uttarakhand
[3]Department of Information Technology,
College of Engineering Roorkee, Roorkee, Uttarakhand

ABSTRACT

Indian Culture exceeds beyond the mere definition of 'simply how people live' as it scientifically operates according to specific, detailed knowledge of the eventual aim of life and the means to attain it. Over the years, a lot of work has been done on object recognition and scene recognition but Event Recognition is still one of the fields wherein lies a huge potential for lots of research work and so with this paper, we put forward our best step to preserve the culture of India. More than 150 images of near about 20 cultural events are collected. Results are derived from support vector machine classifier using features extracted by a pre-trained convolutional neural network- Alex Net. In most visual recognition tasks, it strongly suggested that features obtained from deep learning with convolutional nets should be the chief candidate. Our proposed framework has classified images with a comparable accuracy of 77.72 %.

KEYWORDS

Indian cultural events recognition, Convolutional neural network, Local Binary Patterns, Support Vector Machine.

1. INTRODUCTION

Indian Culture has always been an identity mark for India. Cultural heritage is undoubtedly India's Golden Feather. The Indian culture includes various forms of traditional expressions such as literature, dance, music, rituals etc. Cultural also connects with agriculture, fisheries, forestry etc.

Culture enclose our glorious history that outlines our past, molds who we are today and who we are likely to become. There are many international organizations that aim at protecting cultural heritage, some of them are UNESCO, British Council, and many others. Even people from small towns promote their culture in an antiquated way [1]. Indian heritage foundation in India has also shouldered important projects for promoting the Indian culture. Today, in the contemporary society many festivities are not being celebrated the way they should be, the actual meaning behind the events are getting lost, their purpose getting defeated. Therefore, it needs to be protected from getting into the dormant state.

Natarajan Meghanathan et al. (Eds) : SAI, ICAITA, CSITA, ISPR, Signal - 2018
pp. 01–06, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81001

Deep learning is a division of machine learning which is stimulated by the functioning of the brain. It is considered to be most powerful and proficient model that performs outrageous even on laborious pattern recognition in vision and speech [2] [3]. Indian Government has tried to deal with this issue on different levels but they have failed to identify and preserve India's intangible cultural heritage. And so the need of the hour is to promote in such a way that every group, individual and institutions should get involved in preserving India's Cultural Heritage.

And realizing that, the Ministry of Culture has formed a Coordination Committee on the Living and Diverse Traditions of India to search out new ways to preserve India's insurmountable cultural heritage. The Committee is dealing with the issue nicely by constituting a subgroup whose purpose is to maintain a proper database, keeping records of various traditions of India. And to make sure that the information reaches a larger audience, it is recommending a digital presence of the same.

With this motivation, we hope to fulfill the gaps where the government is lagging on the 'Digital Promotion part' and contribute our bit to the nation so that it rises from the ashes like a phoenix and emerges as a 'GOLDEN BIRD' once again.

2. RELATED WORK

Previous work on cultural event recognition has been done by exploiting temporal model. Cultural events were detected and classified using visual features extracted from neural network with temporal statistics through a hierarchical classifier scheme in the Chalearn Challenge 2015. [1]

3. DATASET COLLECTION

Dataset collection is one of the most important phases. When deep learning approach is employed a strong and larger database is required for high performance. Dataset mostly compromises of images collected from Google. Dataset consists of images from 20 different festivals each consisting of more than 150 images.

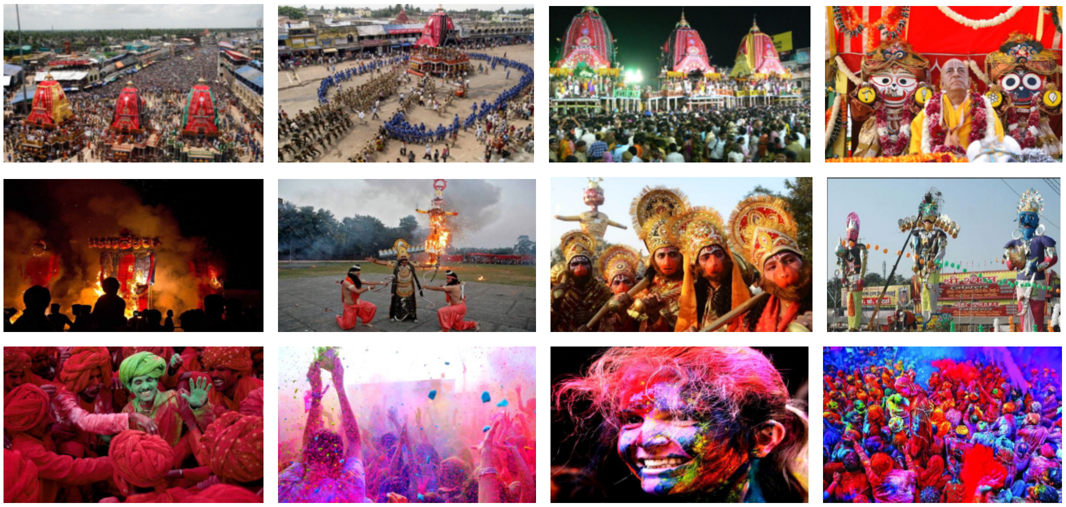


Figure 1. Images of different cultural events

4. METHODOLOGY

4.1. Support Vector Machines

SVM is a supervised learning model, which has recently come into concern in the field of machining learning and solving pattern recognition problem. Their basis is the ‘margin maximization principle’. They perform structural risk minimization, which improves the complexity of the classifier with the aim of achieving excellent generalization performance. In a higher dimensional space, SVM classifies by constructing a hyperplane which optimally segregates the data into two categories. Standard numeric techniques for QP has become impractical for very large datasets. SVM works well in high-dimensional spaces and gives excellent results in texture classification [4].

4.2. K-Nearest Neighbours

In case of lack of knowledge about the distribution of the data, this classification technique works very smoothly. For the classification, the Euclidean distance is calculated between the new instance and the stated training samples [5]. KNN editing has the endowment of removing noisy instances from the training set [6]. Predictions in KNN are directly made using the training data set only.

4.3. Random Forrest

Random forest is a classifier method used for classification which is fast to implement tree-structured predictors are the base constituent of the ensemble and is constructed using interpolation of randomness, therefore, termed as random forest[7]. Random Forest classifier creates many classification trees.

Many technical reports, Bremen (1996, 2000, 2001, 2004), demonstrated the steady gain in the classification and regression accuracy after using this classifier. It is basically used in unsupervised machine learning, used to describe unlabeled data.

4.4. Histogram of Oriented Gradients

HOG is an image descriptor which is used for object detection. The fundamental concept behind HOG is to describe the properties of the image like appearance, shape, color, etc. by distributing the intensity gradients properly. It divides the image into a small connected region called cells and a histogram of gradient direction is compiled in the pixel in each cell.

4.5. Local Binary Patterns

LBP is a type of image descriptor which is used for classifying images in computer vision. Though simple yet is very effective texture operator. It classifies the image by thresh holding each pixel’s neighborhood and a result is a binary number. In various applications, this texture operator has become popular because of its computational simplicity and high discriminating power, which makes it possible to analyze images in challenging real-time settings.

4.6. Gist

In recent times, the GIST descriptor received a lot of demand when it comes to scene recognition. It is a popular approach for web-scale image indexing [8]. It is based on Spatial Envelope, which does not require any type of segmentation. And it has also produced significant results in case of image search. GIST is primarily used to retrieve set of images of similar landmarks, for example, Taj Mahal. And the image point based matching technique refines the result and builds a 3D model of the same. They define the features that separate a scene from the rest.

4.7. Convolutional Neural Network (CNN)

CNN is a class of machine learning that is composed of multiple processing layers .it uses feed-forward artificial neural network or backpropagation algorithm. The first convolutional network was presented in 1989 [9] [10].

The primary eminence of CNN over other classifiers is that it requires little pre-processing as compared to others which have brought further enrichment in processing images, video, speech, and audio.

5. RESULT AND DISCUSSION

5.1. Experimental Scenario 1

In this scenario, we calculate the basic features by using HOG, LBP and GIST feature separately with SVM, KNN, and Random Forest classifier. The Purpose of these computations is to find out the best distinguishing feature that would later add up to our proposed methodology CNN. (Table 1)

5.2. Experimental Scenario 2

We proceed towards CNN, where we first calculate the accuracy of CNN alone with the SVM, KNN, Random Forest classifiers separately. The Purpose is to identify the best classifier to be used for the cultural events along with CNN. (Table 2)

5.3. Our Proposed Approach

Our proposed approach uses the pre-trained Alex net architecture. As per the need of CNN, pre-processing of images is done afterwards. For the feature extraction, Alex net is used In order to make computations less expensive, we have used two layers (FC6 AND FC7) of the Alex net architecture. The mentioned two layers are fully connected layers out of the total of 8layers. 5 layers are convolutional layers. (Table 3)

Table 1. Experimental Scenario 1

Features/Models	**SVM**	**KNN**	**Random Forrest**
HOG	22.5476	16.4598	14.5698
LBP	25.3658	17.5486	18.3569
GIST	22.5789	20.6598	18.2578

Table 2. Experimental Scenario 2

Features/Models	**SVM**	**KNN**	**Random Forrest**
CNN	60.5486	55.6587	52.3698

Table 3. Experimental Scenario 3

Features/Models	HOG	LBP	GIST
CNN+SVM	65.5679	77.7200	62.3569

Out of all experimental scenarios, CNN+SVM+LBP gave the best result with an accuracy of 77.72%.

6. CONCLUSION

In this paper, we employed a deep learning technique for the classification of different cultural events in their respective domains with a comparable accuracy of 77.72%. Indeed, this paper is a contribution to the project lead by Indian government “parampara” for the digital promotion of Indian cultural events.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to our supervisors Dr. Himanshu Chauhan (HOD CS Department), Ms. Supriya Shukla (Assistant Professor CSE Department), and Mr. Neeraj Kr Pandey (Assistant Professor CSE Department) for providing their invaluable guidance, comments, and suggestions throughout the course of the project. We would specially thank Mrs. Supriya Shukla for constantly motivating us to work harder.

REFERENCES

[1] Salvador, Amaia, et al. "Cultural event recognition with visual convnets and temporal models." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2015.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[3] Hinton, Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." IEEE Signal Processing Magazine 29.6 (2012): 82-97.

[4] Kim, Kwang In, et al. "Support vector machines for texture classification." IEEE transactions on pattern analysis and machine intelligence 24.11 (2002): 1542-1550.

[5] Peterson, Leif E. "K-nearest neighbor." Scholarpedia 4.2 (2009): 1883.

[6] Wilson, D. L. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions on Systems, Man, and Communications 2, 3 (1972), 408–421.

[7] G´erardBiau, “Analysis of a Random Forests Model “, Journal of Machine Learning Research 13 (2012)

[8] Douze, Matthijs, et al. "Evaluation of gist descriptors for web-scale image search." Proceedings of the ACM International Conference on Image and Video Retrieval. ACM, 2009.

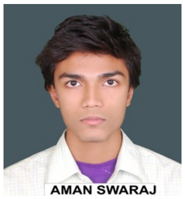
[9] Guan, Donghai, et al. "Nearest neighbor editing aided by unlabeled data." Information Sciences 179.13 (2009): 2273-2282.

[10] https://en.wikipedia.org/wiki/Convolutional_neural_network

AUTHORS

Aman Swaraj

B.TECH -IV, Department of Computer Science and Engineering, College of Engineering Roorkee, amanswaraj007@gmail.com.



Harshita Sahni

B.TECH -IV, Department of Computer Science and Engineering, College of Engineering Roorkee, harshitasahni777@gmail.com.

Archit Agarwal

B.TECH -IV, Department of Information Technology, College of Engineering Roorkee, aavicky995@gmail.com.

Supriya Shukla

Assistant Professor, Department of Computer Science and Engineering, College of Engineering, Roorkee, supriyaluthra@yahoo.co.in

A PROBABILISTIC APPROACH FOR DETECTING SPEECH FILE

Punnoose A K

Flare Speech Systems, India

ABSTRACT

This paper discuss an approach to detect whether a wave file contains speech or not. A frame classifier is trained to classify frames to phones. The inherent biases of the frame classifier, in terms of various aspects of recognition, is captured in terms of probability distributions. Using the distributions of speech and noise, an approach is presented, which scores wave file for the presence or absence of speech. Relevant databases are used to test the detection rate of this approach.

KEYWORDS

Noise Robustness, Neural Networks, Interactive Voice Response Systems, Confidence Scoring

1. INTRODUCTION

In most speech recognition based interactive voice response system(IVRS), a pre-processing step is needed which tells whether a file contains speech or not. A misrecognition in one of the steps could prompt the dialogue manager, which directs the dialogue, to take undesirable paths through the dialog tree. Mostly signal processing based approaches are used to detect the level of noise or speech in a wave file. A major drawback with signal processing based approaches is that, it often makes assumptions about the noise, which is generally not practical.

One such assumption is the stationarity of noise, which assumes that the spectrum of noise is relatively same across time. This allows spectral subtraction to be employed. But in reality, real-world noise conditions seldom follow stationarity in spectrum. In fact real-world noise will be anything but being stationary. Moreover many phones has a lot of similarity with noise, spectrum wise, which will make spectral subtraction difficult.

Another approach is model the speech, rather than noise. As the spectral variations in speech will be limited and more contained as compared to that of noise which could be very broad, it will be easy to model the aspects of speech such as harmonicity, pitch, etc so that differentiation between speech and noise is easier. But a lot of noise types are also harmonic, which will cause difficulties in discriminating speech and noise eventually.

In terms of application, a dialogue manager will have the information regarding what type of confidence scoring for speech, to be employed, depending upon the node. A node in a dialog path

Natarajan Meghanathan et al. (Eds) : SAI, ICAITA, CSITA, ISPR, Signal - 2018
pp. 07–18, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81002

is a system prompt followed by a user utterance. If the dialogue nodes corresponds to a confirmation, where a false positive will be too expensive, the wave file can only be passed to the speech recognition engine, once the there is enough confidence that the file contains speech.

On the other hand if the dialogue node involves the recognition of a word from a list, then skipping the preprocessing step may be prefered, thus allowing the speech recognition engine to output a hypothesis, either frame wise or phone wise or word wise, depending upon the engine. Now using a mathematical model to suggest how a phone might get affected by the presence of noise, some recovery is possible.

In critical applications such as banking, not even a single false positives can be afforded, even at the expense of missing some of the genuine speech files. In such cases, a pre-processing step before passing the wave file to a speech recognition engine is very much necessary. This paper captures the biases of a frame classifier, for noise and speech, and presents a couple of probabilistic models to score the presence of speech in a wave file.

2. PROBLEM DEFINITION

Given the frame classifier output of a wave file, which is a sequence of phones, each corresponding to a frame, derive a confidence score which can indicate whether a file contains speech or not.

3. PRIOR WORK

In [1], author discuss an approach using a set of temporal and spectral features to segment the videos into speech and non speech. Author uses features like Low short-time energy ratio,high zero-crossing rate ratio, Line Spectral Pairs, Spectral centroid, Spectral Roll-off, Spectral Flux, etc. Classifiers are trained to predict whether a segment is speech or non-speech. In [2], authors use a neural network for learning the phone durations. The input features are derived from the phone identities of a context window of phones, along with the durations of preceding phones within that window.

In [3], authors discuss about a noise robust Voice Activity Detection(VAD) system, utilizing periodicity of signal, full band energy and ratio of high to low band signal energy. Voice regions of speech are identified and then proceeds to differentiate unvoiced regions from silence and background noise using energy ratio and energy of total signal. In [4], authors present spectral feature for detecting the presence of spoken speech in presence of mixed signal. The feature is based on the presence of a trajectory of harmonics, in speech signal. The property that, speech harmonics cover multiple frames in time, is treated as a feature.

In [5], authors use harmonics, pitch and subband energy to locate the speech and track the time-varying noise. Pitch measurements are used to detect the vowel segments. Subbands are divided based on energy and frequency and based on predetermined thresholds from determinate noise, voiced parts of potential voice regions, are identified.

4. APPROACH OUTLINE

First a neural network is trained to classify frames to phones. Frames correspond to the usual 25ms of time with a 15ms overlap between successive frames. Context independent phones are used as the labels. Phones are preferred as labels as opposed to subphones. This is because a subphone based forced aligner doesn't align the boundaries well, thus affecting the quality of frame classifier. Assuming a decent level of accuracy, we capture the inherent classification biases of the frame classifier, in terms of phone duration, and in the distribution of softmax probabilities, for noise and speech separately in probability distributions.

Probability Distributions on phone chunk durations and softmax probabilities are defined, for noise and speech. Simple rules are derived from these distributions, to classify files into speech/noise. The rules are made to decrease the false positives as much as possible at the expense of false negatives.

5. DETAILED APPROACH & ANALYSIS

A multi layer perceptron(MLP) is trained to predict phones, with softmax layer at the ouput. For a given feature frame at the input, the MLP outputs a probability vector. The phone which has the maximum value in the probability vector is treated as the detected phone. The classified phone for a frame is also termed the top phone for that frame. A set of continuous frames with the same phone detected is regarded as a phone chunk. Size of the phone chunk is the number of frames in the phone chunk.

Common Notations:

- q : A phone in the phone set, Q.
- q_j : Phone chunk q of size j
- N: denotes noise.
- S: denotes speech.
- $C_N(q_j)$: Count of phone chunk q, of size j, in noise.
- $C_S(q_j)$: Count of phone chunk q, of size j, in speech.
- M: maximum chunk size.

Distribution on Phone Chunk Sizes: Fig 1 and Fig 2 plots the count of the phone /b/ for speech and noise respectively. It is clear from the plots that for noise data, chunks with higher duration are totally absent. This means that phone /b/ is resilient to the presence of noise. This motivates us to make a probability distribution on the phone chunk width, to discriminate between speech and noise.

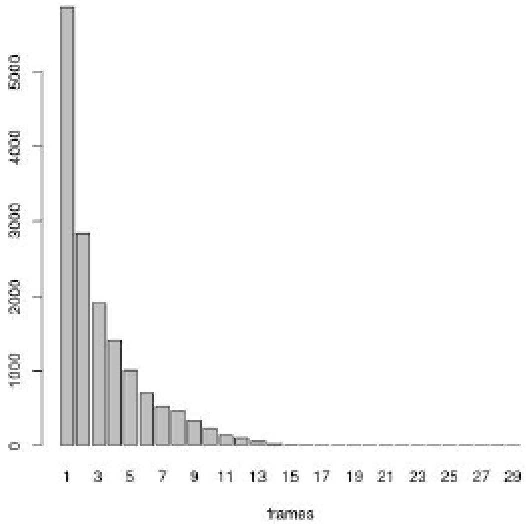


Fig. 1. Phone /b/ : Chunk width vs Count: Speech Data

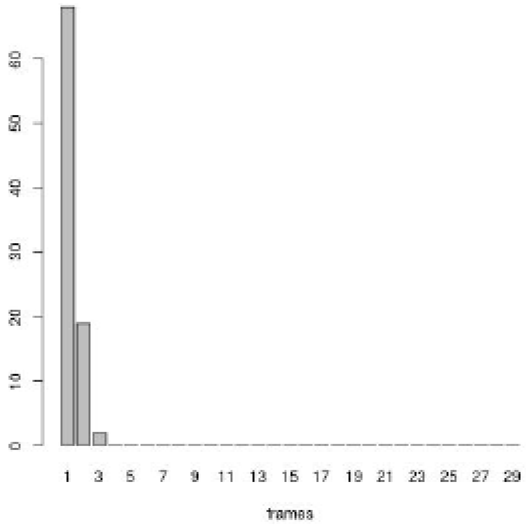


Fig. 2. Phone /b/ : Chunk width vs Count: Noise Data

Define $P_1(q_j \mid N)$ and $P_1(q_j \mid S)$ which is a probability distribution on phone chunk size, for noise and speech respectively. $P_1(q_j \mid N)$ is the probability of phone q of chunk size j, occuring in noise data.

$$P_1(q_j|N) = \frac{\sum_{j=1}^{M} C_N(q_j)}{\sum_q \sum_{j=1}^{M} C_N(q_j)} \qquad (1)$$

where, $P_1(q|N)$, is the probability of finding chunks, be whatever size, of phone q, given N. $P_1(q_j \mid q;N)$, is the probability of finding a chunk of size j, given the phone is q, in the noise data N.

Distribution on Softmax Probabilities: Fig 3 and Fig 4 plots the histogram for the phone /f/, for noise and speech data. Note that these are the instances where the frame is classified as /f/ phone. ie, /f/ is the top phone for that frame. So the data plotted here is the maximum probability of all the phones. It is clear the difference between the probabilities for noise and speech. For speech the probability is concentrated at the right end, while for noise, it is focused more around the middle.

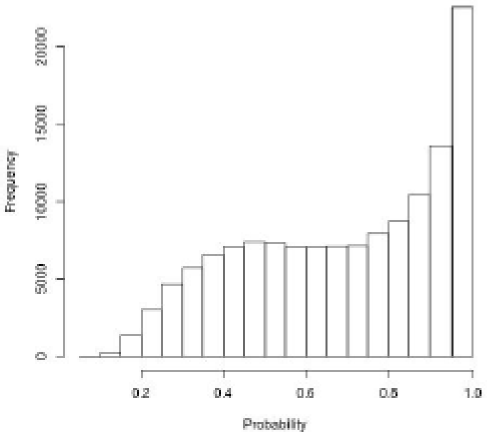


Fig. 3. Histogram of softmax probabilities of /f/, for clean data

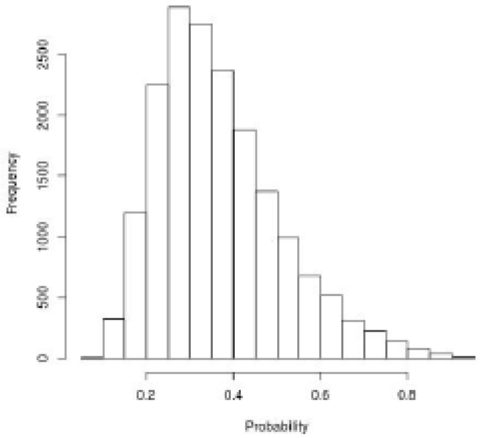


Fig. 4. Histogram of softmax probabilities of /f/, for noisy data

This serves as a valid feature to discriminate frames of speech from noise. We construct the second probability distribution on this data. Denoting p as the softmax probability of the phone, $b(p)$ gives the probability bin of p, and $C(b(p))$ is the count of instances in that probability bin.

$$P(p|N;q) \quad = \frac{C_N(b(p))}{\sum_b C_N(b)}$$

Denoting the probability of noise, given the softmax probability of the top phone q, as P_2(N|(p,q)), and by using Bayes theorem,

$$P_2(N|\langle p,q\rangle) \quad = \frac{P(p|N;q)P(N)}{P(p|N;q)P(N)+P(p|S;q)P(S)} \qquad (2)$$

A. *Using the Distributions*

In equation (1) and (2), distributions are defined on phone chunk level. To make predictions in a file level, we need $P_1(N|\text{wavefile})$ and $P_2(N|\text{wavefile})$. ie, distributions defined at the file level.

1) File Level Phone Chunk Distribution :

Let $[q_j^i]$ be the phone chunk sequence for a wave file, where the superscript i is the index and j is the chunk length. Each of $q^i \in Q$, where $1 \leq i \leq X$, where X is the number of phone chunks in the wave file. Assuming each phone chunk to be independent, the probability of the wave file being noise, can be interpreted as the probability of each chunk in the chunk sequence being noise. The posterior probability can be written as,

$$P_1(N|\text{wavefile}) = P_1(N|q_{j_1}^1 \wedge N|q_{j_2}^2 \wedge ..N|q_{j_X}^X) \qquad (3)$$

where $q_{j_i}^i$ is the i th chunk in the chunk sequence with the length j_i. By the independence of phone chunks

$$P_1(N|q_{j_1}^1 \wedge N|q_{j_2}^2 \wedge ..N|q_{j_X}^X) = \prod_{i=1}^{X} P_1(N|q_{j_i}^i) \qquad (4)$$

where

$$P_1(N|q_{j_i}^i) \quad = \frac{P_1(q_{j_i}^i|N)P(N)}{P_1(q_{j_i}^i|N)P(N)+P_1(q_{j_i}^i|S)P(S)}$$

by Bayes theorem. $P(S)$ and $P(N)$ are the prior probability of speech and noise respectively.

2) *File Level Softmax Probability Distribution:*

Denote the softmax probability and the associated top phone by $[\langle p^i, q^i\rangle]$, where $1 \leq i \leq Z$. Note that top phones can occur intermittently or continuously. Z is the total number of the top phones in the wave file, which is the same as the number of frames in the file. Assuming Z top phones are seen, the probability that the file is noisy is given by,

$$P_2(N|\langle p^1 q^1\rangle \wedge ... \wedge N|\langle p^Z q^Z\rangle) = \prod_{i=1}^{Z} P_2(N|\langle p^i q^i\rangle) \qquad (5)$$

6. EXPERIMENTAL DETAILS & RESULTS

Experimentation is broadly divided into three stages.

1) Train a frame classifier to predict a frame into one of the phones.

2) Using the frame classifier, model the conditional distributions on phone chunk size and softmax probability of top phone, for speech and noise data.

3) Use the distributions for testing speech and noise files to see whether they can be discriminated.

A. Dataset Details

Voxforge dataset is used as the speech data and CHiME dataset is used as the noise data.

Rationale for Voxforge Data: The foremost reason for using Voxforge data is that, it is recorded in an uncontrolled environment by different people with different accent, with different mother tongue, etc. This will give the necessary variability in the data, which is very much crucial for making a speaker independent speech recognition system. This is very much against the popular notion of using a very well known database like TIMIT, which is recorded in a controlled environment, as the focus here is on real world IVRS, where the user response is simply silence or background speech, or just murmuring, or traffic noise, or noise of any other kind. A rough approximation of analyzing a real world speech based information access system will show that roughly only 20% of the user utterance is of any significant speech content. This heavily bias us to use a speech database which is uncontrolled and with wide variability.

Frame Classifier Details: A MLP is trained to predict phones from speech features. Perceptual Linear PredictionCoefficients(plp) is used as feature. plp along with delta and double delta coefficients are used as the feature. Standard 41 phone set of English is used as the labels. Mini batch gradient descent is used as the training mechanism. Cross Entropy Error is used as the measure for backpropagation training. 3 hidden layers are used and weights of MLP are initialized randomly between -1 and +1. Softmax layer is used in the output layer which outputs a probability vector, given a plp frame as input.

Noise Data Details: Pure background noise from CHiME4 Dataset is used as noise data. Background noise in various environment like street, bus, etc are used. Unlike older CHiME datasets, CHiME4 is not segregated based on SNR. CHiME data is divided into 2 subsets and used in the second and third stages.

We present the results for both distributions, independently, to figure out how speech files can be separated from the noisy ones. It is to be noted that for all the three stages discussed above, three different dataset is used. For all the stages for speech, 3 different subset of Voxforge data is employed. For stage 2 and 3, for noise data, different subset of CHiME data is used.

Conditional distributions $P_1(N|q_j)$, $P_1(S|q_j)$, $P_2(N|(p^i q^i))$ and $P_2(S|(p^i q^i))$ are learned in the second stage and the posterior probabilities $P_1(N|\text{wavefile})$ and $P_1(S|\text{wavefile})$ are calculated in the testing stage. With a focus on precision results are given for true positives and false positives, for both approaches.

B. Phone Chunk Size Distribution Results

As our aim is to discriminate speech and noise files, Equation (4) can be rewritten as,

$$P_1(N|q^1_{j_1} \wedge N|q^2_{j_2} \wedge ..N|q^X_{j_X}) \propto \tfrac{1}{X}\textstyle\sum_{i=1}^{X} ln(P_1(N|q^i_{j_1}))$$

By (3), the posterior can be written as

$$P_1(N|\text{wavefile}) \propto \frac{1}{X}\sum_{i=1}^{X} ln(P_1(N|q^i_{j_1})) \qquad (6)$$

This is mainly done to avoid the underflow, while using equation (2). And the results are averaged, to make sure the same scale for otherwise longer files. Phones whose counts falls below a threshold, in the calculation of the conditional densities are excluded from the analysis.

