KIET JOURNAL OF COMPUTING AND INFORMATION SCIENCES

ISSN: 2616-9592

Volume: 3 | Issue: 1 | Jan - June | 2020

KIET JOURNAL OF COMPUTING AND INFORMATION SCIENCES

Volume 3, Issue 1, 2020
ISSN: 2616-9592
Frequency Bi-Annual

Editorial Board

College of Computing & Information Sciences
Karachi Institute of Economics & Technology

College of Computing & Information Sciences

Vision

To develop technology entrepreneurs & leaders for national & international market

Mission

To produce quality professionals by using diverse learning methodologies, aspiring faculty, innovative curriculum and cutting edge research, in the field of computing & information sciences.

AIMS AND SCOPE

KIET Journal of Computing and Information Sciences (KJCIS) is the bi-annual, multi-disciplinary research journal published by **College of Computing & Information Sciences (CoCIS)** at **Karachi Institute of Economics and Technology (KIET)**, Karachi, Pakistan. **KJCIS** aims to provide a panoramic view of the state of the art development in the field of computing and information sciences at global level.

It provides a premier interdisciplinary platform to researchers, scientists and practitioners from the field of computing and information sciences to share their findings and contribute to the knowledge domain at global level. The journal also fills the gap between academician and industrial research community.

KJCIS focused areas for publication includes; but not limited to:

- Data mining
- Big data
- Machine learning
- Artificial intelligence
- Mobile applications
- Computer networks
- Cryptography and information security
- Mobile and wireless communication
- Adhoc and body area networks
- Software engineering
- Speech and pattern recognition
- Evolutionary computation
- Semantic web and its application
- Data base technologies and its applications
- Internet of things (IoT)
- Computer vision
- Distributed computing
- Grid and cloud computing

OPEN ACCESS POLICY

For the benefit of authors and research community, this journal adopts open access policy, which means that the authors can self-archive their published articles on their own website or their institutional repositories. The readers can download or reuse any article free of charge for research, further study or any other non profitable academic activity.

PEER REVIEW POLICY

Peer review is the process to uphold the quality and validity of the published articles. KJCIS uses double-blind peer review policy to ensure only high-quality publications are selected for the journal. Papers are referred to at least two experts as suggested by the editorial board. All publication decisions are made by the journal's Editors-in-Chief on the basis of the referees' reports. We expect our Board of Reviewing Editors and reviewers to treat manuscripts as confidential material. The identities of authors and reviewers remain confidential throughout the process.

COPYRIGHT

DISCLAIMER

The opinions expressed in **KIET Journal of Computing and Information Sciences (KJCIS)** are those of the authors and contributors, and do not necessarily reflect those of the journal management, advisory board and the editorial board. Papers published in KJCIS are processed through double blind peer-review by subject specialists and language experts. Neither the **CoCIS** nor the editors of **KJCIS** can be held responsible for errors or any consequences arising from the use of information contained in this journal, instead; errors should be reported directly to the corresponding authors of the articles.

Academic Editorial Board

Advisory Board

Table of Content

Impaired Glove for Blind and Impaired Person

Ummay Faseeha [1] Samia Ghazala [2] Bushra Rahmani [3]

Rabiya Rafique [4] Suman Taba [5] Najam Us Sehar [6]

Abstract

Blind people come across many challenges in their routine activities which need attention. Mobility without any assistance is their main issue. They can't go anywhere without any support. Furthermore, they also face difficulties in learning. Therefore, proposed system is developed to resolve the issues associated with the blind community or visually impaired people. Thus, embraces a subjective and subsequent quantitative request to comprehend the academic difficulties confronted by blind children in secondary school and universities, their determination techniques and the utilization of innovation. A portable glove is designed that supports visionless people in their routine activities. With the help of this glove, blind can go anywhere without assistance. Navigation feature is also provided in case of obstacles. Obstacles are captured and detected by camera and notified through voice. Additionally, mobile application named as *"Impaired Glove"* is developed for continuous tracking of the sightless person and for object detection.

Keyword: Blind students, visually impaired, academic challenges, education.

1 Introduction

There are about two million visually impaired or blind persons [1]. People with visual impairment fall more often as compared to normal sighted people. They may fail to see or over correct in stepping over environmental hazards and may have difficulty taking corrective action after a stumble. They can't go anywhere without any assistance. This paper is based on a working project, impaired glove, designed to encounter the discussed problems faced by blind people. This glove can assist a blind person anywhere. The glove has the ability to navigate the path, measure distance and tell the blind if there is any hurdle or barrier by generating alarm. "Impaired Glove Mobile Application" is another module of the project which uses mobile camera to take image of object. After capturing image, object detection is performed and name of the detected object is notified to the blind. Impaired glove can also be used by blind students for study purpose such as to detect shapes and recognize text. Shape recognition mode is utilized to learn about the edges of the item and the surface mode enables the user to feel varieties in the surface of a picture. Voice email feature is also provided in mobile application which helps user to email by voice.