Fig 5, plots the results of posterior probabilities, given the speech data. The posteriors are approximated using (7). The posteriors from each speech wavefile is plotted as histograms. Green histogram represents the $P_1(S|\text{speech})$ and blue histogram represents the $P_1(N|\text{speech})$. It is evident from the plot that both the posteriors are clearly separated, given the input speech data.

As the plots are in log scale, the values closer to 0 means more probable. For speech data, it is seen from the plot that the green histogram which is the speech posterior is closer to the 0, than the noise posterior. Also it is evident from the plot that the posterior of noise for speech data is very wide spread than compared to that of speech posterior, which is narrowly concentrated.

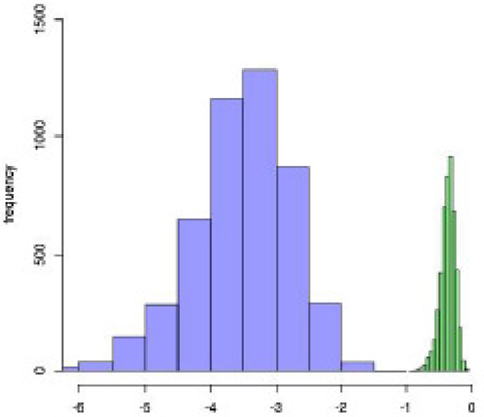


Fig. 5. Speech and Noise Posterior for Speech Data

Fig 6, plots histogram of speech and noise posteriors, given the noise data as input. Green represents the $P_1(S|\text{noise})$, and blue the $P_1(N|\text{noise})$. Both of the histograms are evenly spread in the log domain.

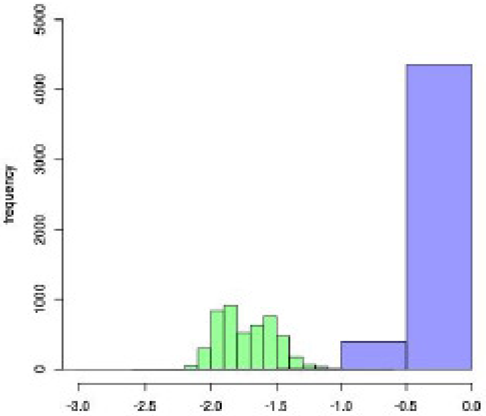


Fig. 6. Speech and Noise Posterior for Noise Data

As the focus is more on speech file detection, it is worth looking at the false positives and true positives. Fig 7, plots the $P_1(S|\text{noise})$ as blue and $P_1(S|\text{speech})$ as green histogram. It is clear that using appropriate threshold on average log posterior values, the speech and noise can be easily separated.

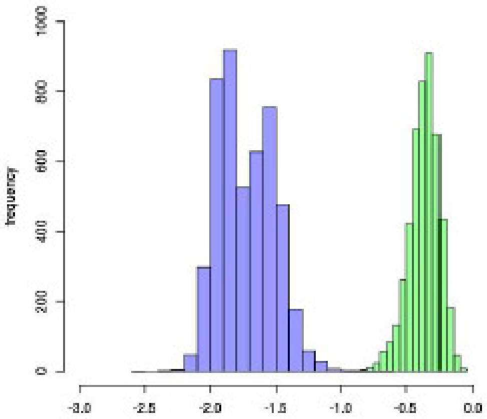


Fig. 7. Speech Posteriors for Noise and Speech Data

Table 1. Results

Threshold τ	True Positives	False Positives
> -0.9	4769	2
> -1	4770	4
> -1.1	4770	10
> -1.2	4770	39

Table 1 shows the true positives and the false positives for different threshold values of average log posteriors.

C. *Softmax Probability Distribution Results*

As in (7), instead of the product of probabilities, we approximate it using the average posterior probability of top phones, for a wave file. ie,

$$P_2(N|\text{wavefile}) \propto \frac{1}{Z}\sum_{i=1}^{Z} ln(P_2(N|\langle p^i, q^i\rangle))$$

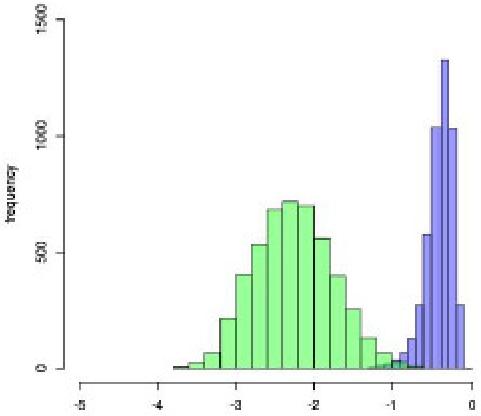


Fig. 8. Speech and Noise Posteriors for Speech Data

Fig 8, plots the results of posterior probabilities, given the speech data. The posteriors are approximated using (7). Blue histogram represents the P_2(S|speech) and green histogram represents the P_2(N|speech).

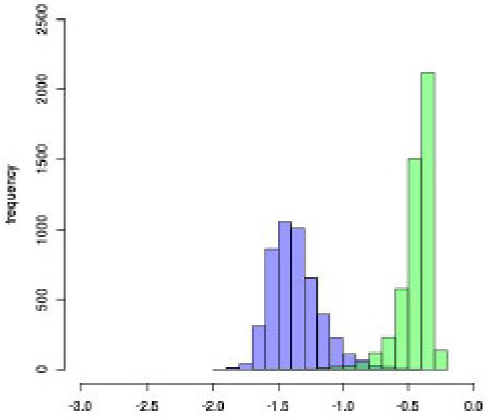


Fig. 9. Speech and Noise Posteriors for Noise Data

Fig 9, plots histogram of speech and noise posteriors, given the noise data as input. Green represents the P_2(*N*|noise), and blue the P_2(*S*|noise).

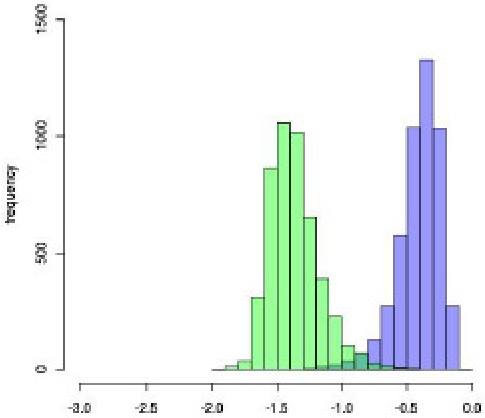
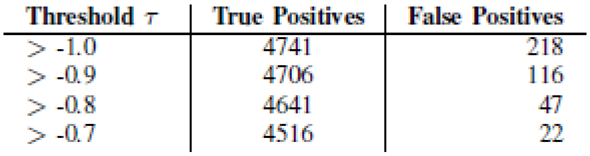


Fig. 10. Speech Posteriors for Speech and Noise Data

Focusing more on true positives and false positives, Fig 10, plots the $P_2(S|\text{speech})$ as blue and $P_2(S|\text{noise})$ as green histogram.

Table 2. Results



Threshold τ	True Positives	False Positives
$>$ -1.0	4741	218
$>$ -0.9	4706	116
$>$ -0.8	4641	47
$>$ -0.7	4516	22

Table 2, shows the true positives and false positives for different value of threshold posteriors.

7. Conclusion and Future Work

A new approach for detecting whether a wave file consists of speech is presented. A frame classifier is first trained to predict the phone, given a frame. The characteristics of the frame classifier is codified with 2 probability distributions, one on phone chunk size, and another one on softmax probability associated with a phone, given a frame. Posterior distributions are approximated in log domain to reduce the dynamic range of scores. Results are shown separately, to show the effectiveness of both of the approach independently.

In future, we plan to use more spectrum derived features, in conjunction with frame level features to increase the overall accuracy of this approach. Spectrum level features provides vital clues, which could be missed by any parameterized features like mfcc or plp, especially for noise robustness.

References

[1] Ananya Misra, " NonSpeech Segmentation in Web Videos",

[2] Hossein Hadian, Daniel Povey, Hossein Sameti, Sanjeev Khudanpur, "Phone duration modeling for LVCSR using neural networks"

[3] E. Verteletskaya, K. Sakhnov, “Voice Activity Detection for Speech Enhancement Application”,

[4] Reinhard Sonnleitner, Bernhard Niedermayer, Gerhard Widmer, Jan Schluter, “A Simple and Effective Spectral Feature for Speech Detection in Mixed Audio Signal”,

[5] Zhihao Ahang and Jinlong Lin,“Robust Voice Activity detection Based on Pitch and Subband Energy”

A DEEP LEARNING APPROACH TO SPEECH BASED CONTROL OF UNMANNED AERIAL VEHICLES (UAVs)

Saumya Kumaar[1], Toshit Bazaz[2], Sumeet Kour[2], Disha Gupta[2],
Ravi M. Vishwanath[1] and S N Omkar[1]

[1]Indian Institute of Science, Bengaluru
[2]National Institute of Technology, Srinagar

ABSTRACT

Speech recognition has been one of the key research domains in computational signal processing. Despite high levels of computational complexity associated with achieving speech recognition in real-time, promising progress has been made under the umbrella of voice controlled robotics. This paper proposes an alternate approach to speech recognition for robotics applications, without adding on external hardware. We use a combination of spectrograms, MEL and MFCC features and a neural network based classification which is usually done offline, whereas the proposed method offers a remote real-time control of the robot that can be used to survey terrains that are otherwise impervious for humans, or monitor activities inside huge structures like wind-mills, gas pipelines etc. The trained model occupies lesser than 4MB on the storage medium of the platform and it also displays metrics of confidence and accuracy of prediction. The overall validation accuracy of the algorithm goes as high as 97% while the testing accuracy of the system is 95.4%. Since this is a classification algorithm, results have been presented on custom voice classification datasets.

KEYWORDS

Deep Learning, Signal Processing, Unmanned Aerial Vehicles, Speech Recognition

1. INTRODUCTION

Most speech recognition applications in robotics rely heavily on either hardware based systems (like VRBot, GeeTech, RKI-1199 etc.) or Googles Speech API. In both these cases, additional requirements come into picture in the form of extra hardware or the need for an internet connection.

Now most of the commercial/hobbyist robotic applications are built using System-On-Chip (SOCs) like Raspberry Pi, Odroid-XU4, Beaglebone etc. which run on Linux-based RTOS platforms and have reasonable computational capabilities. This paper proposes an alternate approach to speech recognition for robotics applications, without adding on external hardware. We use deep neural networks with only fully connected layers for recognizing different possible speech commands given to the drone, via spectrogram classification, in real time. Most of the

Natarajan Meghanathan et al. (Eds) : SAI, ICAITA, CSITA, ISPR, Signal - 2018
pp. 19–30, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81003

research done in spectrogram and other features based classification is usually done offline, whereas the proposed method offers a remote real-time control of the robot that can be used to survey terrains that are otherwise impervious for humans, or monitor activities inside huge structures like wind-mills, gas pipelines etc.

The primary contributions of this paper are listed below :

- In this study, we worked with 8 control commands. Histogram equalization was applied to the spectrograms before feeding them to the network in order to enhance features for the network to learn. Since only 8 words are taken into consideration, speech recognition problem turns into a simple 8-class classification problem.

- A novel deep net architecture with a very small memory footprint, which further gives decent classification accuracy on custom voice/speech dataset.

2. RELATED WORK

Beard *et al.* [1] have created several alternative UAV interfaces in which users operate physical controllers to generate the requisite numerical commands. These interfaces are built using PDAs, full-size computers, a voice-recognition system, a force-feedback attitude joystick, a force-sensing interface using an IBM TrackPointTM, and a novel physical icon interaction scheme. Real-world tests with this interface have demonstrated that ambient wind noise and conversation can wreak havoc on the reliability of the voice-recognition system. A method of muting the microphone input is required, but even with 2461 such a system in place, considerable difficulties arise in environments with strong winds or loud background noises. However, our experience bas shown the voice interface to be very valuable, especially under favorable weather conditions.

UAV control stations feature multiple menu pages with systems accessed by keyboard presses as presented by Draper *et al.* [2]. Use of speech-based input may enable operators to navigate through menus and select options more quickly. This experiment processed and presented the utility of conventional manual input against the speech input for actions performed by UAV operators on the control station at two different levels of mission difficulty. Pilots performed a continuous flight/navigation control job while keeping in mind to complete eight different data input/entry tasks types with each input modality. Results from the experiment have proven that speech input and speech recognition based control was significantly better than manual input or RC control in terms of task completion time, task accuracy, flight/navigation measures, and pilot ratings. Across all the given tasks, data entry time was drastically reduced by approximately 40% with speech input.

The AirSTAR testbed developed by Jordan *et. al* [3] has been developed to provide an in-flight capability to validate various flight critical technologies. The testbed is composed of three elements: a 5.5% dynamically scaled, turbine powered generic transport model (GTM), a Mobile Operations Station (MOS) and associated ground based facilities, and a test range. This research capability, along with wind tunnel testing, full scale flight testing, and flight simulation, provides the methods and tools to develop and test the technologies demanded by the AvSP. The expanded flight envelope of the GTM and the requirements to gather large amounts of data (at high rates) presented unique challenges to the development of the AirSTAR testbed. Because the GTM will be operating outside of the normal benign flight envelope of full scale transport aircraft and most

UAVs, additional measures had to be taken, both on the plane and in the control station, to mitigate the risks associated with this type of flight.

McLain *et. al* [4] UAV research interests have been revolving around cooperative and coordinated control of multiple vehicles and real-time trajectory generation and optimization. Their primary objectives for experimental testing of their research are to validate the feasibility of practical implementation of their methods and to foster innovation to overcome implementation challenges. For the control of UAVs, real-world issues such as sensor noise, communication dropout, communication delay, and computation latency can degrade performance and lead to catastrophic failures. Sensors that are inherently asynchronous with varied sample rates can pose challenges for estimation and coordination. Airframe payload capacity influences the choice of sensors and onboard computers and thus the inherent capabilities of the vehicle. Environmental factors, such as wind, weather, and lighting can adversely affect sensor and control system performance. Field tests often expose the unanticipated challenges that must be dealt with in a real-world scenario. Furthermore, these challenges often force significant innovations to occur to enable success.

Prodeus*et. al* [5] compared ix noise reduction algorithms with the use of a set of indicators. Among them are popular noise reduction algorithms such as spectral subtraction, Wiener filtering, MMSE and logMMSE, and two less well-known Wiener-TSNR and Wiener-HRNR algorithms. It has been proven that when the noise reduction system is used as preprocessing or of automatic/autonomous speech recognition (ASR) system, only a small amount of speech signal quality indicators is in significant consensus with the recognition accuracy or classification rate. In specific, these include Log-Likelihood Ratio (LLR) and Signal Composite Index (SCI) indicators. Furthermore, no single algorithm amongst al of the considered noise reduction algorithms, is the top-most in terms of maximum recognition rate for a very huge variety range of input signal-to-noise ratio all ranging from -10 dB to +30 dB.

They reviewed the theory of discrete Markov chains and showed how the concept of hidden states, can be effectively used. They illustrate the theory with two simple examples, namely coin-tossing, and the classic balls-in-urns system. They discuss the three fundamental problems of HMMs, and give several practical techniques for solving these problems. They also discussed the various types of HMMs that have been studied including ergodic as well as left-right models. They discussed state density function, onservation duration density, and optimization criterion for choosing optimal HMM parameter values. They also discuss the issues that arise in implementing HMMs including the topics of scaling, initial parameter estimates, model size, model form, missing data, and multiple observation sequences. They described an isolated word speech recognizer, that was implemented with HMM. They extend the ideas presented before to the problem of recognizing a string of spoken words based on concatenating individual HMMs of each word in the vocabulary. They briefly outlined how the large vocabulary speech recognizer use ideas of HMM.

A database as well as a recognition experiment was presented in this paper by Hirsch *et. al* [7] to obtain comparable recognition results for the speaker-independent recognition of connected words in the presence of additive background noise and for the combination of additive and convolutional distortion. The distortions are artificially added to the clean TIDigits database. The noisy database together with the definition of training and test sets can be taken to determine the performance of a complete recognition system. In combination with a predefined set-up of a

HTK(Hidden Markov Model Tool Kit) based recognizer it can be taken to evaluate the performance of a feature extraction scheme only.

Hinton *et al.* [8] reviewed exploratory experiments on TIMIT database and used them to demonstrate the power of a two-stage training procedure for acoustic modeling. The DNNs that worked well on TIMIT database were then applied to five different large-vocabulary continuous speech recognition tasks. Their DNNs worked well on all the tasks and on some the tasks it outperformed the state of the art.

According to Graves *et. al* [9], it is possible to train RNNs end-to-end for speech recognition. This approach exploits the larger state-space and richer dynamics of RNNs compared to HMMs, and avoids the problem of using potentially incorrect alignments as training targets. The question that inspired their paper was whether RNNs could benefit from depth in space.

In this paper by Itakura *et. al* [10], an approach to the problem was described from a statistical point of view, and it was shown that the log likelihood ratio, which is the best criterion to test the hypothesis, was reduced to the logarithm of the ratio of prediction residuals, and can be used as a powerful distance measure. This result of their research was applied to automatic recognition of isolated words, where the sequential likelihood ratio test was adopted to reduce the amount of computation.

3. METHODOLOGY

The system was trained on the features of the voice samples (MEL and MFCC) and corresponding spectrograms of 15 subjects from 19-22 years of age speaking 8 different words that were Takeoff, Land, Forward, Backward, Left, Right, Up & Down in 10 different pitches. Among these 15 subjects 9 were male and 6 were female.

An open-source code was used to collect the voice samples and at the same time to create spectrograms corresponding to each sample and then all samples were subjected to 9 pitch variation. The voice samples were recorded in random order, and there was a 5s hint before each sample was collected to tell the subject which word to say. Among the recorded samples, only the samples with noise below a particular level were used.

Then MEL and MFCC features were extracted from these voice samples and a batch generator was used to extract all 1200 samples at a time. These 1200 samples were split into training and test sets. The training and test sets consist of labels of voice samples, the spectrograms and the MEL/MFCC features corresponding to each voice sample.

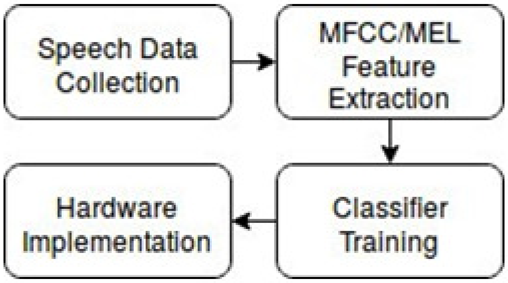


Figure 1 : Flow of the Algorithm

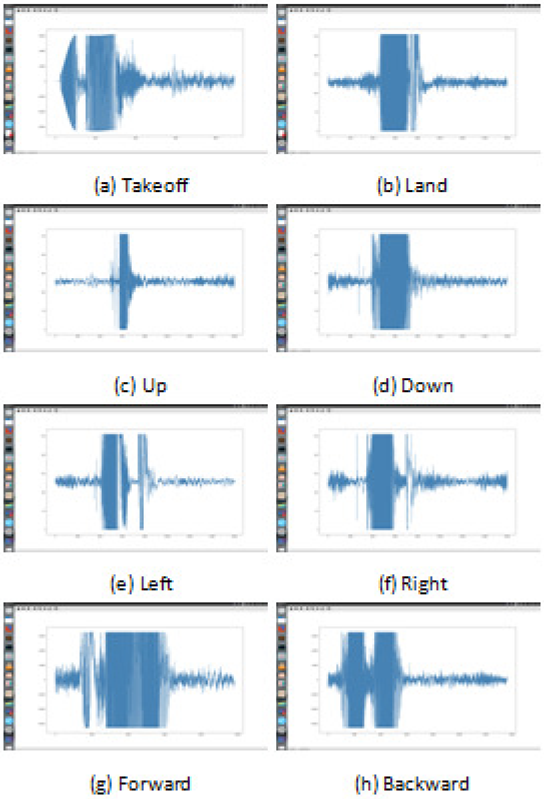
3.1. Dataset

The dataset (Fig. 2) consists of a total of 1200 recorded voice samples and 1200 spectrograms of the subjects from 19-22 years of age speaking 8 different words in 10 different pitches. The words were Takeoff, Land, Forward, Backward, Left, Right, Up & Down. The words were marked with numbers from ranging from 0-7 (0-Takeoff, 1-Land, 2-Forward, 3-Backward, 4-Up, 5-Down, 6-Left & 7-Right). The above mentioned words were chosen specifically for UAV control because UAVs or drones can execute these commands only, so we do not need an extensive speech recognition system for controlling a robot.

A total of almost 3000 voice samples were recorded among which 1200 were marked as correct (having noise below the particular level. All the voice samples recorded were in English and each of the recorded voice samples last for a period of 5 seconds. Sample Dataset is shown in the figure below.

3.2. Feature Extraction

For our research we observed that MFCC and MEL feautre sets to be appropriate for speech classification. Also, spectrograms have been made. The extraction methods are explained as follows.



(a) Takeoff
(b) Land
(c) Up
(d) Down
(e) Left
(f) Right
(g) Forward
(h) Backward

Figure 2.: Sample Data Audio Plots

3.2.1. The Spectrogram

If x is signal of length N, and considering consecutive clips of x of length m where m <<n and let X ∈ $R^{m(Nm+1)}$ be the matrix with the consecutive segments as consecutive columns. In other words, $[x[0],x[1],...,x[m1]]^T$ is the first column, $[x[1],x[2],...,x[m]]^T$ is the second column, and so forth. Both the rows and columns of X are indexed by time. We see that X is a not a mathematically useful representation of x, whose columns are the Discrete Fourier Transforms of the columns

$$\hat{X} = \overline{F}X \tag{1}$$

$$X = \frac{1}{m}F\hat{X} \tag{2}$$

The spectrogram of x with window size m is the matrix $\hat{X}$ are indexed by frequency and the columns are indexed by time. Each location on $\hat{X}$ Note that the rows of $\hat{X}$ corresponds to a point in frequency and time. So $\hat{X}$^ is a mixed time-frequency representation of x. Because the conversion and transformation between X and $\hat{X}$ is also highly redundant.

The spectrogram is a matrix. To visualize it we can view the matrix as an image with the *i*, *j–th* entry in the matrix corresponding to the intensity or color of the *i*, *j–th* pixel in the image.

The spectrograms of various voice samples have been plotted and shown (Fig. 3) with post histogram equalization. Histogram equalization has been done to enhance the features (contrast basically) in the spectrograms.

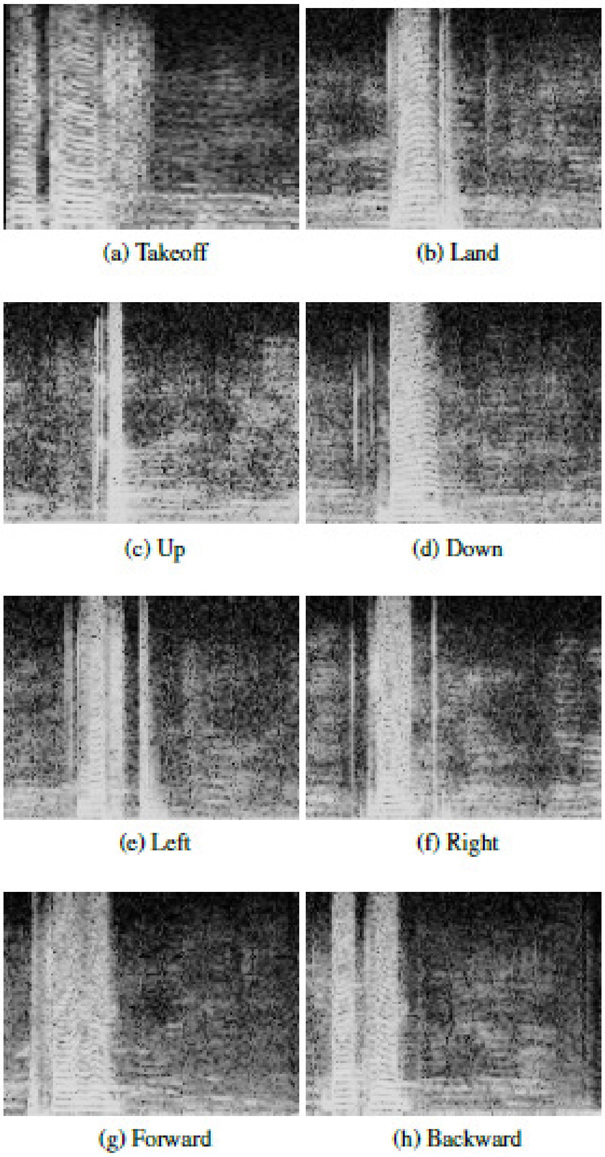
3.2.2. MEL Frequency Cepstral Coefficients (MFCC)

The implementation of Mel Frequency Cepstral Coefficients is one of the standard benchmarked method for audio/speech-based feature extraction. There are about 20 coefficients in ASR, although speech encoding could be probably achieved with the help of only 12-13 coefficients. However, a disadvantage of using MFCC features is it's sensitivity to noise due to its' dependence on spectral form. It is therefore recommended to use techniques that extract information from the periodicity of speech data, which could be used to overcome the above mentioned problem, although human speech may also contain aperiodic content.

As an approximation to Mel-frequency scale, the frequency scale that is used here is approximately linear for frequencies below the range of 1 kHz and logarithmic for frequencies higher than 1 kHz. The motivation for this approximation comes from the fact that the human auditory sensory system is comparatively less frequency-selective as frequency increases beyond 1 kHz. The MFCC features correspond to the cepstrum of the log filterbank energies. To calculate them, the log energy is first computed from the filter bank outputs as

$$S_t[m] = \log\left(\sum_{n=0}^{N-1} |X_t[n]|^2 H_m[n]\right) \quad 0 \le m < M \tag{3}$$

where $Xt[n]$ is the DFT of the t^{th} input speech frame, $H_m[n]$ is the frequency response of m^{th} filter in the filterbank, N is the transformation window size and M is the total number of filters. Then, the discrete cosine transform (DCT) of the log energies is computed as follows :


(a) Takeoff

(b) Land

(c) Up

(d) Down

(e) Left

(f) Right

(g) Forward

(h) Backward

Figure 3 : Spectrograms of the Voice Samples

$$c_t[m] = \sum_{n=0}^{N-1} S_t[n] \cos\left(\pi m \left(\frac{n-0.5}{M}\right)\right) \quad 0 \leq m < M \tag{4}$$

Since the human auditory system is dependent on time based evolution of the spectral content of the signal, attempts are often made to include the extraction of this data as part of MFCC feature analysis. In order to capture the changes in the coefficients over time, first and second difference coefficients are computed as respectively.

$$\Delta\vec{c_t} = \vec{c}_{t+2} - \vec{c}_{t-2} \tag{5}$$

$$\Delta\Delta\vec{c_t} = \Delta\vec{c}_{t+1} - \Delta\vec{c}_{t-1} \tag{6}$$

These dynamic coefficients are then concatenated with the static coefficients $\vec{c}_k$ according to making up the final output of feature analysis representing the t^{th} speech frame.

$$\vec{x_t} = [\vec{c_t} \quad \Delta\vec{c}_t \quad \Delta\Delta\vec{c}_t] \tag{7}$$

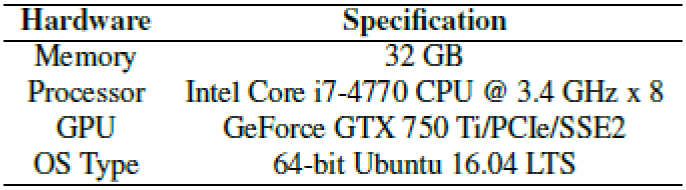
3.2.3. MEL Scale Cepstral Analysis (MEL)

Mel scale cepstral analysis is very similar to perceptual perceptual linear predictive coefficients (PLP), where the short term spectrum is modified based on psychophysically based spectral transformations. In this method, however, the spectrum is warped according to the MEL Scale, whereas in PLP the spectrum is warped according to the Bark Scale. The main difference between Mel scale cepstral analysis and perceptual linear prediction is related to the output cepstral coefficients. The PLP model uses an all-pole model to smooth the modified power spectrum. The output cepstral coefficients are then computed based on this model. In contrast Mel scale cepstral analysis uses cepstral smoothing to smooth the modified power spectrum. This is achieved by direct conversion of the log power spectrum to the cepstral domain using the standard algorithm of Inverse Discrete Fourier Transform (iDFT).

3.3. The VoiceNet Model

In this study, among the 1200 samples extracted using the batch generator, 1080 samples were used for training of the model, and 120 samples were used for testing of the model.We used a regression neural network that takes an input of size (20,170) consisting of 3 fully connected layers, 3 dropout layers and a softmax activation layer. The neural network uses adam as optimizer and categorical cross entropy as loss function. The network has been visualized in the the following graphic (Fig. 4). The training of the network was carried out on a system with specifications listed in Table I.

Table 1: System Specifications



Hardware	Specification
Memory	32 GB
Processor	Intel Core i7-4770 CPU @ 3.4 GHz x 8
GPU	GeForce GTX 750 Ti/PCIe/SSE2
OS Type	64-bit Ubuntu 16.04 LTS

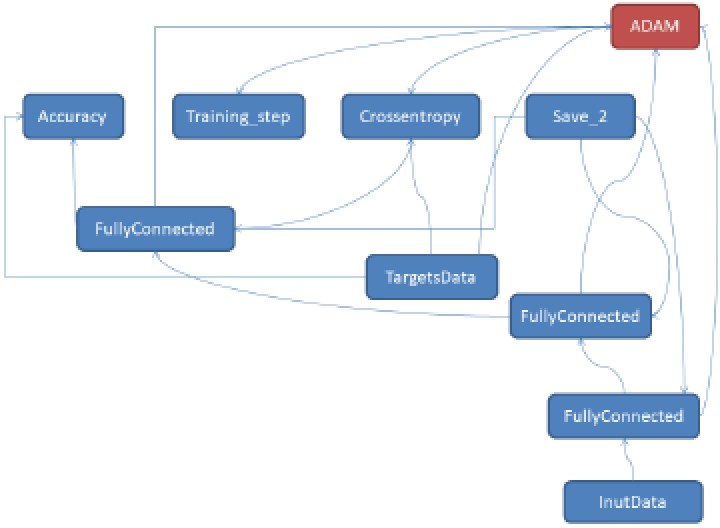


Figure 4.: Network Architecture

4. Hardware Implementation

Two quadrotors have been tested with this algorithm.

4.1. Bebop 2

An off-the-shelf quadrotor, Parrot Bebop 2 (Fig. 5), compatible with Python programming language, was used as hardware platform for algorithm development and testing. WiFi is used for communicating between the systems.

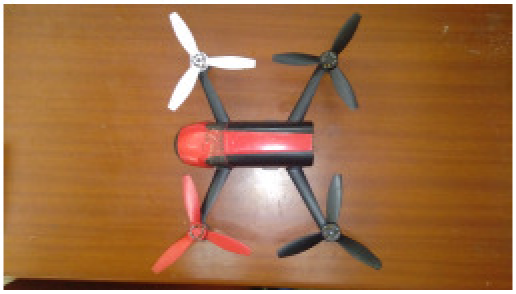
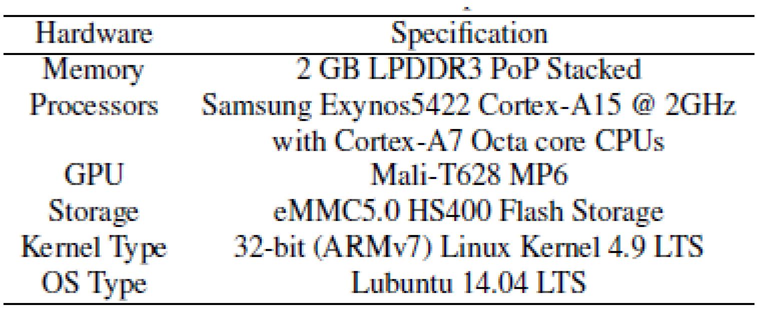


Figure 5: Parrot Bebop 2

4.2. Custom Drone

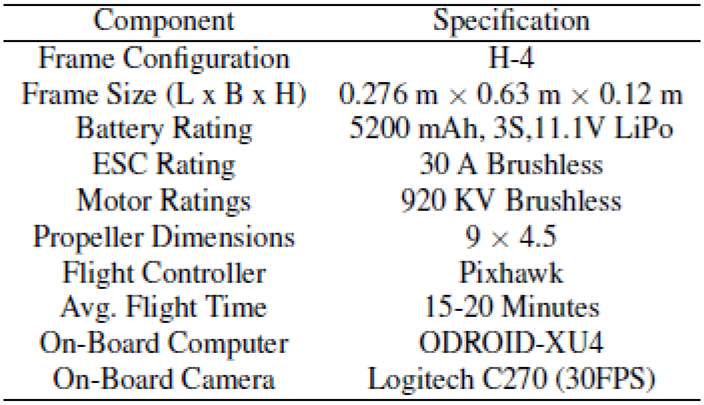
As a common understanding, there is a requirement for a custom-built quadcopter with onboard computational capabilities. The BumbleB (Figure 6), the drone we designed and fabricated, is equipped with a companion ODROID-XU4 single-board-computer (see Table II) which runs the VoiceNet algorithm. The specifications of BumbleB are tabulated in Table III.

Table 2 : ODROID-XU4 Specifications



Hardware	Specification
Memory	2 GB LPDDR3 PoP Stacked
Processors	Samsung Exynos5422 Cortex-A15 @ 2GHz with Cortex-A7 Octa core CPUs
GPU	Mali-T628 MP6
Storage	eMMC5.0 HS400 Flash Storage
Kernel Type	32-bit (ARMv7) Linux Kernel 4.9 LTS
OS Type	Lubuntu 14.04 LTS

Table 3 : Drone Specifications

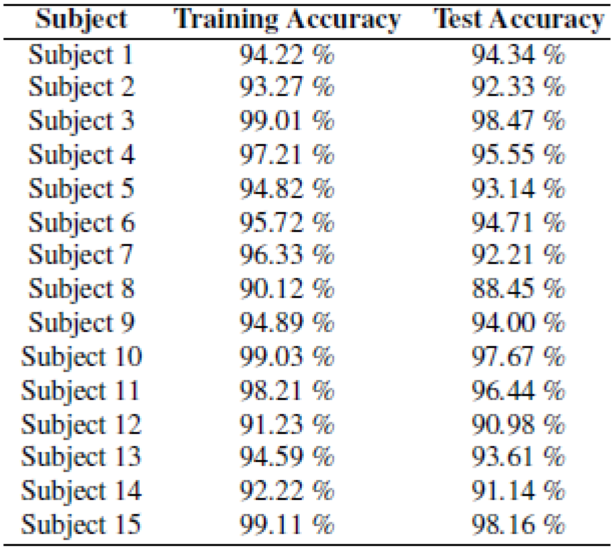


Component	Specification
Frame Configuration	H-4
Frame Size (L x B x H)	0.276 m × 0.63 m × 0.12 m
Battery Rating	5200 mAh, 3S,11.1V LiPo
ESC Rating	30 A Brushless
Motor Ratings	920 KV Brushless
Propeller Dimensions	9 × 4.5
Flight Controller	Pixhawk
Avg. Flight Time	15-20 Minutes
On-Board Computer	ODROID-XU4
On-Board Camera	Logitech C270 (30FPS)

5. EVALUATION AND RESULTS

Since there are not many metrics available pertaining to our current problem statement, we report the classification accuracy of our VoiceNet on custom dataset. We also take into consideration the various pitches of the subjects who were involved in the study. The VoiceNet model takes approximately 1.33 seconds to process an audio sample and predict the word said. This is primarily because of the small neural network designed and various features fed into it. The break-up of the timing is 0.34 seconds for feature extraction and 0.99 seconds for prediction.

Table 4 : Individual Accuracies of Subjects



Subject	Training Accuracy	Test Accuracy
Subject 1	94.22 %	94.34 %
Subject 2	93.27 %	92.33 %
Subject 3	99.01 %	98.47 %
Subject 4	97.21 %	95.55 %
Subject 5	94.82 %	93.14 %
Subject 6	95.72 %	94.71 %
Subject 7	96.33 %	92.21 %
Subject 8	90.12 %	88.45 %
Subject 9	94.89 %	94.00 %
Subject 10	99.03 %	97.67 %
Subject 11	98.21 %	96.44 %
Subject 12	91.23 %	90.98 %
Subject 13	94.59 %	93.61 %
Subject 14	92.22 %	91.14 %
Subject 15	99.11 %	98.16 %

6. DISCUSSION AND CONCLUSION

A novel solution to UAV control has been presented in this paper. The fact that a drone does not need an extensive speech recognition system to odentify only some keywords like take-off, forward etc. This calls for a smaller sized deep nets for speech recognition. Further aspects of this research include decreasing the time complexity even further and making the interface more robust so that it could be integrated with robots of different nature.

REFERENCES

[1] Beard, RandalW., Derek Kingston, Morgan Quigley, Deryl Snyder, Reed Christiansen,Walt Johnson, Timothy McLain, and Michael Goodrich. ”Autonomous vehicle technologies for small fixedwing UAVs.” Journal of Aerospace Computing, Information, and Communication 2, no. 1 (2005): 92-108.

[2] Draper, Mark, Gloria Calhoun, Heath Ruff, David Williamson, and Timothy Barry. ”Manual versus speech input for unmanned aerial vehicle control station operations.” In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 47, no. 1, pp. 109-113. Sage CA: Los Angeles, CA: SAGE Publications, 2003.

[3] Jordan, Thomas, John Foster, Roger Bailey, and Christine Belcastro. ”AirSTAR: A UAV platform for flight dynamics and control system testing.” In 25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference, p. 3307. 2006.

[4] McLain, TimothyW., and RandalW. Beard. ”Unmanned air vehicle testbed for cooperative control experiments.” In American Control Conference, 2004. Proceedings of the 2004, vol. 6, pp. 5327-5331. IEEE, 2004.

[5] Prodeus, A. M. ”Performance measures of noise reduction algorithms in voice control channels of UAVs.” In Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), 2015 IEEE International Conference, pp. 189-192. IEEE, 2015.

[6] Rabiner, Lawrence R. ”A tutorial on hidden Markov models and selected applications in speech recognition.” Proceedings of the IEEE 77, no. 2 (1989): 257-286.

[7] Hirsch, Hans-Gnter, and David Pearce. ”The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions.” In ASR2000-Automatic Speech 11 Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW). 2000.

[8] Hinton, Geoffrey, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior et al. ”Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups.” IEEE Signal Processing Magazine 29, no. 6 (2012): 82-97.

[9] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. ”Speech recognition with deep recurrent neural networks.” In Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, pp. 6645-6649. IEEE, 2013.

[10] Itakura, Fumitada. ”Minimum prediction residual principle applied to speech recognition.” IEEE Transactions on Acoustics, Speech, and Signal Processing 23, no. 1 (1975): 67-72.12

SENTIMENT CLASSIFIER AND ANALYSIS FOR EPIDEMIC PREDICTION

Nimai Chand Das Adhikari, Vamshi Kumar Kurva, Suhas S, Jitendra Kumar Kushwaha, Ashish Kumar Nayak, Sankalp Kumar Nayak and Vaisakh Shaj

Analytic Labs Research Team

ABSTRACT

Intelligent Models for predicting diseases whether building a model to help the doctor or even preventing its spread in an area globally, is increasing day by day. Here we present a noble approach to predict the disease prone area using the power of Text Analysis and Machine Learning. Epidemic Search model using the power of the social network data analysis and then using this data to provide a probability score of the spread and to analyse the areas whether going to suffer from any epidemic spread-out, is the main focus of this work. We have tried to analyse and showcase how the model with different kinds of pre-processing and algorithms predict the output. We have used the combination of words-n grams, word embeddings and TF-IDF with different data mining and deep learning algorithms like SVM, Naïve Bayes and RNN-LSTM. Naïve Bayes with TF-IDF performed better in comparison to others.

KEYWORDS

Natural Language Processing, Text Mining, Text Analysis, Support Vector Machines, LSTM, Naive Bayes, TextBlob, Tweet Sentiment Analysis

1. INTRODUCTION

The power of predictive modelling is gaining the importance in this ever-growing research areas. Researchers are taking the advantage of the power of analytics to analyse the data and create a relation out of it. Text Analysis with the twitter data is one of the most important area of focus for all the researchers. One of the text analysis problem is the sentiment analysis. It has been a useful tool for analysing array of problems that are related to human computer interaction. This can be extended to the fields of sociology, advertising, healthcare [15] and marketing, and thus ultimately to the area of social media. Sentiment could be described as subjective nature of an individual, which relates to the "private state" of an individual. Private state could be described as something which cannot be classified either as an objective observation or for verification. Sentiment analysis is an important field of research that has been closely associated with natural language processing, computational linguistics and text mining, which is used during sentiment analysis to identify the information and extract the same. The extracted information is quantified, then affective states are studied along with the subjective information. It could be said that sentiment analysis aims to ascertain the attitude of the individual, from whom the information is generated, with respect to the contextual polarity of the content which is being analysed [1]. Sentiment analysis is also referred to as "subjective analysis" or "opinion mining", along with

Natarajan Meghanathan et al. (Eds) : SAI, ICAITA, CSITA, ISPR, Signal - 2018
pp. 31–48, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81004

traces of connectivity to affective computing. Affective computing refers to the recognition of emotions by computers (Picard, 2000).

1.1 Sentiment Analysis

Sentiment analysis usually studies those elements which are subjective in nature. These elements are the words, phrases, or on some occasions it might be the sentences. This shows that sentiments exist in the form of small linguistic units. Through sentiment analysis, one can ascertain the actual intent of the author. Hence, this could be described as a phenomenon which is capable of detecting the sentiments from a given content automatically. Indeed, the sentiment analysis has been a boon for the organizations, because through opinion mining organizations are capable of making better decisions than before. In this case, an organization can develop its strategy based on the analysis that is derived from the opinion of the users. Hence, this facilitates better decision making by deciphering the emotions that is embedded in the word or a sentence (Pozzi, et al., 2016). It is a known fact, that sentiment analysis can be used effectively for extracting sentiments from contents displayed on social media. However, there are multiple research works which has demonstrated that this phenomenon can be used effectively to counter the epidemics. One of the research done in this regard, demonstrated that there is a strong relationship between the frequency of social media messages and the online news articles. The epidemic in question for this research was "measles". The research demonstrated, how monitoring of social media can be effectively used for the improvement of communication policies that can create general awareness amongst the masses. The data that has been extracted from the content of social media provide deeper insights into the "opinion" of the public which are at a certain moment, are salient amongst the public, that actually assists the health institutes to respond on an immediate basis to the concerns of public. In other words, through sentiment analysis opinion of the public related to epidemic disease can be sensed, and appropriate action can be taken based on that [4]. Opinion mining of social media content through sentiment analysis helps the public health officials to keep a track of spreading epidemics and take counter measure accordingly. They can also track the locations where the epidemic is spreading. Moreover, through sentiment analysis of the contents of the social media, it will be easier to detect the speed at which the epidemic is spreading. In another research it was found that social media platforms like twitter can be used as an important source of information, in a real-time situation. This helps to understand, how much concerned public is, on the outbreak of epidemic. This can be thoroughly achieved by sentiment classification of the twitter messages to develop an understanding on "degree of concern (DOC)", that is exhibited by the twitter users. The research adopts two-step process for classifying the sentiments, identifying the personal tweets and the negative tweets separately. With the help of this workflow, the researcher developed a tool for monitoring epidemic sentiments, that will visualize the concerns of the users of the twitter, regarding different types of epidemics. In this regard clue-based learning methods and machine learning method were used for classification of the twitter messages. With the help of Multinomial Naïve Bayes method, a classifier was built, and was sentiment analysis of tweets (Ji, et al., 2013). This phenomenon has been also classified as "knowledge-based tweet classification for the sentiment monitoring of diseases. For sentiment analysis of epidemics, the investigation of the sentiment dynamics of the media sources needs to be done primarily. Here, the media sources include tweeter and different online news publications which publishes content on the outbreak of epidemic diseases. A generic approach to perform the sentiment analysis will be as discussed in (Kim, et al., 2015).

1.2 Approaches

There are multiple approaches that has been devised to detect the outbreak of epidemics through twitter. One of the most common approach that is used to create a locational network for a specific country is completely based on the data taken from twitter. The data is taken from the social media of the created location networks and are integrated with an algorithm to detect any form of outbreak of epidemic diseases. This approach can also be used to forecast the breakout of any form of epidemic diseases (Thapen, et al., 2016). Another approach will be to make use of Twitter API to extract the tweets with the epidemic name. Then the tweets are filtered based on a given criteria such as tweeted by patients or GP, with the help of support vector machine identifier (SVM) classifier (Aramaki, 2011). In fact, there are multiple NLP techniques that can be used to extract the tweet data based on the keywords and detect the outbreak of any form of epidemics. There are existing researches which demonstrate that the conventional sentiment analysis methodologies can be successfully used for sentiment analysis in the social networks. This has been in practice since the early 2000. There are multiple evolutions of various types of sources where opinions can be voices. Hence, the current opinion methodologies might no longer be effective, in this redeveloped environment. In this environment, multitude of issues need to be derived from the conventional sentiment analysis along with natural language processing. Overall, this creates a challenging environment with different set of complexities that includes, noisy content, short messages, variant form of metadata (age, sex, location). It is a known fact, that social networks create a clear impact on the languages, and this has become a core challenge of sentiment analysis. There is a constant evolution of language on the social network, which is used to generate the online content. Also, most of the written languages is visualized though some electronic screens such as desktop, laptop, tablets or phones, hence it could be said that the interaction partly happens with the help of technology. Moreover, the language that is used on the social media is more of malleable nature in comparison to the language that has been used for formal writing. The social media language is made up of personal communication and informal opinions, which is afforded by the mass users of the social media platform. This actually makes it more difficult for the conventional sentiment analysis method to analysis the inherent opinion from the given text. Hence, in order to adapt the changing language structures, research needs to implement strong natural language processing skills and linguistic skills, along with the conventional methodologies of sentiment analysis.

In this work we have taken the tweet data to arrive at the prediction of epidemic hit areas or the probability of being affected by any major epidemic that can harm the lives and property. We have used machine learning approach to arrive at the prediction and for comparison and analysis we have used different feature extraction techniques and algorithms to select the best out of it. In the next section we will be discussing about the dataset that we have tried to generate out of the tweet data from the twitter and transforming it into the structured from un-structured data and making it a supervised learning problem. After that we will discuss about the structure of the system for predicting the epidemic and different methodologies taken up for arriving at the better result which in case is the Accuracy and different parameters of the Confusion Matrix. Following that will be the results section and Future aspects of the work and then conclusion.

2. DATA DESCRIPTION AND DATA PROCESSING

The work here, Epidemic Search model using the power of the social network data analysis and then using this data to provide a probability score of the spread and to analyse the areas globally going to suffer from any epidemic spread-out. The easily available social network data from Twitter which in other words known as the tweet-data is very helpful in providing a lot of information about any events happening globally.

2.1 Data Source

In the recent years, social networking has attracted a lot of users. Social networking sites like Facebook, Twitter, Instagram etc. creates a lot of data every second and a lot of information from that can be got. Hence, this creates a space for doing some challenging research by computationally analysing the sentiments and opinions of the textual data which are unstructured in their behaviour. To achieve this, a gradual practice has grown for extracting the information from the data available in the social networking sites like predicting the epidemic in this case. The accuracy of the predicting model thus can be found out from the modelling output. To arrive at the output of the scenario presented here, tweet data is analysed and to extract the tweet data "Twitter API" is used. API needs to be signed up on the twitter and also has to have a login into the developer Twitter account. Following it, an application or an API needs to be developed which can be then used to provide the keys and the tokens for using it in the programming environment.

2.2 Data Extraction

The Twitter API can then be used with the Python Programming language to extract the tweets from the Twitter and store in a HDFS (Hadoop Distributed File System) which is a distributed file system that is designed to run on any commodity hardware. *Tweepy* is a python library that can be used to extract the tweeter tweets. The tweets can be easily collected and can be stored in the JSON format. JSON is a syntax for storing as well as exchanging the stored data.

2.3 Database Management

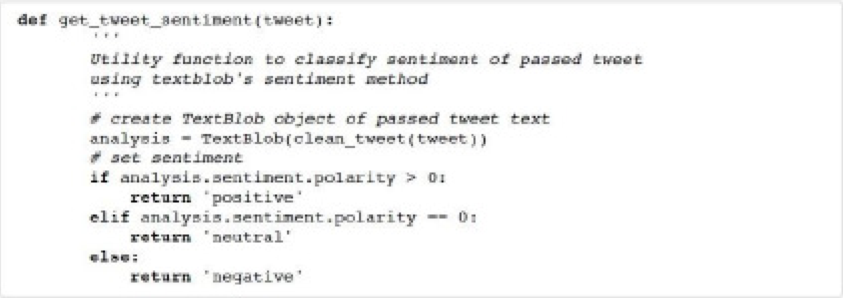
As in present scenario we have large storage of tweets, storing it on a single system and analysis can be difficult due to large data. This problem can be solved using distributed system. Example for storage we can use HDFS file systems or Apache Cassandra database management system. Spark is a cluster- computing framework which can be used with them and for python we can use python supported spark system which is pyspark. This has the advantage of storing very large dataset and to be accessed reliably depending on the bandwidth of the user. Another advantage is in the distributed system many clusters can host and execute directly attach storage and user application tasks. In this system either MongoDB or Spark system can be directly used with Python to extract the tweets and store in the distributed clusters. MongDB is a free and open source cross platform document-oriented database program. In this json like documents are used which has schema. It works on concept of collection and document. Where a document is a set of key-value pairs. And Collection is a group of these MongoDB documents. Here, Collection is equivalent to a RDBMS table. Also, it is contained in a database which is a physical container for collections and each database gets its own set of files on the file system.

2.4 Pre-processing

Removing the stop words like the, an, a etc can be a good step as they don’t determine the polarity or sentiment of the tweet. For this we mostly use stop words in English from nltk package in python. Removing hyper-links, citations, references, hash-tags, multiple white-spaces can be done by regular expressions which makes the tweet description free from the "unrelated" English words and "chat language".

2.4.1 Polarity Generation

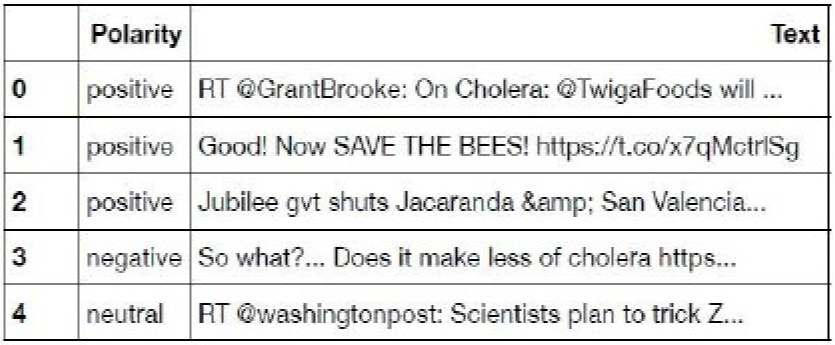
The predicting variable or the dependent variable which in this case is the polarity of the tweet that is found out from the sentiment of the tweets using the textblob in python. It targets some commonly areas like POS tagging (Parts of Speech tagging), Noun-Phrase terms extraction from text, Sentiment Analysis, Text Classification, Language Translation in text etc. Here a simple function for doing such task is used as below which targets for the tweet sentiment analysis: The function for getting the tweet sentiment is as below: This is used to generate the polarity class for the tweet. Thus, making the unstructured data into a structured data.



```
def get_tweet_sentiment(tweet):
        '''
        Utility function to classify sentiment of passed tweet
        using textblob's sentiment method
        '''
        # create TextBlob object of passed tweet text
        analysis = TextBlob(clean_tweet(tweet))
        # set sentiment
        if analysis.sentiment.polarity > 0:
            return 'positive'
        elif analysis.sentiment.polarity == 0:
            return 'neutral'
        else:
            return 'negative'
```

Figure 1: Polarity Generation Function

Below is the head of the data:



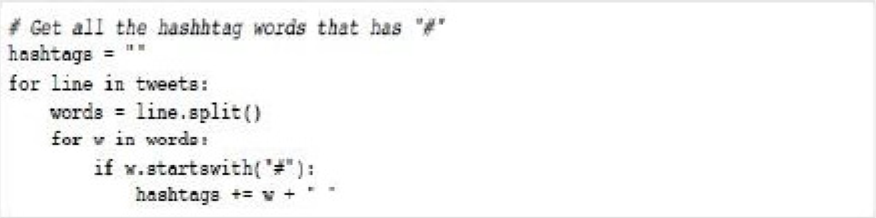
	Polarity	Text
0	positive	RT @GrantBrooke: On Cholera: @TwigaFoods will ...
1	positive	Good! Now SAVE THE BEES! https://t.co/x7qMctrlSg
2	positive	Jubilee gvt shuts Jacaranda &amp; San Valencia...
3	negative	So what?... Does it make less of cholera https...
4	neutral	RT @washingtonpost: Scientists plan to trick Z...

Figure 2: Head of the Data

2.4.2 Hash Tag Analysis

All the words starting with the symbol # are hash-tags. These are helpful in understanding the trending issues. A word-cloud is an image representing the text in which the size of each word is proportional to the frequency of occurrence. Hence the bigger words are the most tweeted topics.

A glance at the word cloud shows that most tweets are about the social problems like diseases, malnutrition, starvation and some countries affected by them. In our case, most frequent hash tagged words are Yemen, Cholera, CholeraNairobi, zika, vaccines, TheStoryOfYemen etc. Below is the function that is used for the hash tag analysis.



```
# Get all the hashhtag words that has "#"
hashtags = ""
for line in tweets:
    words = line.split()
    for w in words:
        if w.startswith("#"):
            hashtags += w + " "
```

Figure 3: Hash Tag Function

When the generated hash tags are generated and represented as the word cloud, it looks as below:

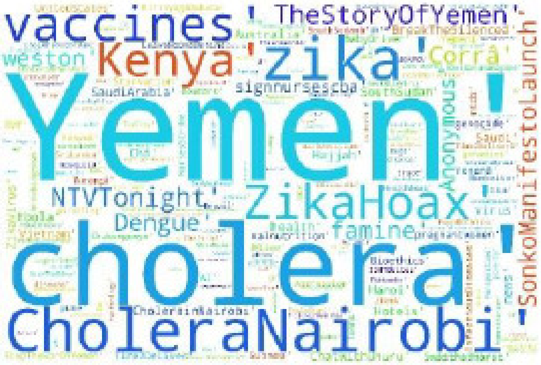


Figure 4: Hash-Tag word Cloud

2.4.3 Top-Users/Citations

Similar to analysing hashtags, we can extract the usernames usually preceded by @ symbol. The word cloud for the Citations is as below figure.

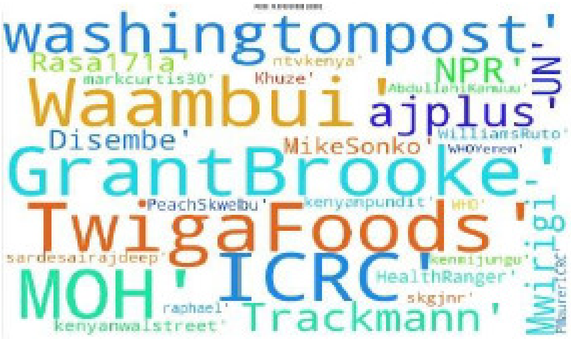


Figure 5: Citations Word Cloud

From the above word-cloud, we find that the top cited words are "washingtonpost", "Waabui", "GrantBrooke","TwigaFoods,"ICRC" etc.

3. SYSTEM DESIGN

In the system design section, here it will be present how the steps are followed to arrive at the prediction results.

1. Step: Tweet Data Streaming- Using Tweeter API

2. Step: HDFS MongoDB- Tweets extracted stored in MongoDB using Python library pymongo

3. Step: Dataset for Training-Available data as text data from tweeter is highly unstructured and noisy in nature and to use it for the modelling purpose, it needs to be cleaned. The different pre-processing techniques used as follows:

    (a) Escaping HTML characters

    (b) Decoding data

    (c) Removal of Stop-words

    (d) Removal of Punctuation

    (e) Removal of Expressions

    (f) Split Attached Words

    (g) Removal of URLs

    (h) Removal of quotes

    (i) Removing tickers

    (j) Removing line-break, tab and return

    (k) Remove whitespaces

4. Step: Label or Polarity Generation-The generated tweets without the sentiments class, imputed with the class using the TextBlob library in Python. Three classes are generated: Positive, Neutral and Negative.

5. Step: Feature Extraction- Different techniques used to analyse the accuracy of the prediction:

    (a) Bag of Words using CountVectorization, Uni-Grams and Bi-Grams

    (b) TF-IDF - Creating a unique value for the terms in a particular document.

    (c) Topic Modelling using LDA - To generate the topics for the corpus

6. Step: Machine Learning Model-The Features extracted or generated is fed into the Machine Learning Model/Algorithms to generate the results.

7. Step: Results Generation

Below is the flow chart Graph for the above Algorithm:

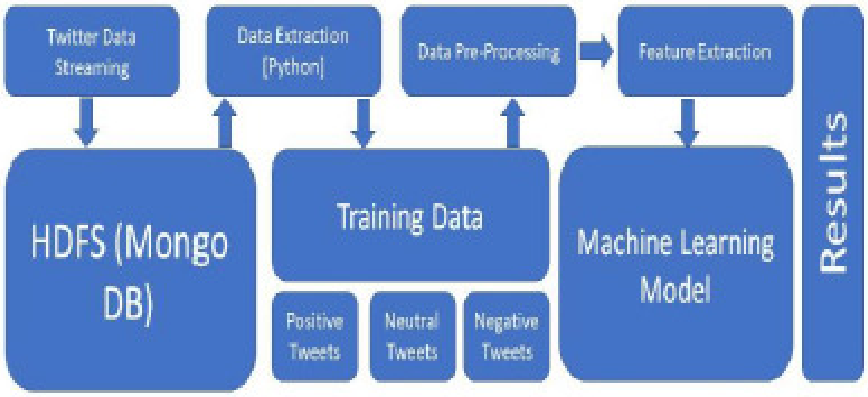


Figure 6: System Design Flow Chat of the Epidemic Prediction from Tweets

4. Experiments and Results

As the unstructured data is converted into a supervised learning process, it is important to see the distribution and the counts of the different classes in the dataset. Let us now see how the distribution of the different classes belonging to the tweets is:

- Positive Tweets: Considering only the positive labelled tweets and extracting words, we can count the frequent words used in positive tweets. Word-cloud of positive tweets shows that they include health, water, vaccine, sanitation among other things. The textblob shows that there are 771 cases as termed as the positive class which is around 26.67% of the total tweets cases. The word-cloud for this classes is as below in the figure.

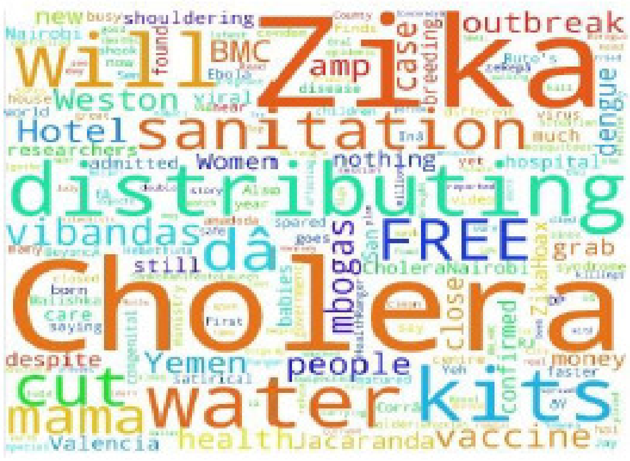


Figure 7: Positive Sentiment Word Cloud

- Negative tweets: Similarly, negative word-cloud shows that outbreak, dengue, worst, Yemen etc. are most used in negative tweets. For this case, the total number for the negative class tends to 692 which comprises of 23.96% of the total cases. The word-cloud for the negative classes is as below in the figure.

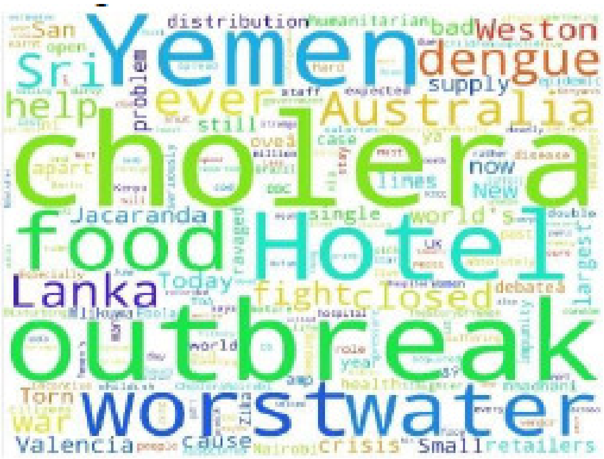


Figure 8: Negative Sentiments Word Cloud

- Neutral tweets: Word-cloud shows that the most used words in neutral tweets are hotel, cholera, Weston spread etc. The total for the neutral case tends to 1425 which is 49.34% of the total cases. The word cloud for this class is as below in the figure.

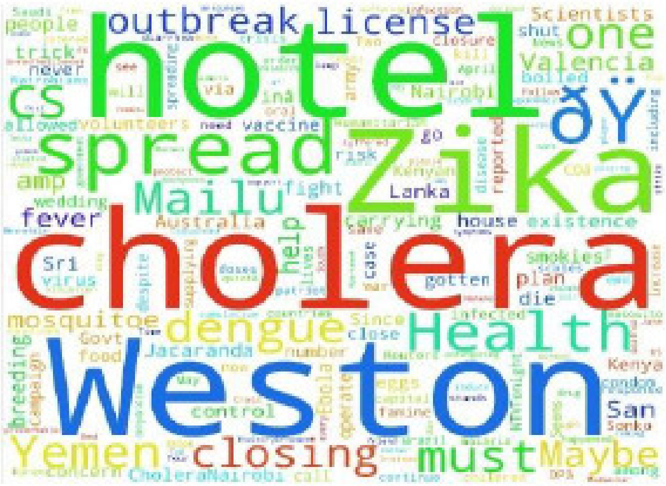


Figure 9: Neutral Sentiment Word Cloud

The histogram of the polarity of all the tweets from the blob. sentiment. polarity shows the distribution of the polarity scores of each tweet class and is represented through the histogram as below:

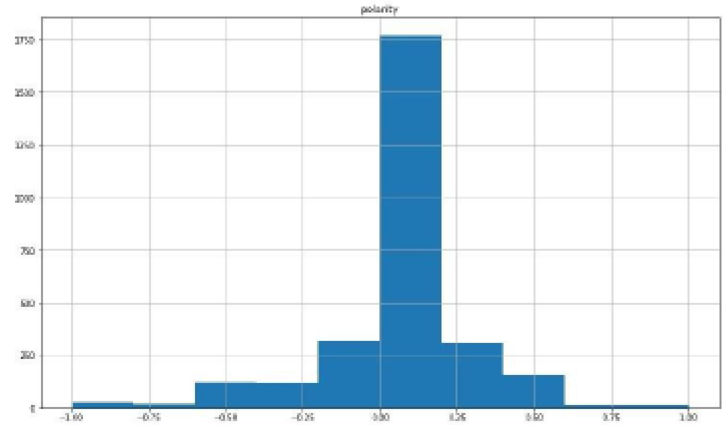


Figure 10: Histogram of the sentiment Polarity

The histogram shows that maximum polarity of the tweets lies in the range [0.0–0.24) approximately. If we closely observe the distribution that the tweet polarity follows tends to be "Normal Distribution". To analyse more on the length of the tweet and the polarity class of the tweet, when plotted, the histogram looks as shown below: This analysis shows that for the negative class, the frequency of the word counts is mostly more near to the 140-word count. The same is seen for the positive class but the frequency distribution is less than that of the negative class. For the neutral class, the frequency distribution is more in between the word counts around 130-150.

4.1 Machine Learning

Once we clean the data and get a rough idea about the data, we can use any supervised ML model for sentiment classification since we already have the labels. Most of the approaches involving text classification uses n-gram features. This comes under Bag of words model as it doesn't care in the exact ordering of the words. Recent advanced models using RNN/LSTM models take the word order into consideration while classifying.

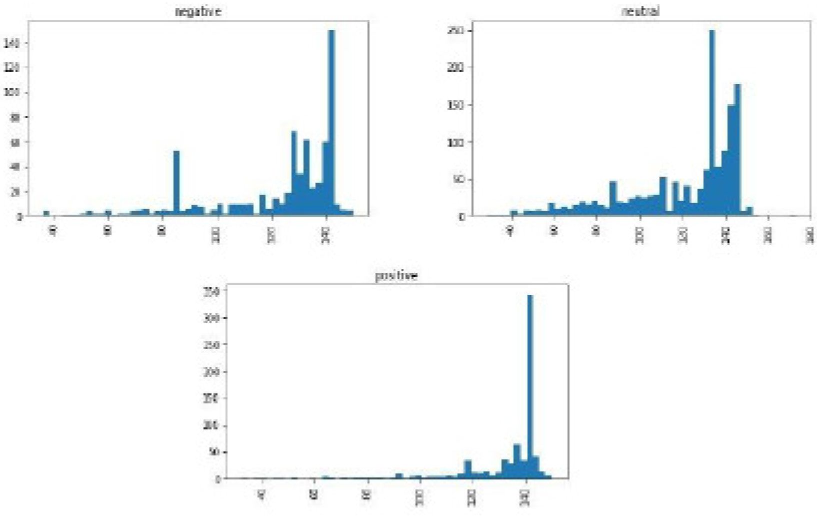


Figure 11: Histogram of the Sentiment Scores of different class

4.1.1 Sentiment Classifier:

Now for classifying the tweets and see how the prediction happens using different classifier, we use the above tweet dataset and pass it through any machine learning algorithm and see how the result is. Below is the comparison of the accuracies of the different models/algorithms that we have used: The above analysis shows that Decision tree classifier performed better followed by the KNearestNeighbour Classifier.

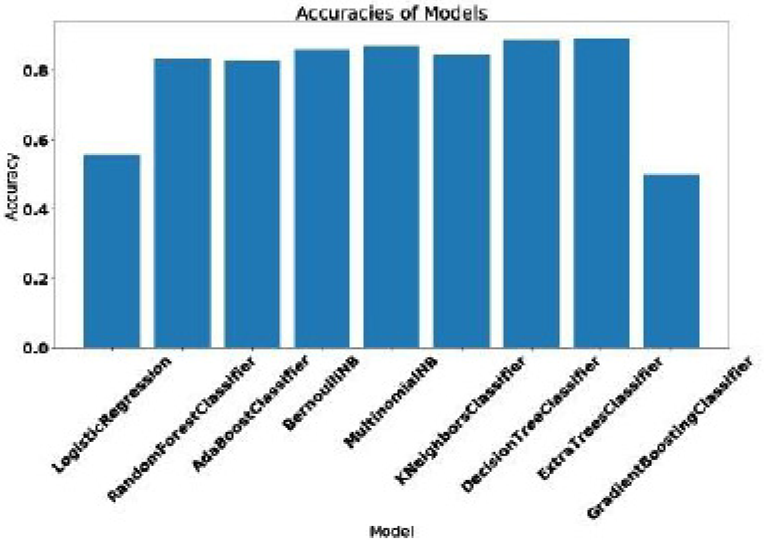


Figure 12: Comparison of the different algorithm performance

4.1.2 Words n-Grams:

A tweet (such as a sentence or a document) is represented as the bag (multi set) of its words, disregarding grammar and even word order but keeping multiplicity.

- Strengths: Traditional, pretty solid feature representation.
- Weaknesses: Lose grammar/word order.
- Hyperparameters: The algorithm definition to execute

Below is the Feature Evaluation of different algorithms using the n-grams concept:

Naïve Bayes using Uni-grams and Bi-grams

Here we have used two Naïve Bayes algorithm. One is Bernoulli Naïve Bayes and other is Multinomial Naïve Bayes. Bernoulli Naïve Bayes follows Bernoulli distribution whereas Multinomial Naïve Bayes model is mainly based on the frequency of data. We have used the 5-fold cross-validation technique in which 4 out of 5 folds (in other words samples) is used for the training of the model and 1 out of 5 folds or the last fold is used for the validation of the model performance. From the 5-fold cross validation technique, average accuracy for the multinomial NB is found out to be 78.11% and that for the Bernoulli NB it’s found to be 78.42%, minimum value for accuracy for multinomial NB is 61.53% and for Bernoulli NB is 65.16% and maximum value of accuracy for multinomial NB is 83.36% and for Bernoulli NB it's found to be 82.7%. We can find that on the basis of the average and minimum accuracy values, Bernoulli Naïve Bayes performed better than Multinomial Naïve Bayes. Whereas we find that the maximum accuracy value if for Multinomial Naïve Bayes.

For the hold-out Validation method, accuracy of multinomial NB is 84.43% and the confusion matrix for the same is as below:

Table 1: Confusion Matrix of Multinomial NB

Label	Predicted Negative	Neutral	Positive
Actual Negative	175	20	6
Actual Neutral	32	373	23
Actual Positive	22	32	184

From this we can analyse that out of 213 negative sample 175 is predicted correctly while 20 are wrong predicted to neutral class and 6 are predicted to positive class. Out of 423 neutral tweets only 32 are predicted to negative class and 23 are predicted to positive class which shows that our neutral labelled tweets are biased towards negative tweet. For positive sentiment tweets it gave total 54 wrong classification where 22 are negative classified and 32 are classified to neutral class and 184 are correctly classified. This also shows that chances of negative tweet to be predicted as positive are comparatively very less than positive tweet to be predicted as negative. Although it gave accuracy of 0.84 but misclassification is more when we see confusion metric. Same analysis we can see from precision, recall and f1-score values as shown below:

Table 2: Analysis

Class-name	Precision	Recall	F1-Score	Support
Negative	0.76	0.87	0.81	201
Neutral	0.88	0.87	0.87	428
Positive	0.86	0.77	0.82	238
Avg. / Total	0.85	0.84	0.84	867

Linear SVM using Uni-gram and Bi-grams

Accuracy for this model is 83.50% and when running using cross-validation method with 5-folds our accuracy ranges from 71.06% to 84.4%. The mean accuracy is around 80.09%. Hence, for the better representation and comparison we will be considering the mean accuracy for our final evaluation. Confusion metric for the model is:

Table 3: Confusion Matrix of SVM

Label	Predicted Negative	Neutral	Positive
Actual Negative	145	68	0
Actual Neutral	1	421	1
Actual Positive	0	73	158

From this we can analyse that out of 213 negative samples, 145 is predicted correctly while all wrong prediction is in neutral. This shows that negative and positive sentiments tweets can be separated much more easily as compared to negative and neutral sentiment tweets. Out of 423 neutral tweets only 2 tweets are predicted wrong and out of the remaining, 421 tweets are predicted correctly. This implies that the model performed better for the cases of neutral tweets. Also, for the positive sentiment tweets, it predicted 73 wrong classification out of 231. Which shows same pattern as that of the negative tweets.

If we see precision, recall and f1-score we can see that F1 score of neutral tweets are higher than negative and positive tweets which supports our analysis presented above.

Table 4: Analysis

Class-name	Precision	Recall	F1-Score	Support
Negative	0.99	0.68	0.81	213
Neutral	0.75	1.00	0.85	423
Positive	0.99	0.68	0.81	231
Avg. / Total	0.87	0.84	0.83	867

Thus, comparing the above analysis:

Table 5: Comparison Results of Different Algorithms

Algorithm	Value 1	Value 2	Value 3
Multinomial NB	78.11%	61.53%	83.36%
Bernoulli NB	78.42%	65.16%	82.70%
SVC	80.09%	71.06%	84.4%

SVC using "Linear" kernel performed extremely well for this analysis.

4.1.3 TF-IDF Vectorizer

We have used the scikit-learn’s TFidfTransformer to arrive at the features to be input to the different classifier. We have used the following classifiers along with their performances:

- Naive Bayes: 94.145%
- SVC: 49.34%
- SVM(TFIDF): 87.9%
- Naive Bayes(TFIDF): 83.21%

Now including the n-gram analysis to our model, the following things we have included and built in the model:

- Unigram classifier (with mark-negation and without)
- Bigram classifier (with mark-negation and without)

- Unigram and bigram classifier (with mark-negation and without)

The following the result analysis:

- Unigram Classifier: 88.75%, 89.44%
- Bigram Classifier: 88.58%, 88.58%
- Unigram and Bigram Classifier: 89.10%, 88.75%

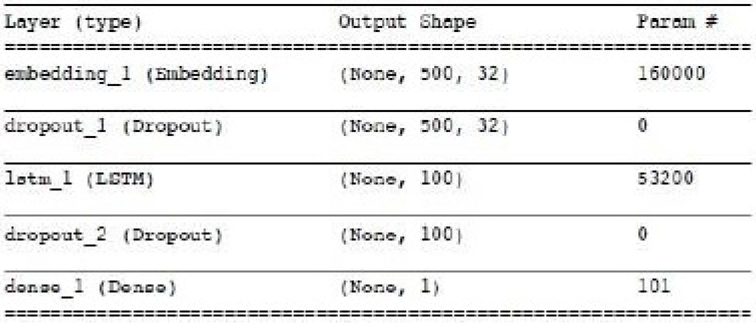
4.1.4 LSTM Networks

LSTM is a variant of recurrent neural network, which takes information of its previous time steps. In LSTM to handle drawback of Basic RNN cell for learning long sequences we corporates gates. For this model basic cleaning has been done and after that we have tokenized input tweets. After tokenization sentence is mapped with word index of its vocabulary and index of padding is kept as 0, padding ensure every input record have same length.

For passing sentences to model few points has been considered

- Preventing learning of least frequent words: vocabulary contains 5000 most frequent words.
- Each sentence is fixed with length of 500: smaller sentences are padded and longer sentences are truncated to length 500 words.

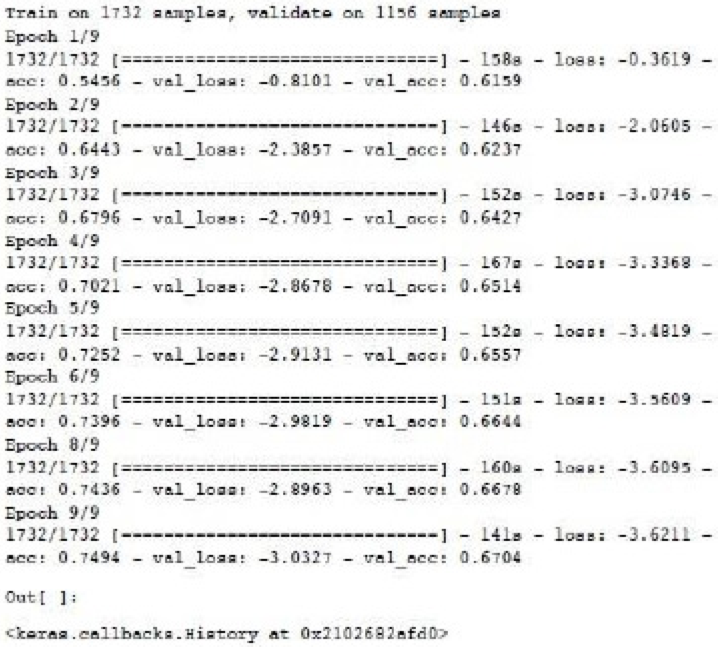
Here we are using self embedding technique for learning word embedding and embedded word size is kept at 32. The model consists of 5 layers:



```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 500, 32)           160000
_________________________________________________________________
dropout_1 (Dropout)          (None, 500, 32)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               53200
_________________________________________________________________
dropout_2 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 101
=================================================================
```

Figure 13: Architecture of LSTM

Below is the training of the LSTM model: We can see that the validation accuracy attained by LSTM is 67.04%.



```
Train on 1732 samples, validate on 1156 samples
Epoch 1/9
1732/1732 [==============================] - 158s - loss: -0.3619 -
acc: 0.5456 - val_loss: -0.8101 - val_acc: 0.6159
Epoch 2/9
1732/1732 [==============================] - 146s - loss: -2.0605 -
acc: 0.6443 - val_loss: -2.3857 - val_acc: 0.6237
Epoch 3/9
1732/1732 [==============================] - 152s - loss: -3.0746 -
acc: 0.6796 - val_loss: -2.7091 - val_acc: 0.6427
Epoch 4/9
1732/1732 [==============================] - 167s - loss: -3.3368 -
acc: 0.7021 - val_loss: -2.8678 - val_acc: 0.6514
Epoch 5/9
1732/1732 [==============================] - 152s - loss: -3.4819 -
acc: 0.7252 - val_loss: -2.9131 - val_acc: 0.6557
Epoch 6/9
1732/1732 [==============================] - 151s - loss: -3.5609 -
acc: 0.7396 - val_loss: -2.9819 - val_acc: 0.6644
Epoch 8/9
1732/1732 [==============================] - 160s - loss: -3.6095 -
acc: 0.7436 - val_loss: -2.8963 - val_acc: 0.6678
Epoch 9/9
1732/1732 [==============================] - 141s - loss: -3.6211 -
acc: 0.7494 - val_loss: -3.0327 - val_acc: 0.6704

Out[ ]:

<keras.callbacks.History at 0x2102682afd0>
```

Figure 14: Training results of LSTM

5. REPRESENTATION

For Visualizing there are a lot of tools available to showcase how the epidemic distribution is globally. Tableau is a visualization tool which can be connected to any database and different kind of visualization can be created to understand the data and better representation of the data.

In this project, Tableau is used to generate a visualization of the epidemic hit areas globally. A sample is presented below:

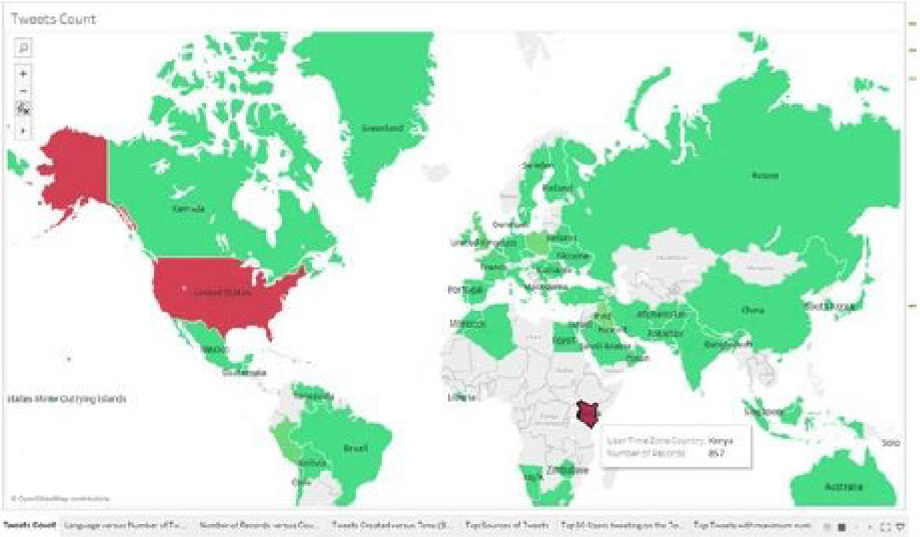


Figure 15: Epidemic Hit Regions using Tweet Analysis

6. CONCLUSION

The epidemic hit area prediction can be used to save a lot of lives globally. Here a lot of pre-processing of the unstructured data is done to make it to a structured data and different features extraction techniques like Count Vectorization, TF-IDF, Topic Modelling etc. is used to feed the data into the machine learning algorithms. The most important is the sentiment of the tweets to see how the polarity of the tweet is. The metric which is used here is the "accuracy" of the prediction and we have used the confusion matrix to arrive at the best performing algorithm. Naive Bayes using TF-IDF performed better than other methodologies and gave a better result.

REFERENCES

[1] Thomas, David R. "A general inductive approach for analyzing qualitative evaluation data." American journal of evaluation 27.2 (2006): 237-246.

[2] Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." Foundations and Trends R in Information Retrieval 2.1–2 (2008): 1-135.

[3] Adhikari, Nimai Chand Das. "PREVENTION OF HEART PROBLEM USING ARTIFICIAL INTELLIGENCE."

[4] Waaijenborg, Sandra, et al. "Waning of maternal antibodies against measles, mumps, rubella, and varicella in communities with contrasting vaccination coverage." The Journal of infectious diseases 208.1 (2013): 10-16.

[5] Miner, Gary, John Elder IV, and Thomas Hill. Practical text mining and statistical analysis for non-structured text data applications. Academic Press, 2012.

[6] Barbosa, Luciano, and Junlan Feng. "Robust sentiment detection on twitter from biased and noisy data." Proceedings of the 23rd international conference on computational linguistics: posters. Association for Computational Linguistics, 2010.

[7] Han, Eui-Hong Sam, George Karypis, and Vipin Kumar. "Text categorization using weight adjusted k-nearest neighbor classification." Pacific-asia conference on knowledge discovery and data mining. Springer, Berlin, Heidelberg, 2001.

[8] Pereira, Fernando C., Yoram Singer, and Naftali Tishby. "Beyond word n-grams." Natural Language Processing Using Very Large Corpora. Springer, Dordrecht, 1999. 121-136.

[9] Niesler, Thomas R., and Philip C. Woodland. "A variable-length category-based n-gram language model." Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on. Vol. 1. IEEE, 1996.

[10] Adhikari, Nimai Chand Das, Arpana Alka, and Raju K. George. "TFFN: Two Hidden Layer Feed Forward Network using the randomness of Extreme Learning Machine."

[11] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002.

[12] Dave, Kushal, Steve Lawrence, and David M. Pennock. "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews." Proceedings of the 12th international conference on World Wide Web. ACM, 2003.

[13] Joachims, Thorsten. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. No. CMU-CS-96-118. Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[14] Tang, Duyu, Bing Qin, and Ting Liu. "Document modeling with gated recurrent neural network for sentiment classification." Proceedings of the 2015 conference on empirical methods in natural language processing. 2015.

[15] Adhikari, Nimai Chand Das, Arpana Alka, and Rajat Garg. "HPPS: HEART PROBLEM PREDICTION SYSTEM USING MACHINE LEARNING." Computer Science & Information Technology: 23.

AUTHORS

Nimai Chand Das Adhikari received his Master’s in Machine Learning and Computing from Indian Institute of Space Science and Technology, Thiruvananthapuram in the year 2016 and did his Bachelor’s in Electrical Engineering from College of Engineering and Technology in the year 2011. He is currently working as a Data Scientist for AIG. He is a vivid researcher and his research interest areas include computer vision, health care and deep learning. He has started the Analytic Labs research group.

Vamshi Kumar Kurva received his Master’s in Machine Learning and Computing from Indian Institute of Space Science and Technology, Thiruvananthapuram in the year 2017. He is currently working as a Data Science Engineer for FireEye. His interest areas include deep learning, video analytics, medical application and NLP.

Sankalp Kumar Nayak has 9+ years of experience in SAP Data Analytics. He has handled and worked on multiple projects related to SAP data analytics in all of its phases (Implementation, support/maintenance, up-gradation and roll-out). Also has 2+ years of experience in dealing in RPA(Automation) projects which is also referred as Business Process Automation.

Jitendra Kumar Kushwaha is pursuing his Master’s in Machine Learning and Computing from Indian Institute of Space Science and Technology, Thiruvananthapuram and is graduating in the year 2019. He received his Bachelor’s in Information Technology from Bundelkhand Institute of Engineering and Technology, Jhansi in the year (2013-2017). His interest area includes NLP, Machine Learning, and Computer Vision.

Ashish Kumar Nayak received his Post Graduate diploma in Applied Statistics from Indira Gandhi Open University in the year 2017 and did his Bachelor’s in computer science engineering from Konark Institute of Science and Technology in the year 2010. He is currently working as a Data scientist for Accenture in finance domain. His interest area includes Machine Learning, Computer Vision, NLP and statistical analysis.

15

Suhas S received his Master’s in Machine Learning and Computing from Indian Institute of Space Science and Technology, Thiruvananthapuram in the year 2016 and did his Bachelor’s in Electronics and Communication Engineering from NMIT, Bangalore in the year 2014. He is currently working as a Research Associate in Indian Institute of Science(IISc). He is a 'Big data' enthusiastic individual with vested interest in 'Data Science' and his research interest areas include predictive modelling, computer vision and applied deep learning.

Vaisakh Shaj received his Master’s in Machine Learning and Computing from Indian Institute of Space Science and Technology, Thiruvananthapuram in the year 2016. He is currently working as a Data Scientist for Intel. He is a vivid researcher and his research interest areas include computer vision, health care and deep learning.

SKYLINE QUERY PROCESSING IN GRAPH DATABASES

Dina Amr and Neamat El-Tazi

Faculty of Computers and Information, Cairo University, Giza, Egypt

ABSTRACT

Skyline queries are mostly used in decision making processes and search space reduction. They received much attention during the past years due to their importance in discarding the unneeded data and providing the users with data that best match their interest. The same attention has been directed to graph databases to handle highly connected data due to the increase in volume and connectedness of today's data. The proposed work aims to augment graph databases with skyline queries. Two skyline query processing algorithms have been proposed; nested loops and divide-and-conquer. They are used to facilitate retrieving skyline results with multiple dimensions over graph databases. Performance evaluation for both algorithms over different sized graph databases and queries with different complexity levels were presented. The conducted experiments proved that divide-and-conquer outperforms nested loops in different cases.

KEYWORDS

Skyline Queries, Graph Database, Graph Querying, Neo4j, Cypher

1. INTRODUCTION

Nowadays most of the real-world applications like social networks can use graph databases to store their data, due to their countless advantages over relational models. One of the main advantages of graph database is that it has flexible schema and new information can be added on the fly. Unlike relational database which requires joins to retrieve a relationship between tables, data in graph database is connected through bidirectional relationships between nodes which make it easy to retrieve any linkage. Another advantage in graph models over relational, is that nodes are created using information supplied by the user which means that the database does not store null values. This helps in saving storage space.

An interesting type of existing queries in relational databases are skyline queries. Skyline queries retrieve a set of interesting points from a large dataset [1]. Skyline results contain the data that is not dominated by any other data [1]. Domination occurs based on some conditions. The conditions are determined according to user's preferences.

Most of decision making processes and data pruning techniques need skyline queries to return the best results based on user's requirements. Using already existing algorithms, we propose how to implement skyline queries over graph databases. We consider skyline queries that have multiple dimensions. The dimensions reflect node properties. A node dominates another node, if it is better in all dimensions. This is based on the selected criteria in the query input. The criteria can be maximum or minimum for the selected dimensions. One of the properties of skyline queries, is

Natarajan Meghanathan et al. (Eds) : SAI, ICAITA, CSITA, ISPR, Signal - 2018
pp. 49–57, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81005

that there are no weights for the dimensions used by the user. Thus, the domination is done based on properties' values only.

Table 1. Hotels Database.

HotelName	PricePerNight	DistancetoBeach	Stars
A	250	20	4
B	300	50	4
C	500	10	5
D	100	80	3
E	380	30	5

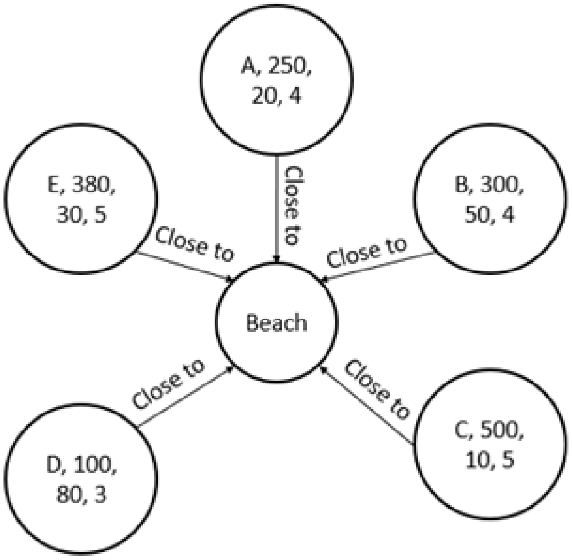


Figure 1. Hotels Database Nodes

A very famous example to illustrate skyline queries is a sample hotel database similar to that in Table 1 and Figure 1 which stores hotel names, price of room per night, distance to the beach and star rating, from this database the user needs to retrieve the hotel which is close to the beach and has low price per night. It is obvious that the two preferences may be conflicting, because hotels close to the beach will most probably be more expensive. This query has two dimensions: the first dimension is the PricePerNight and the second is DistancetoBeach. Thus, when a skyline query is applied to the dataset, it will return hotels which are better than other hotels in both preferences/dimensions. These hotels are called skyline. In this example, the hotels: “A”, “C” and “D” are the skyline results as none of them is better than the others across all dimensions. At this point it is the user's decision to select from the three hotels instead of returning the whole list of hotels to select from. The importance of skyline queries appears more with larger datasets with thousands of hotels. In this paper, we propose a way to implement skyline queries over a graph model.

In addition to the properties of graph that makes us motivated to apply skyline, also the wide spread of using graph databases in real-world applications like social recommendations, authorization and access control and geospatial and logistics which makes it more valuable to have skyline queries on graph databases. We argue that introducing skyline operators to graph databases is an important research point.

The paper is organized as follows. Section 2 illustrates the background related to graph model properties. In Section 3, we review the related work. Section 4 proposes how to process skyline queries inside graph databases using nested loops algorithm. Skyline queries is re-introduced using divide- and- conquer algorithm in Section 5. Experiments, in Section 6, were conducted to show different query performance using different sized datasets and variable number of dimensions. Conclusion and future work are presented in Section 7.

2. PROPERTY GRAPH MODEL

In this paper, property graph model is used which stores data in the form of a graph with nodes and directed edges between those nodes. A graph can be traversed in bidirectional way, which means that there is no need to add duplicate relationship in both directions of an edge [11]. Edges represent relationships that connect nodes. Each two nodes can have more than one relationship. Nodes have properties which describe them. Same applies to the edge properties, which describe the relationship between nodes. Nodes have labels which represent the entity they belong to or the node type. Each node may have multiple labels. Node labels group nodes together to indicate their function in the dataset. Graph size can be determined based on the number of nodes it contains in the database. Indexes on graph databases can occur on the relationships level which helps in fast data retrieval. They also support much more flexibility in updating or extending data and its structure. Thus, more properties can be added to a node and relationships. They can be easily adapted to new business needs.

3. RELATED WORK

Skyline queries help getting the best solutions based on user preferences. The preferences are represented in graph databases in the form of node/edge properties. The properties are com-pared based on some criteria: maximization or minimization. All dimensions are compared at the same time. Nodes that are less interesting to the user are dominated and excluded from the skyline result. This type of queries helps in reducing search space and saving time. Only the most interesting data is included in the result. Thus, skyline queries guarantee that the returned nodes are the ones that most satisfy user requirements.

The Skyline operator was introduced in [1], their objective was to extend relational database systems by Skyline queries. The authors extended the SQL's select statement by proposing SKYLINE OF clause. The authors showed how Skyline operator can interact with other query operators. In [2, 3] the authors focused on dynamic Skyline. The skyline result is determined based on shortest path distance that differs according to each algorithm. Aggregate skyline queries were introduced in [4]. The query combines skyline and group by operators. The experimental results showed that the query execution time is better than using the operators directly in SQL query.

In [5] the authors focused on RDF data stored using vertically partitioned schema model. They introduced an approach for optimizing skyline queries for this type of data. They focused on pruning non-skyline tuples before reaching the complete skyline processing phase. This is done using the header point concept which keeps a summary of the already visited data space regions. The authors proposed two algorithms RSJPH and RSJPH+. The algorithms are considered near complete and they help achieve the trade-off between complete skyline queries and fast response time.

The authors focused on getting skyline for road networks in [6]. Road networks consists of nodes and edges between them. The main goal was to get skyline results based on many points of interest, with two important factors: Size which represent the distance between nodes and relevance which focus on what the user exactly requested.

New parallel algorithm named SKY-MR+ was introduced in [7]. The algorithm uses MapReduce to process skyline queries. Experiments were conducted to prove the scalability and effectiveness of the algorithm.

In [8], the authors focused on the data items that have incomplete data. The dimensions to be compared in the skyline query are not presented in some data items. The authors developed an algorithm called ISkyline, which handles the missing data issue. The experiments conducted showed the efficiency and scalability of ISkyline.

Skyline results may be affected by outliers. This challenge was addressed in [9]. The authors implemented an algorithm which focuses on the degree of membership of a result to the skyline and the typicality degree. The main goal was to exclude outlier data from skyline result. In [10], the authors summarized the basic properties of skyline queries. They also discussed how they can be extended and generalized. We argue that introducing skyline queries to graph databases is an important research. This paper uses two algorithms for processing skyline queries over graph databases.

4. SKYLINE NESTED LOOPS ALGORITHM

Processing skyline queries in graph databases can be implemented using nested loops algorithm. It is applied on the whole set of nodes to be compared. It compares every single node with all the other nodes having the same label. All dimensions of each node are compared with their relative dimensions of the other nodes. The comparison is done based on the user’s query. The query can be maximum or minimum of both dimensions. The node’s dimensions are compared, if the query asks for the maximum of all dimensions, then a node that have all dimension’s values less than any other node, will be dominated.

Algorithm 1 Skyline using Nested Loops

```
inputs
    G (N, E) (Graph with nodes and edges between them), P (Node property to be returned), D
    (List of dimensions or edge properties to be compared)
outputs
    N (Final skyline nodes)
Read nodes and edges properties
Add all nodes of same label into collection N
for i in N do
    for j in N do
        for c in D do
            if all c of i > all c of j then
                remove i from N
            end if
        end for
    end for
end for
Return N
```

Algorithm 1 represents minimization for all dimensions and it can be applied on maximization. For illustration, we applied the nested loop algorithm on Neo4j [11] graph database. We implement the skyline query within cypher [12] query language using the proposed adapted algorithm and execute it on a Neo4j graph engine. We use cypher query language to represent the different skyline queries that can be generated using nested loops algorithm.

The main advantage of using nested loops algorithm is its high applicability as it can be used in any graph database being extensible to a large number of dimensions. On the other hand, it has some cons; it cannot get early skyline results, the whole dataset should be scanned before returning any skyline point which leads to query time complexity of $O(n^2)$, where n is the number of nodes to be compared inside the database. In addition, it completely relies on main memory, which may lead to many iterations given small memory capacity and large graph size. To enhance the performance of the skyline operator, divide-and-conquer skyline algorithm was also adapted to work on graph databases in the next section.

5. SKYLINE DIVIDE-AND-CONQUER ALGORITHM

This section proposes the extensions made to the divide-and-conquer algorithm to make it work efficiently with graph stores to get skyline. The divide-and-conquer algorithm is considered the best-known algorithm for the worst-case scenario [1] where no nodes are dominated. Thus, no node is better than the other in all dimensions. The whole data set is divided into sub-groups. Each group of nodes are compared together, and the final skyline result is the collection of skyline of each subgroup. It avoids re-comparing nodes that are already visited.

Algorithm 2 Skyline using Divide-and-Conquer

```
inputs
    G (N, E)(Graph with nodes and edges between them), P(Node property to be returned), D
    (List of dimensions or edge properties to be compared)
outputs
    N (List of final skyline nodes)
Calculate median of all dimensions and edges values in D
Divide nodes into blocks based on medians values
Call Nested loops algorithm for each block of nodes
Call Nested loops algorithm for each two blocks of nodes together
Merge partial skyline results
Return N
```

Divide-and-conquer algorithm solved the problem of comparing all individual nodes with all other nodes. It supports early domination by getting partial skyline from each block. Since this algorithm supports multiple dimensions d, the data can be partitioned into 2^d blocks. The complexity of the algorithm is $O(n \log^{d-1} n)$ where n is the number of nodes to be compared in the whole dataset and d is the number of dimensions. Implementing skyline queries using divide-and-conquer algorithm improves query performance according to the experimental results. The average query execution time and performance comparison for the two algorithms are presented in the next section.

6. PERFORMANCE EVALUATION

In this section, we test the performance of the two proposed skyline algorithms. By changing the environmental settings, the efficiency of the two algorithms vary. The following subsections present the experiments setup and evaluation discussions.

6.1. Experiments Setup

We used Neo4j graph database [11] to conduct the performance experiments as being the most popular graph model used in fraud detection, social networks, recommendation engines and graph-based search. The experiments are conducted on a laptop running on windows 10. The processor has a Core(TM) i5 1.70 GHz CPU and 8GB of memory. If the specifications of the machine running the experiments are changed, it will affect the results. As an example, if the memory increases, the query processing time will decrease, as both skyline algorithms rely on main memory. The two algorithms were implemented using Cypher query language. We used two different datasets for conducting the evaluation; MovieLens database [13] which consists of 10,000 nodes and a synthetic database with 1,000,000 nodes.

We transformed the tuples of the MovieLens database into nodes with properties. The transformation process is done through Cypher query language, where the query reads the database in the form of tuples stored in a CSV file, and then generates the nodes with properties. While the synthetic dataset represents hotel database. We used [14] to generate the data in the form of CSV file, and in the same way it is transformed into nodes through Cypher query language.

The MovieLens database consists of nodes with label “Movie”, each node has properties; MovieID, Name, Rating, ReleaseYear and OscarWins. In the synthetic database, two labels exist; “Hotel” and “Beach”. The relation between “Hotel” and “Beach” is “Close to” which represents the closeness of the hotel to the beach. This relationship has one property called DistancetoBeach which stores the distance between the hotel and the beach. The nodes with label “Hotel” have the properties; HotelID, HotelName, Price, PoolSize, RestaurantQuality, StarsRating, ServiceQuality and NumberOfRooms.

6.2. Varying Number of Dimensions

The number of dimensions to be compared for skyline query can greatly affect the performance. Experiments were conducted on different number of dimensions with fixed database size and average execution time was recorded. For the MovieLens database we executed the algorithm on 10,000 nodes and for the synthetic database the algorithm was executed on 1,000,000 nodes. The results are shown in Figure 2 and Figure 3 respectively.

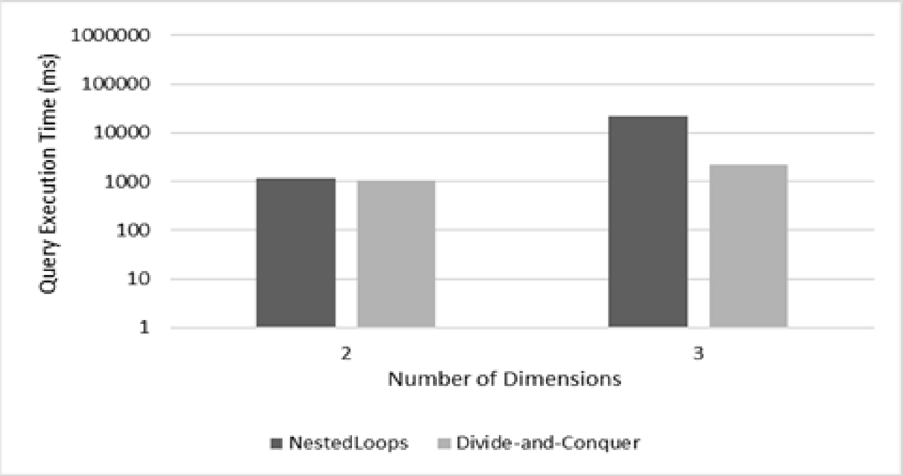


Figure 2. Comparing performance versus different number of dimensions for both skyline algorithms on MovieLens database [13]

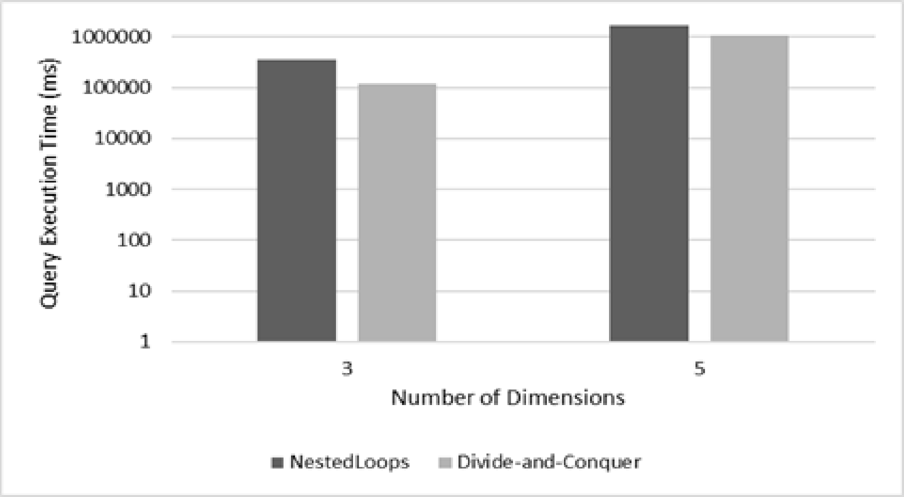


Figure 3. Comparing performance versus different number of dimensions for both skyline algorithms on the synthetic database

Query average execution time is highly affected by dimensionality, while the number of dimensions to be compared increases, the execution time increases and the difference between the two algorithms become more obvious.

6.3. Varying Dataset Size

In this experiment, the number of dimensions was fixed to 3 dimensions and we used multiple data set sizes. The results of each algorithm are shown in Figure 4 and Figure 5.

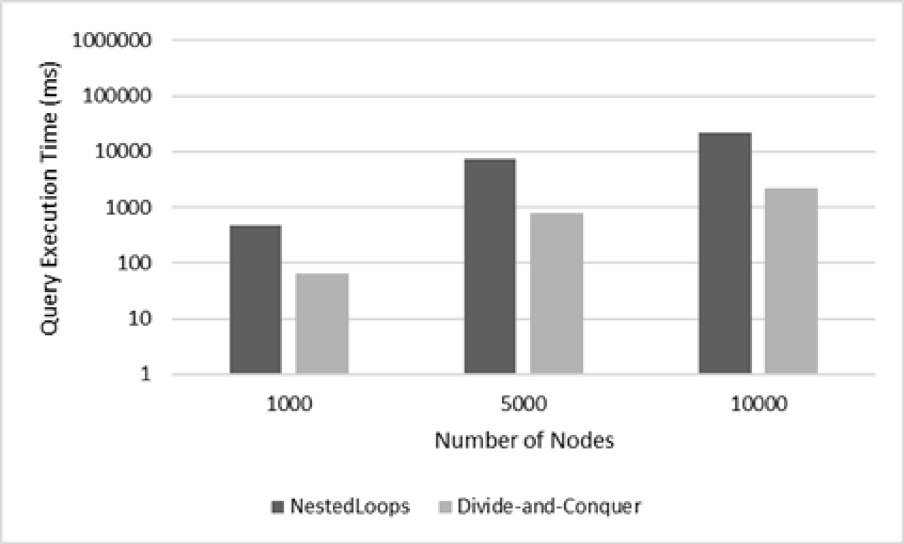


Figure 4. Comparing performance versus different dataset size for both skyline algorithms on MovieLens database [13]

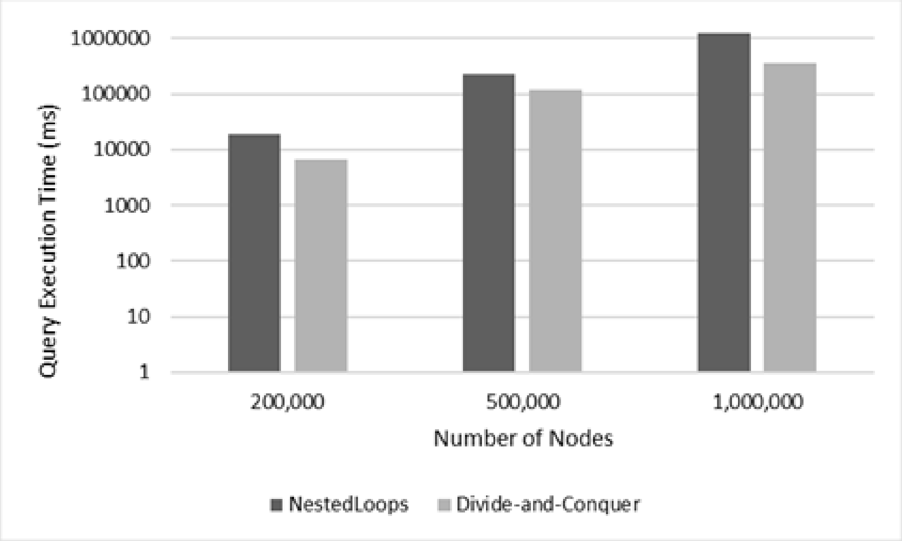


Figure 5. Comparing performance versus different dataset size for both skyline algorithms on the synthetic database

From the result of the experiments, we conclude that divide-and-conquer algorithm performs better than nested loops in all cases with varying the data set size and number of dimensions. However, with small number of dimensions the execution time for both algorithms appears to be very close, the superiority of divide-and-conquer algorithm appears more with large number of dimensions.

7. CONCLUSION AND FUTURE WORK

In this paper, different algorithms were used to process skyline queries over graph databases This type of queries is mostly important in decision making processes, data pruning and visualization. The most well-known algorithms in processing skyline queries over relational databases are nested loops algorithm and divide-and-conquer algorithm. We adapted the two algorithms to work with the graph model. The nested loop algorithm has a time complexity of O (n^2) where the divide-and-conquer has O (n $\log^{d-1}$ n). Experiments for comparing the two algorithms over graph databases were conducted. An evaluation over different databases sizes and queries was also implemented and results was presented in the paper where it shows that the divide-and-conquer algorithm showed better performance than nested loops algorithm over graph databases. As a future work, we are planning to augment graph databases with skyline operator to facilitate the operation of getting skyline results on graph databases.

REFERENCES

[1] Borzsony, S., Kossmann, D., Stocker, K. 2001. The Skyline operator. Proc. 17th Int. Conf. Data Eng. 1–20.

[2] Zou, L., Chen, L., Özsu, M.T., Zhao, D. 2010. Dynamic skyline queries in large graphs. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 5982 LNCS, 62–78.

[3] Angel C Bency. 2014. A Study on Dynamic Skyline Queries. Published in International Journal for Research in Applied Science and Engineering Technology (IJRASET).

[4] Matteo Magnani, Ira Assent. 2013. From Stars to Galaxies: skyline queries on aggregate data. Presented at the proceedings of the 16th international conference on extending database technology.

[5] Chen, L., Gao, S., Anyanwu, K. 2011. Efficiently evaluating skyline queries on RDF databases. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part II. LNCS, vol. 6644, pp. 123–138. Springer, Heidelberg.

[6] Pande, S., Ranu, S., Bhattacharya, A. 2017. SkyGraph: Retrieving Regions of Interest using Skyline Subgraph Queries. Presented at the proceedings of the VLDB Endowment Volume 10 Issue 11.

[7] Yoonjae Park, Jun-Ki Min, Kyuseok Shim. 2017. Efficient Processing of Skyline Queries Using MapReduce. IEEE Transactions on Knowledge and Data Engineering.

[8] Khalefa, M.E., Mokbel, M.F., Levandoski, J.J. 2008. Skyline query processing for incomplete data. Presented at the IEEE 24th International Conference on Data EngineeringPages 556-565.

[9] Hélène Jaudoin, Pierre Nerzic, Olivier Pivert, Daniel Rocacher. 2016. On Making Skyline Queries Resistant to Outliers. In Advances in Knowledge Discovery and Management Volume 665 of the series Studies in Computational Intelligence pp 19-38.

[10] Jan Chomicki, Paolo Ciaccia, Niccolo' Meneghetti. 2013. Skyline Queries, Front and Back. ACM SIGMOD Record Volume 42 Issue 3.

[11] Neo4j. https://neo4j.com/

[12] Cypher. http://neo4j.com/docs/developer-manual/current/cypher/

[13] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872 [2] Gizem, Aksahya & Ayese, Ozcan (2009) Coomunications & Networks, Network Books, ABC Publishers.

[14] Database test data. http://www.databasetestdata.com/

INTENTIONAL BLANK

COLLABORATIVE TRACKING IN DISTRIBUTED MULTI-SENSORS VIDEO-SURVEILLANCE SYSTEMS

Marion Sbai[1] and Samy Meftali[2] and Djamel Aouali[2]

[1]Laboratoire CRISTAL, University of Lille, France
[2]Decarte Engineering, Paris, France

ABSTRACT

Video processing applications are becoming more complex and greedy in terms of computing resources. Thus, the designers of video surveillance systems are moving more and more towards distributed systems, comprising several video sensors collaboratively working to carry out tracking tasks in particular. However, there is a plethora of collaborative tracking algorithms, in the literature, each with its own advantages and disadvantages.

The purpose of this paper is to present the most common collaborative tracking algorithms and discuss the strengths and weaknesses of each.

KEYWORDS

Video surveillance – Tracking – Camera – Multi-sensor – performances

1. INTRODUCTION

Nowadays, video surveillance becomes an absolute imperative for the security of goods and people. Thus, such systems are increasingly found in all kinds of companies and administrations, to more or less automatically perform video surveillance tasks in general and target tracking in particular.

The common feature of most detection and tracking algorithms is their high computational complexity due to amount of data that needs to be processed. This is especially important for live applications, such as video surveillance systems for threats detection or traffic monitoring, thus using one single camera in a complete and efficient video-surveillance system become almost impossible.

Recent trends in surveillance systems and the appearance of high speed wireless network protocols make possible today the use of hundreds of smart cameras in the form of a multiple sensors distributed system that perform video analysis on site on a collaborative way.

The purpose of a tracker is to associate target objects in consecutive video frames to determine their identities and locations. Multi-object tracking is one of the most fundamental tasks of high-level automated video content analysis due to its extensive application: human-machine interaction, security and monitoring, video communication and compression, traffic control and video montage.

Natarajan Meghanathan et al. (Eds) : SAI, ICAITA, CSITA, ISPR, Signal - 2018
pp. 59–73, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81006

Multi-view trackers combine data from different camera views to estimate the temporal evolution of objects in a monitored area. The data to be combined can be represented by the characteristics of the object (*such as position, color and silhouette*) or by the object's trajectories in each view.

In this context, we start with a single-target tracking review using *EKCF* and then briefly describe several approaches such as *ECKF*, *JPDA-EKCF*, *GKCF*, *Extended Information Consensus Filter (EICF)* and The *EIWCF*. In summary, we describe in detail the problems of multi-target tracking in a camera network scenario [1] with a detailed comparison between different algorithms.

The rest of this paper is organized as follow. Section 2 introduces the distributed processing in camera networks. A survey of Consensus algorithms for distributed tracking is presented in section 3. Section 4 discusses and analysis different tracking algorithms. Section 5 concludes this paper.

2. DISTRIBUTED PROCESSING IN CAMERA NETWORKS

2.1. Kalman Filter

A partially distributed target tracking approach using a cluster-based Kalman filter was proposed in [2]. Here, a camera is selected as a cluster head which aggregates all the measurements of a target to estimate its position using a Kalman filter and sends that estimate to a central camera.

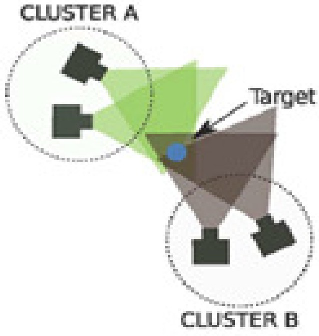


Figure 1: Multiple clusters tracking the same object in a wireless camera network.

2.2. Distributed tracking

In a distributed tracking, each camera node exchanges its estimates with its neighbors until a desired accuracy is achieved. [2]

A Distributed Kalman Consensus filter and subsequent variants have been proposed in [3], [4], [5]. It was a completely distributed solution for estimating the dynamic state of a moving target. However, there are major considerations in applying the method to camera networks due to the nature of video sensors as well as non-linearity, naivety and redundancy. Cameras are directional sensors, each with a limited view of the entire theater of action, with data having high bandwidth and complexity. We will now show how the consensus-based approaches to distributed estimation in multi-agent system literature can be applied to design a consensus-based tracking algorithm in camera networks.

3. Consensus Algorithms For Distributed Tracking

3.1. Mathematical framework

Let C be the set of all cameras in the network. We can then define the subset of all cameras viewing target T_j as $C_j^v \subset C$ and the rest of the cameras as $C_j^{v-¿¿} \subset C$. Each camera C_i will also have its set of *neighboring cameras* $C_i^n \subset C$. Based on the communication constraints due to bandwidth limitations and network connections, we define the set C_i^n as all the cameras with which C_i is able to communicate directly. In other words, C_i can assume that no cameras other than its neighbors, C_i^n, exist as no information flows directly from non-neighboring cameras to C_i. Note that the set of neighbors need not be geographical neighbors. We also define the set of *overlapping cameras* of C_i as $C_i^o \subset C$; since all the cameras can change their *PTZ* parameters and have therefore several possible fields of view, we define the set C_i^o as all the cameras with which C_i can *potentially* have an overlapping field of view. By definition, it becomes clear then that for each $C_i \in C_j^v$, it is true that $C_j^v \subset \{C_i^o \cup C_i\}$. We define $C_i^c \subset C$ as the connected component that C_i is in.We assume $C_i^o \subset C_i^c$, that is to say, C_i is able to exchange information with its overlapping cameras directly or via other cameras. A diagrammatic explanation of the notation is given in ***Fig.2.*** [6] We consider the situation where targets are moving on a ground plane and a homography between each camera's image plane and the ground plane is known. We will show how the state estimation for each target by each camera (*i.e., each camera's estimates based on its individual measurements*) can be combined together through the consensus scheme. This method is independent of the tracking scheme employed in each camera. If the network of cameras is connected, then consensus is achieved across the entire network.

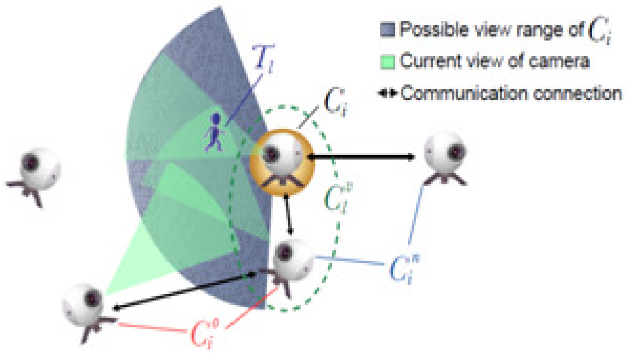


Figure 2: Conceptual illustration of camera network topologies

3.2. Algorithm of Extended Kalman-Consensus Filter for a single target

The *Extended Kalman-Consensus Filter algorithm* is developed to solve the problem of *nonlinearity* in the case of *the Kalman-Consensus filter*. This filter is a technique for estimating the state of a noise-disordered nonlinear process using multiple observations from a distributed set of detection nodes. All detection nodes attempt to estimate the same state by determining how their observations affect that state and communicating with neighboring nodes. The algorithm is designed to be more accurate due to measurement diversity, expandable to a large number of nodes, and robust against node loss during operation.

The Extended Kalman-Consensus Filter allows us to track targets on the ground plane using multiple measurements in the image plane taken from various cameras. This allows each camera C_i to have at any time step ***k***, a consensus state estimate $\acute{x}_i^j$ and estimate error covariance P_i^j for each target T_j . To model the motion of a target T_j on the ground plane, we consider a linear discrete time dynamical system

$$x^j(k+1)=A^j(k)x^j(k)+B^j(k)w^j(k),x^j(0)\quad \mathcal{N}\left(\mu_i^j,P_0^j\right).$$

And nonlinear observation model for each camera C_i ,

$$z_i^j(k)=h_i\left(x^j(k)\right)+v_i^j(k). \qquad \textbf{\textit{(eq.1)}}$$

Where:

✓ $h_i(.)$: *is the mapping from the ground to the image plane for camera* C_i.

✓ $w^j(k)\wedge v_i^j(k)$: *are zero mean white Gaussian noise.*

✓ $x^j(0)$: *is the initial state of the target.*

Due to the nonlinear nature of the observation model, the linear Kalman-Consensus Filter proposed in [5] cannot be applied as is. An extension to deal with the non-linearity of the observation model is required. Taking into account the nonlinear nature of our dynamical model, we propose an Extended Kalman-Consensus distributed tracking algorithm on the basis of the Kalman-Consensus Filter detailed in [5]. The following are our basic Kalman Filter iterations, as implemented in each camera.

✓ ***Prediction***:

$$P(k+1)=A(k)M(k)A(k)^T+B(k)Q(k)B(k)^T, \qquad \textbf{\textit{(eq.2)}}$$
$$\acute{x}(k+1)=A(k)\hat{x}(k). \qquad \textbf{\textit{(eq.3)}}$$

✓ ***Correction***:

$$\begin{aligned} M(k)^{-1}&=P(k)^{-1}+H(k)^T R(k)^{-1} H(k),\\ K(k)&=M(k)H(k)^T R(k)^{-1}, \qquad \textbf{\textit{(eq.4)}}\\ \hat{x}(k)&=\acute{x}(k)+K(k)\left(z(k)-h\left(\acute{x}(k)\right)\right). \end{aligned}$$

Here, ***P*** and ***M*** are the *a priori* and *a posteriori* estimate error covariance matrix, respectively, and ***H*** is the *Jacobian* matrix of partial derivatives of ***h*** with respect to ***x***, i.e.

$$H_{[m,n]}=\left.\frac{\partial h_{[m]}}{\partial x_{[n]}}\right|_{x=\acute{x}(k)}, \qquad \textbf{\textit{(eq.5)}}$$

This algorithm is performed at each camera node C_i. At each time step k and for each target T_j , we assume that we are given the estimated prior target state $\acute{x}_i^j(k)$ and the error covariance matrix P_i^j. At time step $k=0$, the Extended Kalman-Consensus filter is initialized with $P_i^j = P_0$ and $\acute{x}_i^j = E\langle x_i^j(0)\rangle$. The consensus algorithm is shown in *Algorithm1*.

The consensus process (*Algorithm1*) is performed at each C_i for each T_j that is in the scene viewed by the camera network. C_i^n is the neighboring camera set of C_i and defined as all cameras with which C_i can directly communicate. If C_i is viewing a target T_j , it obtains T_j 's position on its image plane z_i^j, and calculates the Jacobian matrix H_i^j of its observation model and consensus state estimate $\acute{x}_i^j$. After that, the corresponding information vector u_i^j and matrix U_i^j are computed with the given measurement covariance matrix R_i^j and H_i^j. Next, the predicted measurement g_i^j and corresponding residue r_i^j are calculated. C_i then sends a message m_i^j to its neighbors which includes the computed information matrix, residue and its estimated target state $\acute{x}_i^j$ at previous time step ***(k–1).*** Similar to [5], we define the information matrix and vector of $C_i \in C_j^{v-¿¿}$ as $U_i^j = 0$ and $u_i^j = 0$ by assuming that their output matrices are zero, i.e., $H_i^j = 0$ for all $C_i \in C_j^{v-¿¿}$ to avoid any ambiguity arising from the lack of measurements in these cameras. C_i then receives similar messages m_l from the cameras in its neighborhood. The information matrices and residues received from these messages are then fused by C_i with its own information matrix and residue and the Extended Kalman-Consensus state estimate is computed. Finally, the ground plane state $\acute{x}_i^j$ and error covariance matrix P_i^j are updated according to the assumed linear dynamical system.

3.3. Algorithm of JPDA-EKCF for tracking multiple targets

This algorithm is designed to solve the problem of data association by local measurements especially in the case of intra-camera data association i.e., to associate measurements observed by a camera with the targets as well as allows to track joint targets and track maintenance under a probability of detection and an unknown clutter rate. The *Joint Probability Data Association (JPDA)* is coupled to the *Extended Kalman Consensus Filter* (EKCF), which manages nonlinearity.

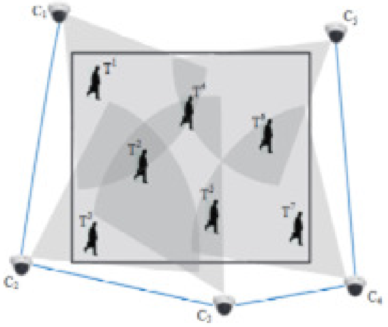


Figure 3: Tracking multiple targets with camera network.

3.3.1. Intra-Camera Association

Due to the weakness of low level video processing methods. Some targets may not be detected because of an occlusion or similar appearance in the background. A direct assignment of the measurement target can lead to poor performance *(problem of naive nodes).* The possibility of false assignment and missed target detection should be considered.

Joint Probability Data Association (*JPDA*) [7] computes an estimate over the various possibilities of measurement-to-track associations. Assume that at time step ***k***, there are $\boldsymbol{N_T}$ targets in the scene and camera C_i obtains $N_M^i(k)$ measurements, $Z_i(k)=\{z_i^1(k),\dots,z_i^{N_M^i}(k)\}$.The history of measurements at camera C_i is denoted as $\text{ƶ}\,(k)\text{¿}\{Z_i(1),\dots,Z_i(k)\}$. Let x^j denote the state of target T_j . Its *a posteriori* state estimate and *a prior* state estimate by camera C_i are denoted as $\hat{x}_i^j$ and $\acute{x}_i^j$, respectively. The state estimate of target T_j at camera C_i is:

$$\hat{x}_i^j=E[x^j|\text{ƶ}\,i]=\sum_{n=1}^{N_M^i}E\text{¿¿}$$

Where χ_i^{jn} denotes the event that measurement z_i^n associates to target T_j at camera C_i.
As an extension to standard *Joint Probability Data Association Filter* (JPDAF) **[8]**, the *Extended Kalman Filter* can be used to estimate ¿. Let us denote $\beta_i^{jn}=P(\chi_i^{jn}|\text{ƶ}\,i)$ and

$$\beta_i^{j0}=1-\sum_{n=1}^{N_M^i}\beta_i^{jn}$$

to represent the probability that target T_j has no measurement associated with it. Then the state estimate can be written as

$$\hat{x}_i^j=\beta_i^{j0}\acute{x}_i^j+\sum_{n=1}^{N_M^i}\beta_i^{jn}\left(\acute{x}_i^j+K_i^j\left(z_i^n-h_i\left(\acute{x}_i^j\right)\right)\right),$$

$$\hat{x}_i^j=\acute{x}_i^j+K_i^j\left(\sum_{n=1}^{N_M^i}\beta_i^{jn}z_i^n-\left(1-\beta_i^{j0}\right)h_i\left(\acute{x}_i^j\right)\right)\quad \textbf{\textit{(eq.6)}}$$

$$\hat{x}_i^j=\acute{x}_i^j+K_i^j\left(\tilde{z}_i^j-\left(1-\beta_i^{j0}\right)h_i\left(\acute{x}_i^j\right)\right),$$

Where

$$\tilde{z}_i^j=\sum_{n=1}^{N_M^i}\beta_i^{jn}z_i^n,$$

and

$$K_i^j=P_i^jH_i^{j^T}\left(H_i^jP_i^jH_i^{j^T}+R_i\right)^{-1}=P_i^jH_i^{j^T}\left(W_i^j\right)^{-1}.$$

H_i^j is the *Jacobian matrix* of partial derivatives of h_i with respect to x_i^j. The error covariance of the estimate is given by

$$M_i^j = P_i^j - \left(1-\beta_i^{j0}\right) K_i^j W_i^j \left(K_i^j\right)^T + K_i^j \widetilde{P}_i^j \left(K_i^j\right)^T,$$

Where

$$\widetilde{P}_i^j = \sum_{n=1}^{N_M^i} \beta_i^{jn} \left(z_i^n - h_i\left(\acute{x}_i^j\right)\right)\left(z_i^n - h_i\left(\acute{x}_i^j\right)\right)^T - \left(\tilde{z}_i^j - h_i\left(\acute{x}_i^j\right)\right)\left(\tilde{z}_i^j - h_i\left(\acute{x}_i^j\right)\right)^T,$$

While tracking target in clutter, validation gates are usually used to filter out measurements from clutter within the environment. A validation gate is a metric of "*acceptability*", i.e., within the gate, it is treated as a valid measurement, otherwise it is rejected. Let P_D be the probability that the correct measurement is detected, and P_G be the probability that the correct measurement, if detected, lies within the gate. As shown in [9], by assuming a *Poisson distribution* for false measurements lying in the gate and a *Gaussian distribution* for associating a measurement with a target, using *Bayes rule*, the β_i^j '*s* can be calculated as:

$$\beta_i^{jn} = \frac{\exp\left(\frac{-\left(v_i^{jn}\right)^T \Lambda^{-1} v_i^{jn}}{2}\right)}{b + \sum_{n=1}^{N_M^i} \exp\left(\frac{-\left(v_i^{jn}\right)^T \Lambda^{-1} v_i^{jn}}{2}\right)}$$

$$\beta_i^{j0} = \frac{b}{b + \sum_{n=1}^{N_M^i} \exp\left(\frac{-\left(v_i^{jn}\right)^T \Lambda^{-1} v_i^{jn}}{2}\right)}$$

$$b = \left(2\pi\right)^{\frac{d}{2}} \lambda_f \det\left(\Lambda\right)^{\frac{1}{2}} \left(1 - P_D P_G\right) / P_D,$$

where $v_i^{jn} = z_i^n - h_i\left(\acute{x}_i^j\right)$, Λ *is the covariance of the distribution of* ***v****,* ***d*** *is the dimension of measurement vector and* λ_f *is the expected number of occurrences of the Poisson distribution.*

3.3.2. Inter-Camera Association

In distributed tracking of multiple targets, each camera has its own set of estimated tracks and also receives track estimates from its neighbors. Therefore, it is necessary to establish an association between these tracks. This can be formulated as a maximum matching problem in a weighted bipartite graph [10] which minimizes the matching cost. The Hungarian algorithm [11] can be used to find the maximum matching. Different distance metrics can be used to find the matching cost between two track estimates from different cameras.

3.3.3. JPDA-EKCF algorithm

We now show that distributed multiple target tracking can be achieved by integrating data association with a distributed single target tracker. In [12], Joint Probability Data Association (*JPDA*) is coupled with Kalman-Consensus Filter (*KCF*) estimator, where *JPDA* is used to perform local measurement to track associations. This algorithm is referred as *JPDAKCF*. Due to the nonlinear nature of the observation model in the camera network, an extension to deal with the non-linearity is required. Here, we describe an *Extended Kalman-Consensus* Filter coupled with *Joint Probability Data Association* along the lines of the *JPDA-KCF* detailed in [12]. The entire process is shown in *Algorithm2*.

The *JPDA-EKCF algorithm* is performed at each C_i for each T_j that is in the scene under surveillance, where C_i^n is the neighboring camera set of C_i and defined as all cameras with which C_i can directly communicate. Camera C_i computes the assignment of the measurements to targets using *JPDA*. Then C_i calculates the *Jacobian matrix* H_i^j of its observation model with respect to the consensus state estimate $\hat{x}_i^j$ of last time step. After that, the corresponding information vector u_i^j and matrix U_i^j are computed with the given measurement covariance matrix R_i^j and H_i^j. Next, predicted measurements and its corresponding residues are calculated. C_i then sends a message M_i to its neighbors which includes the computed information matrices, residues and its estimated target state $\hat{x}_i^j$ and error covariance P_i^j at previous time step ***(k−1).*** C_i then receives similar messages M_l only from the cameras in its neighborhood. Based on the received information, C_i finds the inter-camera track-to-track matching's. The information matrices and residues received from these messages are then fused by C_i with its own information matrices and residues according to the cross camera track matching results and the *Extended Kalman-Consensus* state estimate is computed. Finally, the ground plane state $\hat{x}_i^j$ and error covariance matrix P_i^j are updated according to the assumed linear dynamical system.

3.4. Generalized Kalman Consensus Filter Algorithm

This approach solves the problem of naive nodes. A naive node can associate an observation with a bad target. This can affect the tracking performance of the nodes that actually observe the target by causing them to move away from their estimates. The proposed GKCF is presented in Algorithm 3. Here we first introduce the weighted mean consensus. Then we show how to integrate this consensus pattern into our framework. We then implement the Distributed Kalman Filter (DKF[1]) with the weighted mean consensus results and show how to propagate our covariance and state estimates. For the purpose of easy representation, we use W to denote the information matrix, or inverse covariance matrix, i.e., $W^{-1} = P$. In this section, we will use W^{-1} to replace P as in sections III.2 and III.3.

3.4.1. Weighted Average Consensus

Let the initial state estimate of all N_c agents be $x_i(0)$ with information matrix $W_i(0)$. As we use this information matrix term as weights in the *weighted average consensus algorithm*, the terms weight and *information matrix* will be used interchangeably. Also,

1[1] DKF: (*Distributed Kalman Filter*): Helps to reduce the disagreement of estimates by different nodes.

let $W(0)=\sum_{i=1}^{N_c} W_i(0)$.

So, the global weighted average of the initial states is

$$x^{¿}=W(0)^{-1}\sum_{i=1:N_c} W_i(0)\,x_i(0). \qquad \textit{(eq.7)}$$

Define the weighted initial state of each agent as

$$\tilde{x}_i(0)=W_i(0)\,x_i(0). \qquad \textit{(eq.8)}$$

Weighted average consensus [3] states that if the iterative update in Equations *(eq.12)* and *(eq.13)* is performed for all $i=1,\ldots,N_c$, then each of the terms $W_i(\kappa)^{-1}\tilde{x}_i(\kappa)$ tends to the global weighted average $x^{¿}$ as $\kappa\rightarrow\infty$. As a by-product, the weights also converge to the average of the initial weights. Both these properties of the weighted average consensus will be utilized in our approach.

We assume that the initial information matrix $W_i(0)$, is provided at the initial time step by the target detection mechanism. It would ideally be zero for nodes that are not detecting the target. For nodes that are detecting the target, the initial value would be $W_i(k-1)=H_i^T R^{-1} H_i$.

At the k^{th} iteration, the agents communicate with each other with the $W_i(k-1)$ and $\tilde{x}_i(k-1)$ information. Then, using the previously discussed average consensus scheme, they get an updated prior state estimate $\hat{x}_i^{-¿|k|¿}$ and weight estimate $W_i(k)$ (*see eqns. (eq.12), (eq.13) and (eq.14)*). This prior estimate tends towards the global normalized weighted average as stated before.

This approach solves the problem of *naive nodes*. A naive node can associate an observation with a bad target. This can affect the tracking performance of the nodes that actually observe the target by causing them to move away from their estimates. The proposed GKCF is presented in *Algorithm 3*. Here we first introduce the weighted mean consensus. Then we show how to integrate this consensus pattern into our framework. We then implement the Distributed Kalman Filter with the weighted mean consensus results and show how to propagate our covariance and state estimates. For the purpose of easy representation, we use W to denote the information matrix, or inverse covariance matrix, i.e., $W^{-1}=P$ In this section, we will use W^{-1} to replace P as in sections *III.2* and *III.3.*

3.4.2. Weighted Average Consensus

Let the initial state estimate of all N_c agents be $x_i(0)$with information matrix $W_i(0)$. As we use this information matrix term as weights in the *weighted average consensus* *algorithm*, the terms weight and *information matrix* will be used interchangeably. Also, let $W(0)=\sum_{i=1}^{N_c} W_i(0)$. So, the global weighted average of the initial states is

$$x^{¿}=W(0)^{-1}\sum_{i=1:N_c} W_i(0)\,x_i(0). \qquad \textbf{\textit{(eq.7)}}$$

Define the weighted initial state of each agent as

$$\tilde{x}_i(0)=W_i(0)\,x_i(0). \qquad \textbf{\textit{(eq.8)}}$$

Weighted average consensus [3] states that if the iterative update in Equations *(eq.12)* and *(eq.13)* is performed for all $i=1,\ldots,N_c$, then each of the terms $W_i(\kappa)^{-1}\tilde{x}_i(\kappa)$ tends to the global weighted average $x^{¿}$ as $\kappa \to \infty$. As a by-product, the weights also converge to the average of the initial weights. Both these properties of the weighted average consensus will be utilized in our approach.

We assume that the initial information matrix $W_i(0)$, is provided at the initial time step by the target detection mechanism. It would ideally be zero for nodes that are not detecting the target. For nodes that are detecting the target, the initial value would be $W_i(k-1)=H_i^T R^{-1} H_i$

At the k^{th} iteration, the agents communicate with each other with the $W_i(k-1)$ and $\tilde{x}_i(k-1)$ information. Then, using the previously discussed average consensus scheme, they get an updated prior state estimate $\hat{x}_i^{-¿|k|¿}$ and weight estimate $W_i(k)$ (*see eqns. (eq.12), (eq.13) and (eq.14)*). This prior estimate tends towards the global normalized weighted average as stated before.

3.5. Extended Information Consensus Filter (EICF)

This approach allows the effect of *naivety* and *nonlinearity* to be managed without requiring knowledge of other nodes in the network.

We propose two distributed filters for tracking targets in wireless camera networks, ECF1 and EIF2, which compute the local information, $y^i_{k\vee k}$, and $Y^i_{k\vee k}$, differently. *EICF1* runs at each node C_i and computes the local information values, $y^i_{k\vee k}$ and $Y^i_{k\vee k}$, based on their own respective measurement information, i^i_k and I^i_k:

$$Y^i_{k\vee k}=Y^i_{k\vee k-1}+I^i_k \qquad \textbf{\textit{(eq.17)}}$$
$$y^i_{k\vee k}=y^i_{k\vee k-1}+i^i_k$$

and then exchange the values $y^i_{k\vee k}$ and $Y^i_{k\vee k}$ with neighbours to achieve average consensus.

EICF2 computes local information values, $y^i_{k\vee k}$ and $Y^i_{k\vee k}$, based on its own measurement information and also that of neighbouring nodes:

$$Y^{i}_{k\vee k}=Y^{i}_{k\vee k-1}+\sum_{j\in\left|c_i\cup C_i^{n}\right|} I^{j}_{k} \qquad \textbf{(eq.18)}$$

$$y^{i}_{k\vee k}=y^{i}_{k\vee k-1}+\sum_{j\in\left|c_i\cup C_i^{n}\right|} i^{j}_{k}$$

EICF2 reaches convergence faster than *EICF1*, at the cost of additional communication to send the measurement information terms. Hence, *EICF2* is recommended when sufficient communication resources are available.

The iterative information exchange between neighbours results in redundancy which causes correlation among the nodes' estimates. Hence, the *EICF* results are sub-optimal because of such correlation among the individual node estimates. In the update step (*see Eq. 10*) of a filter, the two terms involved are the priors, $y^{i}_{k\vee k-1}$ and $Y^{i}_{k\vee k-1}$, and the measurement information about the target, i^{i}_{k} and I^{i}_{k}. The prior information is the result of the prediction on previous estimates, $y^{i}_{k\vee k}$ and $Y^{i}_{k\vee k}$,, which are computed after consensus. Hence, the redundancy always lies in the prior information terms, $y^{i}_{k\vee k-1}$ and $Y^{i}_{k\vee k-1}$.

3.6. Extended Information Weighted Consensus Filter (EIWCF)

This algorithm uses the EIWCF to handle the three main problems (*naivety, redundancy and non-linearity*). However requires knowledge of the number of cameras N_c, the basic principle of these algorithms is to weight the estimates of the nodes according to their covariance information. When N_c is not available, EICF can be used at the cost of not managing the redundancy problem.

Via proper weighting of prior and measurement information, IWCF mitigates the problem of redundancy [13]. By applying the concept of *IWCF* to *EIF*, we propose a non-linear distributed filter called the Extended Information Weighted Consensus Filter *(EIWCF).* Here the prior information is weighted by $1/N_c$ and the consensus proposals are prepared as:

$$v^{i}_{k\vee k}=\text{¿}\ \frac{1}{N_c}y^{i}_{k\vee k-1}+i^{i}_{k}$$

$$V^{i}_{k\vee k}=\text{¿}\ \frac{1}{N_c}Y^{i}_{k\vee k-1}+I^{i}_{k} \qquad \textbf{(eq.19)}$$

After achieving consensus on the v and V terms, the results are multiplied by N_c:

$$y^{i}_{k\vee k}=N_c\, v^{i}_{k\vee k}$$

$$Y^{i}_{k\vee k}=N_c\, V^{i}_{k\vee k} \qquad \textbf{(eq.20)}$$

These estimates are not affected by non-linearity, naivety and redundancy. However, *EIWCF* requires the knowledge of the number of nodes in the network (*see Eqs. 12 and 13*). Thus, *EIWCF* cannot be applied when such knowledge is not available whereas *EICF1* or *EICF2* can be used. If sufficient communication resources to receive neighbours' measurement information, $\boldsymbol{i}_k^j$ and $\boldsymbol{I}_k^j$, is available, *EICF2* achieves faster convergence than *EICF1*. Hence the choice depends on the available communication resources.

4. ANALYSIS AND COMPARISON

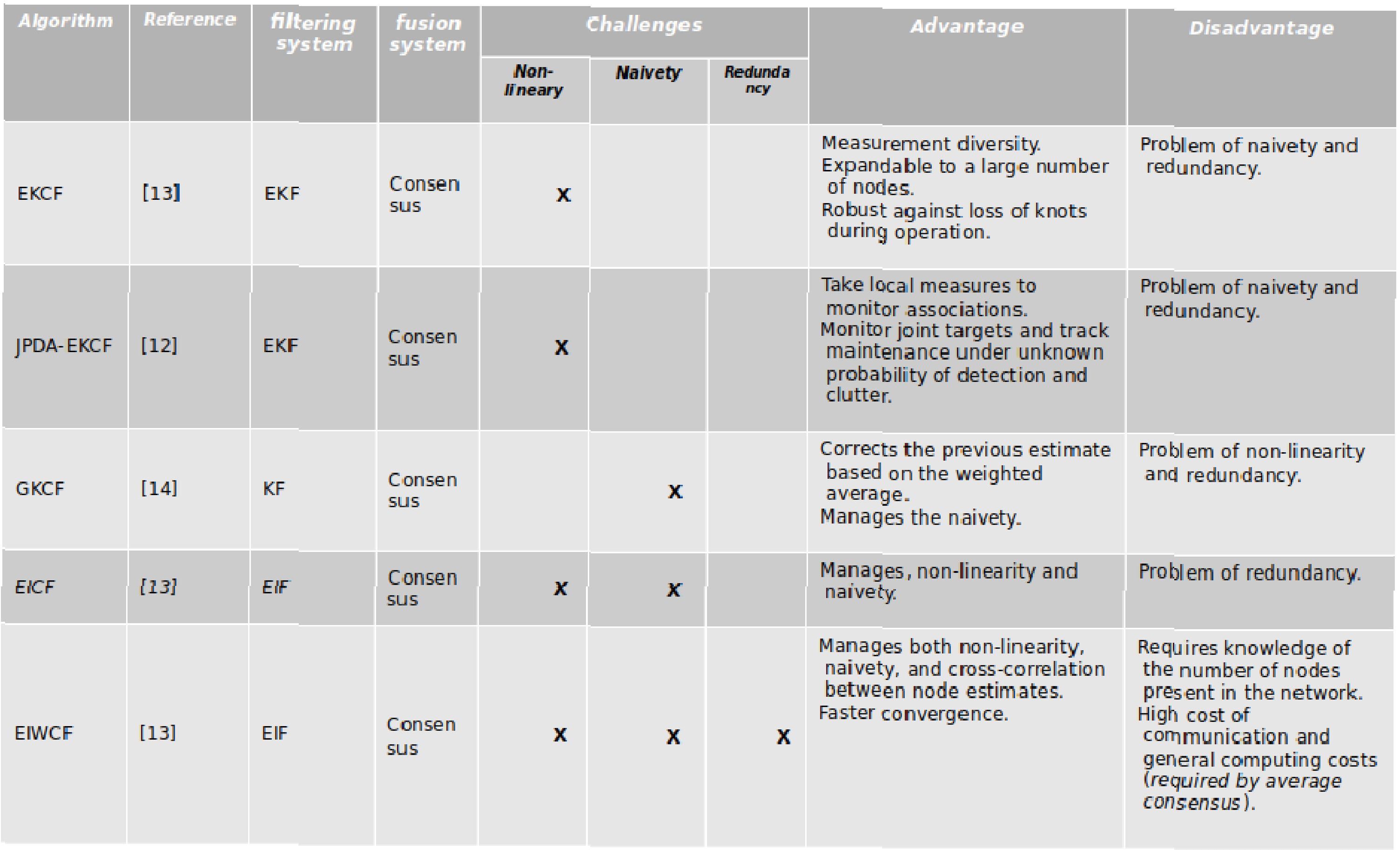
Based on our proposed approaches, there have been three major issues in consensual distributed tracking for camera networks that are as follows:

- ✓ Non-linearity.
- ✓ Naivety.
- ✓ Redundancy.

The *Kalman Extended Consensus Filter* (EKCF) algorithm is extensible to a large number of nodes, robust against the loss of nodes during operation, and has measurement diversity. The JPDA-EKCF makes local measurements to track associations, track joint targets, and track maintenance under unknown probability of detection and clutter. However, these filters do not deal with naivety and redundancy in a camera network, but the GKCF handles naivety and corrects the previous estimate according to the weighted average but it does not deal with nonlinearity and redundancy.

Afterwards, we proposed an *Extended Information Consensus Filter* (EICF). This filter performs a weighted averaging while addressing the problem of *naive nodes* and *nonlinearity*. To overcome the *redundancy problem*, we have also proposed a weighted consensus extended information filter (EIWCF). The EIWCF handles *naivety*, *redundancy* and *non-linearity*, and achieves faster convergence by correctly weighting past and measurement information. However, it requires knowledge of the number of nodes in the network.

The table below (*see Fig 4*), summarizes a detailed comparison between the different algorithms based on references.



Algorithm	Reference	filtering system	fusion system	Challenges			Advantage	Disadvantage
				Non-lineary	Naivety	Redundancy		
EKCF	[13]	EKF	Consensus	x			Measurement diversity. Expandable to a large number of nodes. Robust against loss of knots during operation.	Problem of naivety and redundancy.
JPDA-EKCF	[12]	EKF	Consensus	x			Take local measures to monitor associations. Monitor joint targets and track maintenance under unknown probability of detection and clutter.	Problem of naivety and redundancy.
GKCF	[14]	KF	Consensus		x		Corrects the previous estimate based on the weighted average. Manages the naivety.	Problem of non-linearity and redundancy.
EICF	*[13]*	*EIF*	Consensus	x	x		Manages, non-linearity and naivety.	Problem of redundancy.
EIWCF	[13]	EIF	Consensus	x	x	x	Manages both non-linearity, naivety, and cross-correlation between node estimates. Faster convergence.	Requires knowledge of the number of nodes present in the network. High cost of communication and general computing costs (*required by average consensus*).

Figure 4 :Algorithms for distributed tracking in camera network.

5. CONCLUSION

Video tracking can be defined as a problem of locating a moving object (*or multiple objects*) over time based on the observations of the object in the images. In other words, the purpose of a tracker is to associate target objects in consecutive video frames to determine their identities and locations. Multi-object tracking is one of the most fundamental tasks of high-level automated video content analysis through its extensive applications. Maintaining the stability of tracks on multiple video targets over extended periods and extended areas remains a difficult problem. Among the most basic monitoring methods are the *Kalman filter* and the *JPPAF filters*, we presented a distributed state estimation method based on the *Generalized Kalman Consensus Filter* (GKCF), which has exceeded the KCF approach under such conditions. However, in themselves, these methods are generally not able to follow extended spatial horizons. Since the measurement model of a camera is non-linear, algorithms based on the *Kalman filter* can not be used. Nonlinear filters such as the *Kalman Extended Consensus Filter* (EKCF) do not deal with naivety and redundancy. To overcome the redundancy problem, we have also proposed the *Extended Information Weighted Consensus* (EIWCF) filter by combining the *Extended Information Filter* (EIF) and the *Information-weighted Consensus Filter* (IWCF). The EIWCF handles naïveté, redundancy and non-linearity, and achieves faster convergence by correctly weighting past and measurement information. However, it requires knowledge of the number of nodes in the network.

As future work, we will explore the reduction of communication and computational overhead required by average consensus. The management of dynamic link structure and asynchronous networks are other possible future works.

REFERENCES

[1] Samuel Davey, Neil Gordon, Ian Holland, Mark Rutten, Jason Williams. “Bayesian Methods in the Search for MH370”, Commonwealth of Australia 2016.

[2] W. Li, W. Zhang “Multiple target localization in wireless visual sensor networks” Front. Comput. Sci., pp. 496-504. 7 (4) 2013.

[3] Donato Di Paolaa, Antonio Petittia, Alessandro Rizzob. “Distributed Kalman Filtering via Node Selection in Heterogeneous Sensor Networks”. ISSIA-Bari, Italy; September 2013.

[4] Kar, J. M. F. Moura, and K. Ramanan, “Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication”. IEEE Transactions on Information Theory, vol. 58, no. 6, pp. 3575–3605, 2012.

[5] S. Das and J. M. F. Moura, “Distributed Kalman filtering and network tracking capacity” in 47th Asilomar Conference on Signals, Systems, and Computers, pp. 629–633. 2013.

[6] Amit K. Roy-Chowdhury, Bi Song.“Camera Networks: The Acquisition and Analysis of Videos over Wide Areas”, University of California, Riverside, 133 pages. January 2012.

[7] Xiao Chen, Yaan Li, Yuxing Li, Jing Yu and Xiaohua Li “A Novel Probabilistic Data Association for Target Tracking in a Cluttered Environment”. Northwestern Polytechnical University, China, December 2016.

[8] M. Chandrajit, R. Girisha, T. Vasudev and M. Hemesh. “Data Association and Prediction for Tracking Multiple Objects”, Indian Journal of Science and Technology, Vol 9 (33). September 2016.

[9] Taek Lyul Song ; Hyoung Won Kim ; Darko Musicki. “Iterative joint integrated probabilistic data association for multi-target tracking”, IEEE Transactions on Aerospace and Electronic Systems Vol.51, April 2015.

[10] Abul K. M. Azad, Mohammed Misbahuddin. “Web-Based Object Tracking Using Collaborated Camera Network” . Northern Illinois University, DeKalb, IL, USA. Apr 27, 2018.

[11] Humayra Dil Afroz, Dr.Mohammad Anwar Hossen. ”New Proposed Method for Solving Assignment Problem and Comparative Study with the Existing Methods”. Journal of Mathematics (IOSR-JM). Volume 13, Issue 2 Ver. IV, pp 84-88. Mar. - Apr. 2017.

[12] Subhro Dasy, José.M, F. Mouraz, “Distributed estimation of dynamic fields over multi-agent networks”, NY 10598, USA – Jan. 2017.

[13] Ziren Wang, Guoliang Liu, Guohui Tian, “Human skeleton tracking using information weighted consensus filter in distributed camera networks”, Chinese Automation Congress (CAC), Oct. 2017.

[14] Nemanja Ilié, Khaled Obaid Al Ali, Milos S. Stankovic and Srdjan S. Stankovic, “Distributed Multitarget Tracking in Camera Networks Using Multi-step Consensus”, Proceedings of 4th International Conference on Electrical, Electronics and Computing Engineering, Kladovo, Serbia, June 05-08. 2017.

INTENTIONAL BLANK

IMPROVED LSB BASED IMAGE STEGANOGRAPHY USING RUN LENGTH ENCODING AND RANDOM INSERTION TECHNIQUE FOR COLOR IMAGES

G. G. Rajput and Ramesh Chavan*

Department of Computer Science,
Rani Channamma University, Belagavi, KA, India 591156

ABSTRACT

Image Steganography is a technique for securing the secret message using a cover image in such a manner that the alterations made to the image are perceptually indiscernible. In this paper a novel method for secret message hiding in color images is proposed. The message is encoded by extracting the RGB components of a color image. Run length encoding is performed on the data and insertion of the data in least significiant bits(LSB) of the pixel is guided by linear congruential generator (LCG). A 3R-3G-2B LSB pattern is recommended for insertion of the data making the information more secure without bringing any significant distortions to the original image. The experiments performed on various color images demonstrate the efficacy of the proposed algorithm in terms of PSNR of cover image and that of stego-image.

KEYWORDS

cover, secret message, LSB, LCG, RLE, stego-image.

1. INTRODUCTION

Image Steganography allows for two parties (sender and intended receiver) to communicate secretly and covertly. The general principle underlying the image steganographic method is to embed the secret message in the image without bringing change in the characteristics of the image. Assuming that, an attacker has unlimited computation power and is able and willing to perform a variety of attacks, it should not be possible for the attacker to decode the message (Visual Attacks, Enhanced LSB Attacks, Chi-Square Analysis, and other statistical analyses). The embedding method should be such that, the stego-image(information coded image) should not reveal the existence of secret image/message. One of the approaches to code the secret message in an image is to place the secret message in the noise component of a signal. If it is possible to code the information in such a way that it is indistinguishable from true random noise, an attacker has no chance in detecting the secret communication. However, such an approach is not suitable for noise-free images. The simplest way of hiding information in an image is to replace the least significant bit (LSB) of every element(pixel) with one bit of the secret message. Since flipping the LSB of a byte (or a word) only means the addition or subtraction of a small quantity, the

Natarajan Meghanathan et al. (Eds) : SAI, ICAITA, CSITA, ISPR, Signal - 2018
pp. 75–82, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81007

sender assumes that the difference will lie within the noise range and that it will therefore not be generally noticed. However, the approach is not secure since an attacker can extract the LSBs and simply "decode" the cover, just as if he were the receiver. Instead, an approach of inserting the information bits in LSBs of randomly selected elements will make the information more secure. However, the intended receiver should be aware of the procedure of random selection to retrieve the secret message. The key to this may be sent by the sender through secret channel (eg. personal email) to the intended receiver. On the other side, the key may be embedded in the one of the elements of the image and the information regarding the same may be sent to intended receiver through secret channel. In this paper, we propose to use this approach for hiding the secret message in LSBs of the color image. To make the system more secure, first we perform run length encoding on the secret message, secondly, perform angular rotation of the color image and then use a pre-defined pattern for message insertion in elements of the RGB components of the color image. After the insertion of message, lastly, we perform reverse angular rotation on the imageto obtain a stego-image (image with a secret message).

2. LITERATURE SURVEY

Many techniques have been proposed in the literature for hiding messages in images such that the alterations made are indiscernible in the generated stego-image. The spatial domain techniques manipulate directly the pixel bit values to embed the secret message (eg. LSB, pixel-value differencing). The secret bits are written directly to the cover image pixel bytes making it easy to implement. Consequently, the spatial domain techniques are simple and easy to implement. The transform domain techniques involve image transformation such as cosine transformation, Fourier transform and wavelet transformation. However, there are techniques that share the characteristic of both of the spatial domain and transform domain (eg. pattern block encoding, spread spectrum methods and masking). The fact that, the resulting images should be statistically indistinguishable from untampered images has been studied in the form of PSNR values.

A review on image stegnographic techniques is presented in [4,5].

Aura [6] has introduced a flexible scheme applicable to random access covers, especially to digital images. He developed a secret key steganography system based on pseudorandom permutations. Due to the construction of the scheme, the secret information is distributed over the whole cover in a rather random manner.

A protocol which allows public key steganography has been proposed by Anderson in [7, 8]; it relies on the fact that encrypted information is random enough to "hide in plain sight". If the stego-message is not targeted towards a specific person, but for example is posted in an Internet newsgroup, the problem worsens. Although the protocol also works in this case (only the intended receiver can decrypt the secret message, since only he has the correct private key) all possible receivers have to try to decode every posted object.

Ajit Danti et.al [9] have proposed a 2-3-3 LSB insertion method, where in eight bits of secret data is inserted in LSB of RGB (Red, Green and Blue) pixel values of the cover image in 2,3,3 order, respectively, to embed a color secret image into a cover image.

Chin□Feng Lee et.al [11] scheme performs the logical Exclusive□OR (XOR) operation to smoothen the secret bit stream and to embed the result into a cover medium. Additionally, the proposed scheme employs generalized difference expansion transform for image recovery after data extraction; consequently, the image fidelity can be preserved.

The Least Significant Bit (LSB) is one of the main techniques in spatial domain image Steganography. Many of the proposed algorithms in the literature are based on LSB insertion methods because of the fact that an altered image with slight variations in its colors, in LSB positions of the color pixels, will be indistinguishable from the original by a human being, just by looking at it. However, a simple LSB implementation is vulnerable to attacks [13]. Hence, extended implementation of LSB method are proposed in the literature [14,15,16].In RGB based steganography, the R, G, and B components(channels) are treated as independent bytes and LSB substitution is applied.

Parvez and Gutub [17] proposed RGNB based technique. The idea in that, for insignificant colors, significantly more bits can be changed per channel. For example, suppose in a pixel with R=55, G=255 and B=255, a change in the R channel will not show a significant distortion. The lower color value of a channel has less effect on the overall color of the pixel than the higher value. Therefore, more bits can be changed in a channel having ‘low’ value than a channel with a ‘high’ value. However, the choice of pixels is straight forward and the capacity is unpredictable. In the technique proposed by Gutub et al. [18], the RGB image is used as cover media and the cipher text is hidden inside the image using a pseudorandom number generator (PRNG) thereby including more randomization in selection of pixels. The PRNG produces two new random numbers per iteration, say seed1 and seed2. The seed1 random number is used to determine the RGB component where cipher text will be hidden and seed2 determines the number of bits that can be hidden in it. However, the capacity is unpredictable due to the choice of seed2 value. Kaur et al. [19] proposed a RGB intensity based algorithm in which variable number of bits are hidden in different channels. The LSBs of one of the three channels is used as indicator and data is stored in other two channels. The advantage in this technique is usage of 4 LSBs in some of the data channels, which increases the hiding capacity. Both security and capacity is enhanced.

In this paper, we propose an RGB based LSB insertion in a way that the text message is secured and not vulnerable to attacks. The variation of LSB method is proposed using run length encoding scheme and random selection of pixels. A specific fixed pattern is defined for choice of number of LSBs in each of R, G, and B channels. Moreover, the insertion of secret message is done by performing angular rotation of the cover image and reversing back to its original position after insertion making the scheme more secure.

3. PROPOSED METHOD

Digital images are recorded as a matrix or array of small picture elements, or pixels. Each pixel is represented by a numerical value. In general, the pixel value is related to the brightness or color. In case of color digital images, the commonly used color space is RGB. In RGB cube model, a pixel in a color image possesses three components; Red (R), Green (G), and Blue (B). Each component comprises of 8 bits. These R, G, and B components (channels) can be treated as independent bytes and LSB substitution can be applied. In simplest LSB substitution, it means 3 data bits can be hidden in one pixel. However, it is not wise to implement in this form, since such approach is vulnerable to attacks for secret message retrieval. The method proposed in this paper is described below.

Hiding the Secret Message (Data Hiding):

The cover image is a color image with 24 bits per pixel described in RGB color space. The secret text message is binarized and stored as stream of bits. Run length encoding is performed on the stream of bits [12].Angular transformation is performed on the cover image and the three channels, R, G, and B, respectively, of the cover image are extracted and Run Length Encoded data is inserted in the LSBs of the pixels of the channels in the following pattern: 3 LSBs of R channel, 3 LSBs of Green channel and 2 LSBs of Blue channel- a total of 8 bits are used per color pixel. However, the choice of pixel is based on linear congruential generator (LCG). Given a seed, LCG generates a sequence of pseudo random numbers which are taken as pixel positions in the channels and the sequence is followed to insert the secret data in LSBs positions in pattern specified. The number of pixels used for inserting the data is recorded in the last pixel of the cover image. After the insertion, reverse angular transformation is performed to generate the final stego-image. The algorithm for generating stego-image is presented below. The seed value (stego-key) used for LCG method is sent to the intended receiver through a secure medium.

Step 1. Read the cover medium i.e. color image.

Step 2. Read the secret message(text), perform runlength encoding and then binarize.

Step 3. Compare size of binarized secret data against size of cover image to ensure that the cover image is not distorted after embedding. For example, for true image 24bit of size 20x20 pixels, (8 bits/ pixel) 3200bits of binarised data can be embedded using LSB technique.

Step 4. A sequence of random positions is generated using LCG method with a choice of seed value. These positions represent the pixel positions in the channels of color image.

Step 5. Starting from the first random position of pixel, insertion of data is performed in 3R-3G-2B pattern

Step 6. The number of pixels used for inserting the is written in LSB of the last pixel of the image.

Step 7. Reverse angular transformation is performed to retain original position of the cover.

Step 8. Output the stego image

Secret Message Retrieval

The process of retrieving the secret message from stego-image is presented below.

Step 1. Read the stego image.

Step 2. Using stegokey (seed value),generate the sequence of random numbers representing the position of the pixels used for inserting text in RGB channels. Following the pixel positions, read the data bits in 3-3-2 pattern and store it in the array. The number of pixels to read is known from the data embedded in last pixel of the stego- image.

Step 3. Perform run-length decoding on the extracted bits.

Step 4. Output the secret message.

4. EXPERIMENTAL RESULTS

Windows wallpapers are used to implement the proposed method. The wallpaper images have resolution of 1920x 1200 pixels, 24-bit true color. The quality of the stego-image is measured in terms of parameters, namely, Mean-Squared Error (MSE) and Peak Signal-to-Noise Ratio(PSNR) [20].

The mean-squared error (MSE) between two images g(x,y) (cover image) and $\hat{\mathbf{g}}$(x,y)(stego-image), is defined as

$$\mathrm{E_{MSE}} = \frac{1}{\mathrm{MN}}\sum_{\mathrm{n=1}}^{\mathrm{M}}\sum_{\mathrm{m=1}}^{\mathrm{N}}[\ \hat{\mathrm{g}}(\mathrm{x,y}) - \ \mathrm{g(x,y)}]^2\text{------------------------}(1)$$

where mean-squared error depends strongly on the image intensity scaling, PSNR scales MSE according to image range and is given by

$$\mathrm{PSNR} = \ -10\ \log_{10}\frac{\mathrm{e_{MSE}}}{\mathrm{S}^2}\ \text{--}(2)$$

where S is the maximum pixel value.

The Structural Similarity Index (SSIM) quality assessment index is based on the computation of three terms, namely the luminance term, the contrast term and the structural term. The overall index is a multiplicative combination of the three terms.

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma}\text{----------------------}(3)$$

Where,

$$l(x,y) = \frac{2\mu_x\mu_y+C_2}{\mu_x^2+\mu_y^2+C_1}\text{----------------------------------}(4)$$

$$c(x,y) = \frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2}\ \text{-----------------------------------}(5)$$

$$s(x,y) = \frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3}\text{------------------------------------}(6)$$

Where µx, µy, σx,σy, and σxy are the local means, standard deviations, and cross-covariance for images *x, y*. If α = β = γ = 1 (the default for Exponents), and C3 = C2/2 (default selection of C3) the index simplifies to:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}\text{-------------------------}(7)$$

The stego-image obtained for sample images are shown Figure 1. The corresponding MSE, PSNR values and SSIM values are tabulated in Table 1. A subjective test was also performed by asking the selected viewers to compare the images before and after information hiding.

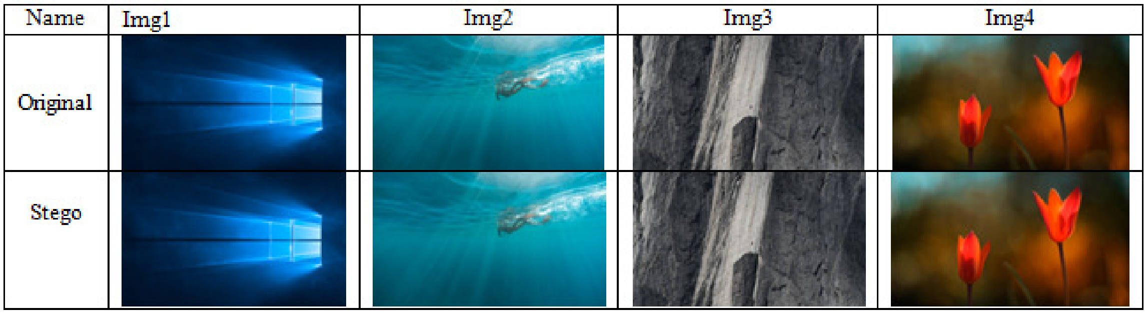


Figure 1. Original image & Stego-Image

Table 2:MSE, PSNR & SSIM of image

	MSE			**PSNR**			**SSIM Value**
Image	**R**	**B**	**G**	**R**	**B**	**G**	
Img1	0.00	0.00	0.00	78.9340	76.8051	79.6121	1.0000
Img2	0.00	0.00	0.00	81.4715	76.8606	79.5288	1.0000
Img3	0.0	0.00	0.00	78.5079	76.5043	79.1556	1.0000
Img4	0.00	0.00	0.00	78.8163	76.9480	80.0724	1.0000

5. CONCLUSION

An efficient method based on RGB steganography is presented in this paper. The secret message is embedded in the RGB channels of the cover image in a specific pattern i.e. 3-3-2. The positions of the pixels are chosen at random using LCG. The security of the data is ensured by first performing run-length encoding on the secret message and this run length encoded bits are inserted in the cover image by performing angular rotation of the image. Reverse angular rotation is performed to generate stego-image. The specific pattern 3-3-2, the seed value used in generating random pixel positions and angular rotation forms the stego-key which is send to the intended receiver using a secure medium. The performance of the proposed method is noted in terms of PSNR and it is observed that the alterations made are indiscernible in the generated stego-image. Our proposed algorithm is targeted to achieve increased text embedding capacity into the cover image followed by ensuring high security of the secret message.

REFERENCES

[1] Foley, J., et al., Computer Graphics, Principles and Practice, Reading, MA: Addison Wesley, 1990

[2] N.F. Johnson, S.C. Katzenbeisser, “A survey of steganographic techniques”, in: S. Katzenbeisser, F.A.P. Petitcolas (Eds.), Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Inc., Norwood, 2000.

[3] N.F. Johnson, S. Jajodia, “Exploring steganography: seeing the unseen”, IEEE Computer 31 (2) (1998) 26–34.

[4] A. Cheddad, J. Condell, K. Curran, and P.M. Kevitt, “Digital image steganography: survey and analysis of current methods”, Signal Processing, vol. 90, pp.727-752, 2010.

[5] Gandharba Swain, Saroj Kumar Lenka, Classification of Image Steganography Techniques in Spatial Domain: A Study, International Journal of Computer Science & Engineering Technology (IJCSET),5(3), pp 219-233, 2014

[6] Aura, T., "Practical Invisibility in Digital Communication," in Information Hiding: First International Workshop, Proceedings, vol. 1174 of Lecture Notes in Computer Science, Springer, 1996, pp. 265–278

[7] Anderson, R. J., "Stretching the Limits of Steganography," in Information Hiding: First International Workshop, Proceedings, vol. 1174 of Lecture Notes in Computer Science, Springer, 1996, pp. 39–48.

[8] Anderson, R. J., and F. A. P. Petitcolas, "On The Limits of Steganography," IEEE Journal of Selected Areas in Communications, vol. 16, no. 4, 1998, pp. 474–481

[9] G.R. Manjula, Ajit Danti,” A Novel Based Least Significant Bit (2-3-3) Image Steganography in Spatial Domain”, Intenational journal of security, privacy and Trust Management(IJSPTM) Vol.4 No 1 february 2015.

[10] R.Z. Wang, C.F. Lin, J.C. Lin, “Image hiding by optimal LSB substitution and genetic algorithm”, Pattern Recognition 34 (3) (2001) 671–683.

[11] Chin-Feng Lee, Chi-Yao Weng, Aneesh Sharma, ”Steganographic access control in data hiding using run-length encoding and modulo-operations” SECURITY AND COMMUNICATION NETWORKS ; 9:139 –148 Published online 16 June 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.333.

[12] Rafael C. Gonzalez, Richard E. Woods, “Run-Length Encoding”,” Digital Image Processing”, 3rd edition, Chapter 8, section 8.2.5, pp.553-559, 2011.

[13] C. K. Chan, and L. M. Chang, “Hiding data in images by simple LSB substitution”, Pattern Recognition, vol.37, pp.469-474, 2004.

[14] M. A. B. Younes, and A. Jantan, “A new steganography approach for image encryption exchange by using least significant bit insertion”, International Journal of Computer Science and Network Security, vol.8, no.6, pp.247-254, 2008.

[15] H. B. Kekre, A. A. Athawale, and P. N. Halarnkar, “Increased capacity of information hiding in LSB’s method for text in image”, International Journal of Electrical, Computer and System Engineering, vol.2, no.4, pp.246-249, 2008.

[16] G. Swain, and S. K. Lenka, “LSB array based image steganography technique by exploring the four least significant bits”, CCIS, Vol. 270, part II, 2012, pp.479-488.

[17] M. T. Parvez, and A. A. Gutub, “RGB intensity based variable-bits imagesteganography”, in Proceedings of IEEE Asia-pacific Services Computing Conference, 2008, pp.1322-1327.

[18] A. Gutub, A. Al-Qahtani, and A. Tabakh, “Triple-A secure RGB image steganography based on randomization”, in Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, 2009, pp.400-403.

[19] M. Kaur, S. Gupta, P. S. Sandhu, and J. Kaur,“A dynamic RGB intensity based steganography scheme”, World Academy of Science, Engineering and Technology, vol.67, pp.833-836, 2010.

[20] Krenn,R.,“Steganograph and Steganalysis”, http://www.krenn.nl/univ/cry/steg/article.pdf.

[21] MSE &PSNR, http://in.mathworks.com/help/vision/ref/psnr.html.

[22] SSIM, http://in.mathworks.com/help/images/ref/ssim.html.

[23] G. G Rajput, Ramesh Chavan, “A Novel Approach for Image Steganography Based on LSB Technique”, International Conference on Compute and Data Analysis Proceedings ICCDA '17, May 19-23, 2017, Lakeland, FL, USA © 2017 Association for Computing Machinery, ACM ISBN 978-1-4503-5241-3/17/05.

[24] G. G Rajput, Ramesh Chavan “A Novel Approach for Image Steganography Based on Random LSB Insertion in Color Images”, Proceedings of the International Conference on Intelligent Computing Systems (ICICS 2017 – Dec 15th – 16th 2017), India, Elsevier’s SSRN eLibrary – Journal of Information Systems & eBusiness Network – ISSN: 1556-5068.

SIMULATION AND MODELING OF ANN-BASED PROGNOSIS TOOL FOR A TYPICAL AIRCRAFT FUEL SYSTEM HEALTH MANAGEMENT

Vijaylakshmi S. Jigajinni[1] and Vanam Upendranath[2]

[1]Department of Electronics and Communication Engg., Basaveshwar Engineering College, Bagalkot-587 102, Karnataka, India
[2]Aerospace Electronics and Systems Division, CSIR-National Aerospace Laboratories, Bengaluru-560017, Karnataka, India

ABSTRACT

The ability to predict the aircraft fuel system health/operating condition and possible complications that occur during the long flight of an aircraft helps to improve the performance of the aircraft engine. Prognostics and Health Management (PHM) methodology includes fault detection, diagnosis, and prognosis. In this paper, we propose an Artificial Neural Network (ANN) based fault prognosis tool for a typical aircraft fuel system. Prognostics method using ANN's promise to provide a new approach to manage the fuel flow and fuel consumption of aircraft engine more effectively. This method identifies the presence of faults and mitigates them to maintain a proper fuel flow to the engine. Overlooking the presence of any faults in time could potentially be catastrophic which can lead to possible loss of lives and the aircraft as well. The developed tool works on the logical rules developed as per the engine's fuel consumption and quantity of fuel flow from the tanks. Here, we discuss the algorithm and the results of using ANN models to predict the health condition of the fuel system of aircraft.

KEYWORDS

AIRCRAFT FUEL SYSTEM, ANN, FAULT ANALYSIS, DIAGNOSIS, PROGNOSIS, HEALTH MANAGEMENT

1. INTRODUCTION

Prognostics and Health Management (PHM) is the study of breakdown mechanisms and lifecycle management of a system [1]. It is a method that helps to assess the consistency of a system under its operating conditions to analyze the time of failure and mitigate the system risks [2]. An aircraft is a complex system of system operating as a group of interrelated systems [3]. Every aircraft system is responsible for safe operation.

Prognostics is the process of prediction based on present and prior conditions. Diagnostics pertains to the recognition and separation of faults or failures [4, 5]. The goal of prognostics is to assess the overall future healthiness or condition of a system. It also deals with the prediction of the quality of a system including the Remaining Useful Life (RUL) of the system. In an aircraft, fuel to the engine is made to flow through fuel pipelines. A malfunction in any of the components, like, leakage in tanks, pump breakdown, pipeline leakage, and stuck valve *etc.,* may lead to improper functioning of the fuel system as well and can result in the failure of the mission.

Natarajan Meghanathan et al. (Eds) : SAI, ICAITA, CSITA, ISPR, Signal - 2018
pp. 83–93, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81008

In this work, a simulation model is developed to monitor and manage the health condition with a rule-based prognostics mechanism thus helping to make such predictions possible. The process of prognostics is a mathematical computation mechanism that predicts the future health of a complex system, fuel system in this context, based on the amount of past and current data available. The ultimate predictions made are based on data collected from multiple tanks with warnings, alerts, and safety measures. Continuous availability of useful data facilitates in improving the ability to diagnose and predict the effective functional life of a system. As the complexity of a given system increases, it also makes identification, isolation and finding the root cause of a fault in the system very difficult [6], thus increasing the work of the maintenance engineers. With these increasing demands on the safety of systems and dependability, a broad range of fault detection, diagnostic and prognostic methodologies have been projected in the literature [7].

Artificial Intelligence (AI) techniques based on neural networks are effective for modeling the complete health management of aeroplane fuel system. An ANN model can imitate a non-linear relationship between the required input and predicted output with good precision [8]. ANN is trained properly before it is used to model as per our required input-output relationship of the fuel system. Automatic updates of ANN model consider the data for any changes in working conditions of the considered system [9]. This study focuses on proper management of the flow of fuel to the engine by isolating the faults and mitigating them using this ANN prognosis tool.

2. ARTIFICIAL NEURAL NETWORK (ANN)

The main aim of the proposed prognostic model is to build a feed-forward mechanism using Artificial Neural Networks, to regulate input parameters to obtain the desired results. Learning and training process of the input-output patterns of the fuel system is done by a rule-based mechanism. This method helps to learn and adapt not only from environmental changes but also from changes in the output *i.e.* fuel consumption by the engine.

Different types of sensors are installed in an aircraft system. As sensors become smaller and smarter, the use of such sensors helps to gather a large volume of data which can be processed for prognostics [10]. Artificial Neural Networks models match with the biological neural systems that process parallel information [11]. ANN consists of two layers connected to the peripherals: an input layer to collect the data and an output layer to represent the result of the network. An example of a simple neural network is as shown in Figure 1. $X_1,\ldots,X_n$, represents the 'n' number of input signals and *Wk1,…, Wkn*, represents the weights associated with each signal. These weighted inputs are added in a summing junction and an output *Yk* is obtained through the activation function *F*.

In this neural network model, the summation function aggregates a weighted sum of inputs and the activation function converts the sum into the final output of the network [12]. Among the different training methods, Back Propagation(BP) is the most efficient one. Learning in the neural network is achieved by collecting the information in the form of training the data set. The weights are considered based on the type of training algorithm adopted.

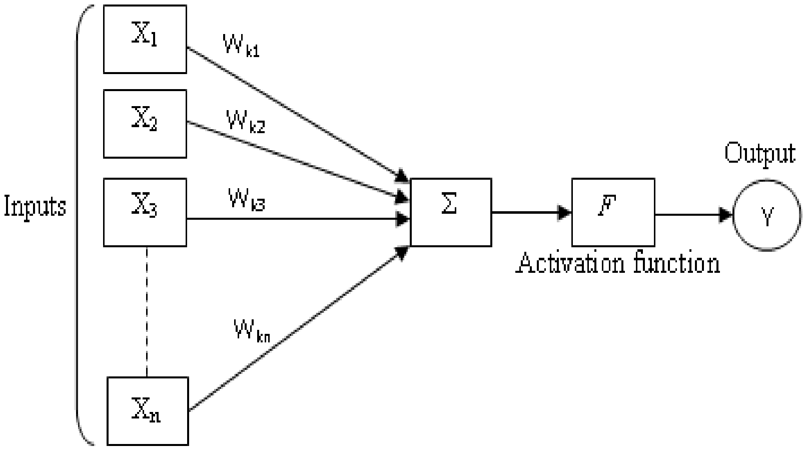


Figure1. A Neural Network model

This prognostic model includes four layers; an input layer, two intermediate hidden layers and an output layer of neurons. The feed-forward neural network equations for each step are as shown:

$$\mathrm{V_k} = \sum_{j=1}^{n} Wkj\ Xj \tag{1}$$

$$Y'(k) = S(V_k) \tag{2}$$

$$Y = \theta\left(\sum_{k=1}^{n} Y'(k)\right) \tag{3}$$

Proper training of the neural network model once done can be used for any type of incomplete or new data. The response obtained give predictions based on the inputs and adjusted weights accordingly. The prognostics engine uses input data (the fuel flow rate) and historical information (previous engine consumption rate) to train the ANN model for making predictions in relation to a working condition. The output function is described as:

$$V_k = f\,(W_{k1},\ W_{k2},\ \ldots\ ,\ W_{kn}) \tag{4}$$

The model with the least error level was considered by comparing results by training the model with a different number of layers with multiple iterations.

3. SIMULATION OF THE PROPOSED PROGNOSIS TOOL

Figure 2 shows the block diagram of the prognosis tool with aircraft fuel tanks, pumps and pipeline routes. Generally, the fuel tanks in the aircraft are in the aircraft's fuselage and wings [13]. A typical small aircraft fuel system model is simulated in the Simulink, by considering eight centrifugal fuel pumps. Out of eight fuel pumps, two pumps are used for fuel delivery between the left and right wings and two other pumps for backup for any emergency conditions and remaining four main pumps for fuel delivery to the engines.

The primary objective of this work is to monitor continuously the fuel flow to the engines without any restrictions, to reach the required fuel consumption rate. Any fault occurred in the fuel tanks is detected and mitigated by the ANN-based controller. In a fuel system, there are various parameters which change due to change in the altitude of the aircraft. For example, ambient temperature variations can cause the water contaminants in the fuel to condense and settle at the bottom of the fuel tanks. Later ice crystals may form blocking the filter which interrupts the flow of fuel to engines. But the unique characteristics of ANNs can learn such data variations with the inbuilt rules for the given system. ANNs also maintain long-term memory and distinguish patterns even in changing environments, changing altitudes and noisy surroundings.

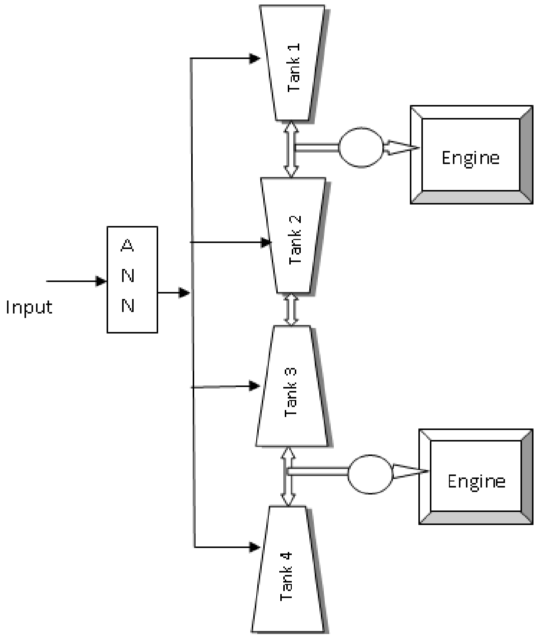


Figure 2. Block diagram of the prognosis tool for a typical aircraft fuel system

Because of these changing features of the fuel system of the aeroplane, ANNs are promising methods for prognostics [14]. This prognosis tool is used to manage and monitor the fuel system and to control the fuel flow as per the fuel consumption rate of the engines. It performs fault detection and the corresponding predictions and suggestions are made so as to maintain constant and required fuel flow rate to the engine throughout the flight.

In this paper, a fuel flow rate prediction model is projected using multilayer Feed-Forward Neural Networks (FFNN). The input-output relation of the FFNN model with two inputs and a single output is as shown in Figure 3.

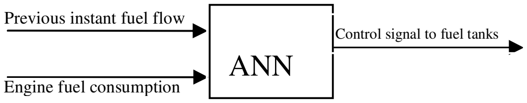


Figure 3. I/P - O/P relation of ANN Model

Back Propagation is an effective training algorithm to minimize the output error. During the process of operation of the fuel system, the BP algorithm measures and calculates the gradient of the error and adjusts the weights of the neural model with respect to the required fuel flow rate. Thus, the ANN prediction model generates the necessary control signals to fetch the required fuel flow rate to the engine. Figure 4 shows an approach for updating process of the ANN model. For maintenance of fuel system, the maintenance engineers generally follow a scheduled maintenance regime. Timely maintenance keeps the working condition of the fuel system within the required range of operation. Any leakage in tanks, pumps failure or other faults can alter the operation or may lead to damage of aircraft. Therefore, it is necessary to continuously update the ANN model with the current data, to maintain the required fuel flow rate.

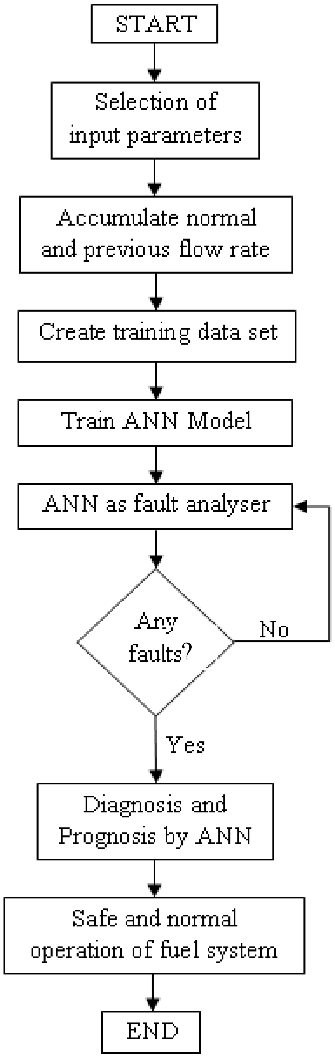

START
Selection of input parameters
Accumulate normal and previous flow rate
Create training data set
Train ANN Model
ANN as fault analyser
Any faults?
No
Yes
Diagnosis and Prognosis by ANN
Safe and normal operation of fuel system
END

Figure 4. Flowchart of ANN based prognostic tool for fuel system

A Fuel Management System (FMS) gives fuel measurements based on distance to travel, wind and time. When an aeroplane is programmed for a flight route, the fuel monitoring and management system have a capability of displaying the total flight endurance, amount of fuel available and an estimation of remaining fuel. The fuel display in the cockpit can be unreliable if there are tank leaks, pipeline leaks, components failure or plumbing malfunctions [15]. The main task of the fuel management system is to provide the estimation of fuel for the complete flight. This estimation in the FMS is obtained by actual rate of fuel consumption and amount of fuel available in the fuel tanks. In the current FMS, maintenance cost is high and need to check the proper functioning of all subsystems to maintain actual fuel flow rate. Any anomaly in the process leads to the catastrophic damage to the system.

Some of the general factors faced during the process of fuel management are fuel exhaustion, fuel starvation, and fuel contamination. Fuel starvation is an onboard condition wherein the engines will not receive any information regarding the availability of fuel. Fuel exhaustion is another condition where the aeroplane's engines are running out of fuel because of some malfunction in the fuel system. Presence of foreign particles like water, surfactants, dirt in the fuel cause fuel contamination which may lead to engine breakdown through damaging or the blocking of the fuel system subcomponents [16]. Hence, this ANN prognosis tool helps to detect and diagnose the occurrence of any kind of faults, which is not possible with the programmed fuel management system. Also, with the proposed tool, redundant components in the fuel system can be reduced.

4. Simulation Results and Discussion

The ANN-based prognostic tool is implemented in MATLAB/Simulink. In this work, the model of the aircraft fuel system is simulated similar to the methodology of the paper [17]. The fuel management process is visualized using the ANN-based prognostic tool. The simulated model of a typical aircraft fuel system is as shown in Figure 5. Simulink model of the fuel tank, fuel pump, fuel line and geometry of the aircraft fuel tank are simulated and details of the same are available in the paper [18]. For simulation, the fuel assumed is the liquid Hyjet-4A of which the characteristics are available in the simulink toolbox. The fuel temperature of 22.72°C and the viscosity of 1 are assumed respectively. The Simulink model of aircraft fuel pipeline with an internal diameter of 10mm geometry factor of 64 is built similar to the actual pipelines with metal pipes. An axial-centrifugal pump with electric driven motor is modeled and opted in the place of the actual fuel pump. During simulation of the fuel pump, the angular velocity of 1770 rpm and the correction factor of 0.8 are set.

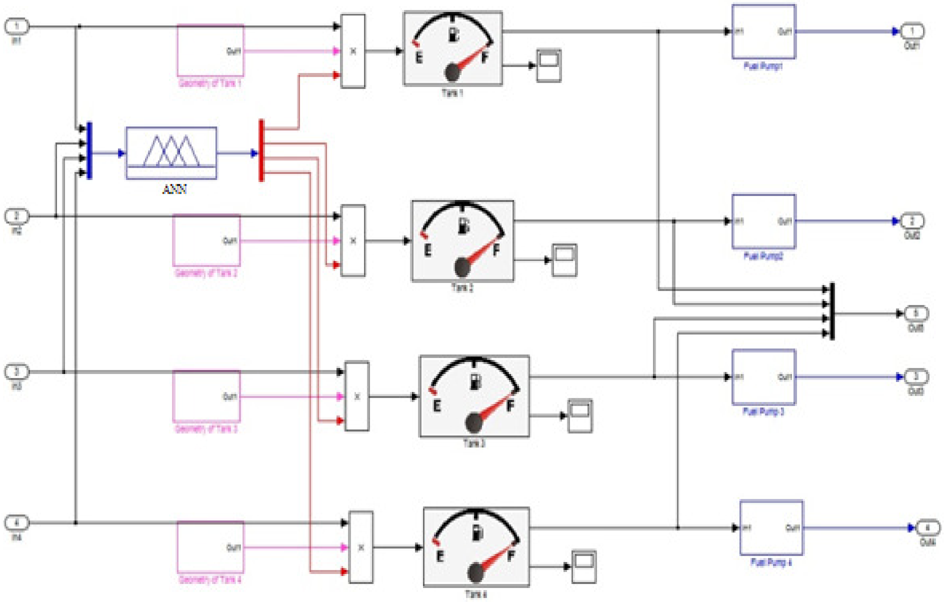


Figure 5. The Simulink model of the aircraft fuel system with ANN as a controller

The fuel system is exposed to inertia, vibration, fluid, and load of aircraft during operation which has to be considered without breakdown. The content of fuel tank(s) should provide at least 30 minutes of continuous engine operation with full power. The Simulink model of a simple four tank fuel system is designed along with fuel pumps, pipelines, and fuel indications. As a controller, the ANN-based prognostic engine is connected. It detects the fault occurrences and takes necessary action to correct by training the neurons according to the input parameters. The output generated from the ANN model are the control signals obtained based on the previous instant flow rate of fuel and rate of fuel consumed by the engines. Thus, the control signal fetches the required rate of fuel to the engine(s) without any change, irrespective of any anomalies during operation. It takes a few minutes and/or hours to visualize the fuel leaks because usually, fuel has a slow evaporation rate.

Hence, it becomes difficult to identify fuel leaks immediately. The effectiveness of this method is evaluated by the recognized results with the proposed ANN prognostic technique. Twenty seconds of simulation time is used in this model. The fuel management test result without a controller is depicted in Figure 6a and the fuel consumption requirement is illustrated in Figure 6b.

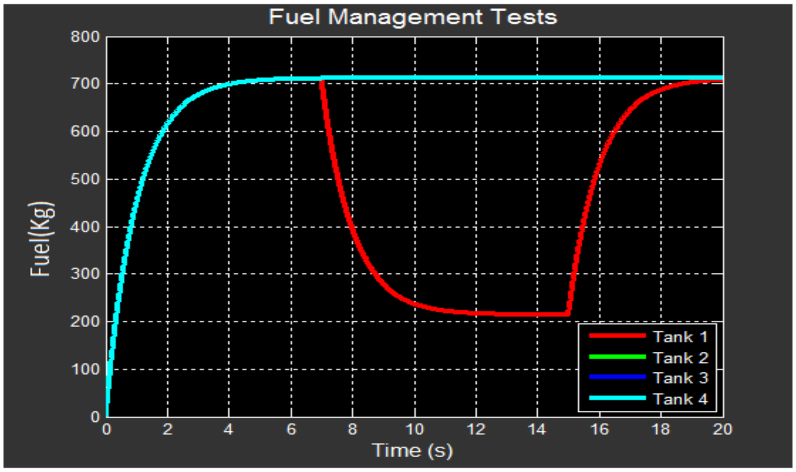


Figure 6a. Fuel management in the aircraft fuel system without a controller

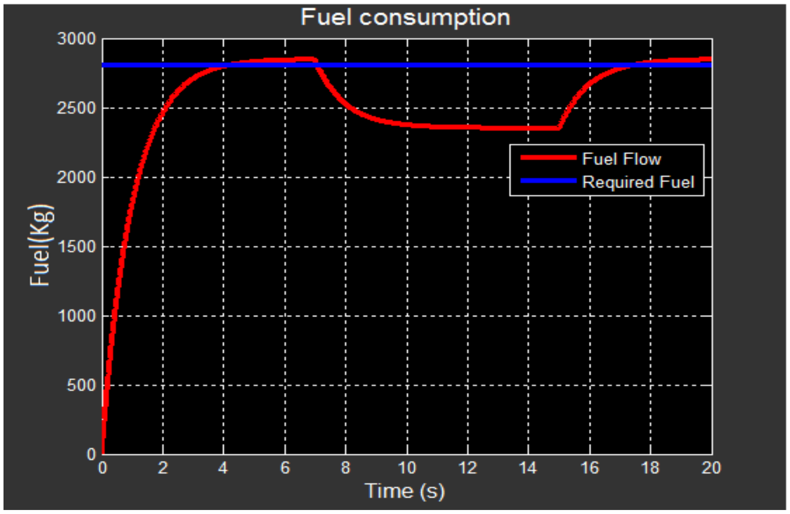


Figure 6b. Fuel consumption in the engine of the aircraft fuel system without a controller

From Figure 6b it is clear that the required fuel for a small aircraft fuel system considered is about 2800 kg/hr, which is fulfilled by four fuel tanks with each of 700 kg/hr within 4 to 6.5 seconds. After 4 seconds, the level of fuel in one of the tanks is reduced due to the faults. During the simulation, the fuel level is reduced intentionally by changing the inputs of ANN accordingly. Thus, the delivery of the fuel and fuel flow rate to the engine is affected. This sudden decrease of fuel (fault) is not correctly identified by the automatic or programmed fuel management system.

Therefore, the performance of the fuel system gets affected by the change in the rate of fuel flow. Figure 7 shows the fuel management tests using the ANN controller as prognosis tool, which detects the decrease of fuel level in tank 1 and diagnoses it, by fetching the required rate of fuel from other tanks. Thus, this technique helps to maintain the fuel flow rate and avoid unnecessary landing of aircraft or any other kind of critical situations.

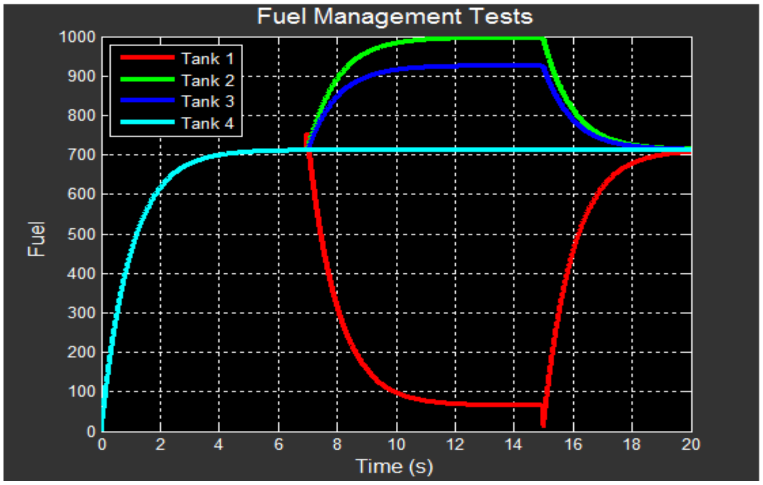


Figure 7. Fuel management in the aircraft fuel system using ANN

The approach of ANN technique as a prognostic tool to manage the fuel is more efficient method compared to other programmed fuel management systems. It detects the time of the fault, diagnoses it and also takes the corrective steps to mitigate it, by fetching the required fuel from the other remaining tanks. The weight updation process of the BP algorithm used by the ANN technique identifies the occurrence of faults and corrects to maintain the required fuel rate. This tool can manage fuel flow of 2600kg/hr as shown in Figure 8.

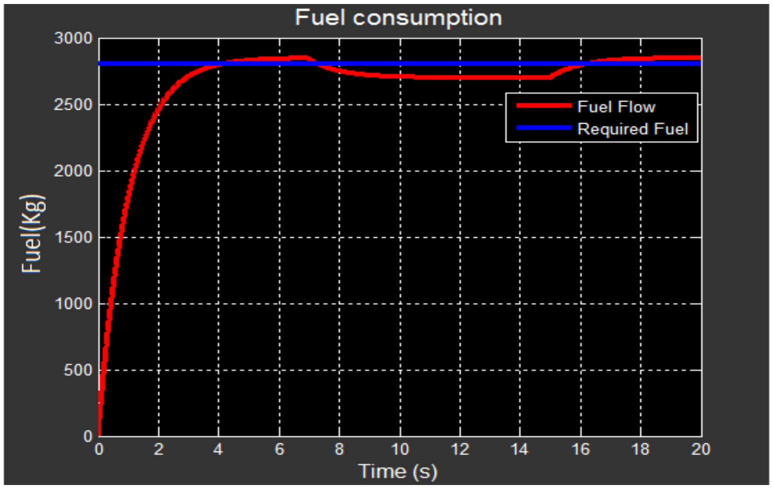


Figure 8. Fuel consumption in the engine of the aircraft fuel system using ANN

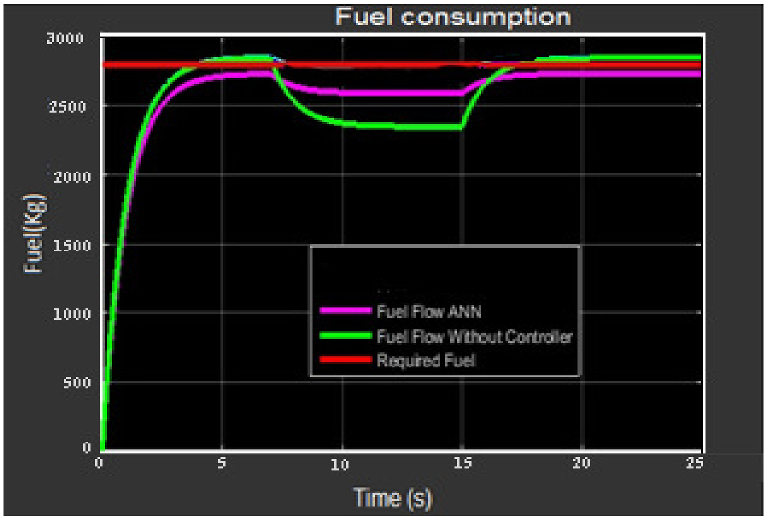


Figure 9. Comparison of fuel consumption

From the comparison result, as shown in Figure 9, the proposed method effectively detects the fault in the fuel tank and manages the fuel requirement of the aircraft engine, with management tests performed without any controller.

6. CONCLUSION

Prognostics is a process of failure analysis followed by the health prediction of the system. We have developed an artificial neural network-based fault prognosis tool for a typical four tank aircraft fuel subsystem in this paper. This method using ANN promises to deliver and manage the fuel flow and helps to monitor the fuel level in each tank of the fuel system. The proposed prognosis tool identifies the presence of faults, mitigates them and maintains the proper fuel flow to the engine at the required fuel consumption rate by generating the proper output signal. The efficiency of the simulated model is verified through a comparison with the same fuel system without a controller. From the comparison analysis, it is shown that this prognostic tool employs a unique and effective methodlogy to detect, diagnose and mitigate the fault conditions. The tool is simulated in MATLAB and Simulink for a laboratory environment.

REFERENCES

[1] Serdar Uckun, Kai Goebel, and Peter J.F. Lucas, “Standardizing research methods for prognostics”, 2008 International Conference on Prognostics and Health Management.

[2] Michael Pecht, “Prognostics and health management of Electronics”, Wiley 2008G.

[3] Biswas Gautam, Gyula Simon, Nagabhushan Mahadevan, Sriram Narasimhan, John Ramirez and Gabor Karsai, “A robust method for hybrid diagnosis of complex systems”, in Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes, pp.1125-1131, 2003.

[4] Inseok Hwang, Sungwan Kim, Youdan Kim and Chze Eng Seah, “A survey of fault detection, isolation, and reconfiguration methods”, IEEE Transactions on Control Systems Technology, Vol.18, No.3, pp.636-653, 2010.

[5] Nikhil M. Vichare and Michael Pecht, “Prognostics and health management of electronics”, IEEE Transactions on Components and Packaging Technologies, Vol 29, No. 1, March 2006.

[6] Isermann Rolf and Peter Balle, “Trends in the application of model-based fault detection and diagnosis of technical processes”, Control engineering practice, Vol.5, No.5, pp.709-719, 1997.

[7] R. Isermann, "Supervision, fault-detection and fault diagnosis methods - an introduction", Control Engineering Practice, Vol. 5, No. 5, pp. 639-652, (1997).

[8] Talebi H A and K Khorasani, “A neural network-based multiplicative actuator fault detection and isolation of nonlinear systems”, IEEE Transactions on Control Systems Technology, Vol.21, No.3, pp.842-851, 2013.

[9] Tayarani-Bathaie Seyed Sina, Zakieh Nasim Sadough Vanini and Khashayar Khorasani, “Dynamic neural network-based fault diagnosis of gas turbine engines”, Neurocomputing, Vol.125, No.11, pp.153-165, 2014.

[10] Zhang Xiaodong, Thomas Parisini and Marios M Polycarpou, “Sensor bias fault isolation in a class of nonlinear systems”, IEEE Transactions on Automatic Control, Vol.50, No.3, pp.370-376, 2005.

[11] S. S. Haykin, Neural networks and learning machines, 3. ed. Upper Saddle River: Pearson Education, 2009.

[12] Shen Ting, Fangyi Wan, Weimin Cui, and Bifeng Song, “Application of prognostic and health management technology on aircraft fuel system”, In Prognostics and Health Management Conference of IEEE, pp.1-7, 2010.

[13] Jimenez Juan F, Jose M Giron-Sierra, C Insaurralde and M Seminario, “A simulation of aircraft fuel management system”, Simulation Modelling Practice and Theory, Vol.15, No.5, pp.544-564, 2007.

[14] M. Yu, D. Wang, M. Luo, and L. Huang, “Prognosis of hybrid systems with multiple incipient faults: Augmented global analytical redundancy relations approach,” IEEE Trans. Syst., Man, Cybern. A Syst., Humans, vol. 41, no. 3, pp. 540–551, May 2011.

[15] www.flightlearnings.com/2017/08/02/fuel-management-systems Date of access:18/6/18.

[16] “Aircraft fuel system” chapter 14 published by Federal Aviation Administration (FAA).

[17] Robert Breda, Vladimir Beno, “Modeling of the control circuit of aircraft fuel system”, Przegląd Elektrotechniczny, Vol.89, pp.172-175, 2013.

[18] Vijaylakshmi Jigajinni, Upendranath Vanam, “ANFIS based fault diagnosis tool for a typical small aircraft fuel system” Part of the Advances in Intelligent Systems and Computing book series (AISC, volume 479) ISBN: 9789811017087 (online) 9789811017070 (print) DOI: 10.1007/978-981-10-1708-745.

AUTHORS

Mrs. Vijaylakshmi S. Jigajinni obtained Bachelor’s degree in Instrumentation Technology from Visveshvaraya Technological University (VTU) of Belgaum-590018, Karnataka, India in the year 2003 and Master’s degree in Digital Communication from the same university during 2009. Currently, she is an Assistant Professor at Department of Electronics and Communication Engineering of Basaveshwar Engineering (Autonomous) College, Bagalkot, affiliated to VTU, Belagavi, Karnataka, India. Her areas of interests include Artificial Intelligence, Sensors, and Control systems.

Dr. Vanam Upendranath obtained his Master's degree in Electronics from REC/NIT Warangal, India in 1981, and Ph.D. from University of Trento, Italy in 2005. He was a Scientist in Electronics Systems Area at Central Electronics Engineering Research Institute (CSIR-CEERI), Pilani during 1983- 2010. He was also a Visiting Researcher at the ECE Dept., Johns Hopkins University, the USA during his Ph.D. tenure. From 2010 onwards he has been associated with Integrated Vehicle Health Management (IVHM) program at National Aerospace Laboratories (CSIR-NAL), Bangalore. His areas of interest include Embedded Systems, Wireless Sensor Networks and IVHM for aerospace applications.

AUTHOR INDEX