[1] *Jinnah University for Women, Karachi | ummay.faseeha@gmail.com*
[2] *Jinnah University for Women, Karachi | samia_ghazala@yahoo.com*
[3] *Jinnah University for Women, Karachi | bushrarahmani45@gmail.com*
[4] *Jinnah University for Women, Karachi | rabiyasheikh83@gmail.com*
[5] *Jinnah University for Women, Karachi | summantaba@gmail.com*
[6] *Jinnah University for Women, Karachi | najamussehar18@gmail.com*

The rest of the paper is arranged as follows. Section II covers background study of the related sensors. Related work comes under the heading of section III. This section includes related applications which work for community services. Section IV is devoted to describe the proposed system covering its modules with technology used for each module. Finally, section V concludes the paper.

2 Background

Ultrasonic sensor, Arduino, IR sensors are used to develop impaired glove, where ultrasonic sensor measures the distance and sends out a high frequency pulse when there is any barrier/ obstacles. The sensor has 2 openings on its obverse. One initial transmits ultrasonic waves, (comparable to a little speaker), alternate receives them, (similar to a small amp) [2]. Furthermore, IR sensors in glove that sense and detect the shapes. It is utilized to get a handle on the edges of the item. An IR sensor can detect the daintiness or dimness of a surface with material criticism from a vibration engine. In addition, Arduino is used to embed coding of the hardware like IR sensor, ultrasonic sensor [3]. Arduino is a single board microcontroller intended to make the application more accessible which are interactive items and its environment. The hardware highlights with an open-source hardware board outlined that enables the user/client to append different extension boards. Keeping in mind the end goal to begin, they are essentially associated with a PC and USB cable or with an AC-to-DC connector or battery [4].

3 Related Work

Several related work for blind people has been done to resolve their mobility, navigation and other related issues.

A Applying QR code and portable phones for blinds

Applying QR code and portable phones for blinds, worked on a barcode based framework to help the blind and impaired person to recognize objects in nature is introduced. The framework depends on using QR codes (two-dimensional barcode) appended to a protest and examined utilizing a camera telephone outfitted with QR reader software. The software decodes the barcode to a URL and mentors the mobile's program to get a sound record from the Web that holds an oral description of the object. Our proposed outline is required to be helpful progressively communication with several situations.

B Usable gestures for blind

Regardless of developing awareness of the accessibility issues surrounding touch screen use a by impaired people, planners still face challenges while making open touch screen interfaces. One noteworthy stumbling block is lack of understanding about how blind people really utilize touch screens. We led two client contemplates that looked at how blind and sighted people utilize touch screen motions. To start with, we directed a signal elicitation think about in which 10 visually impaired and 10 located individuals imagined motions to perform basic processing assignments on a tablet PC. We found that visually impaired individuals have different gesture preferences than located individuals, including inclinations for edge-based gestures and motions that include tapping virtual keys on a keyboard.

C An Integrated Indoor/Outdoor System

There are many navigation systems for blind yet rare can give dynamic associations and versatility to changes. Nobody of these frameworks work flawlessly both inside and outside. Drishti utilizes an exact position estimation framework, a remote association, a portable PC, and a verbal statement line to manage impaired clients and support them to movement in common place and new conditions freely and securely.

D Tactile display for Blind

Tesla Touch is an innovation that gives material sensation to moving fingers on touch screens. In view of Tesla Touch, we have created applications for the visually impaired to interpret and make 2D tactile information. In this paper, we exhibit these applications, show perceptions from the association, and examine Tesla Touch's potential in supporting correspondence among outwardly impaired people.

4 Proposed System's Description

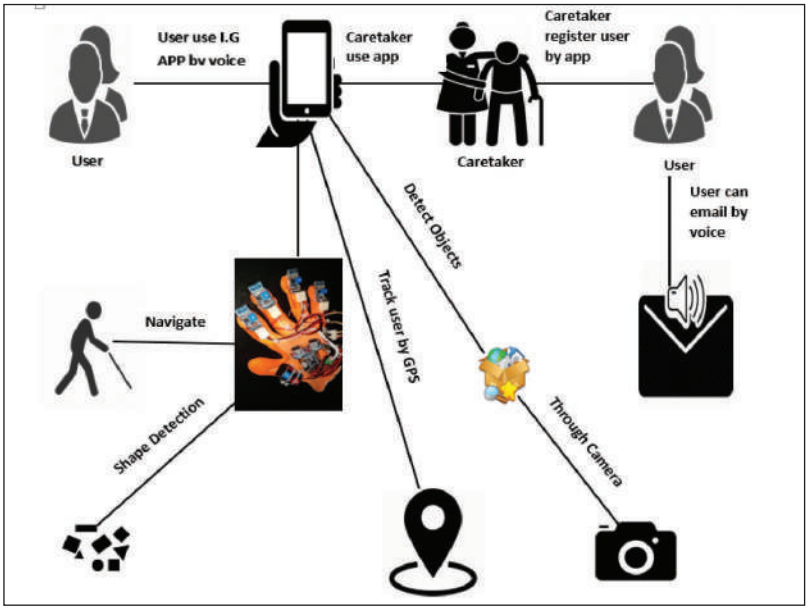

Figure 1: Proposed diagram of impaired glove

The above proposed diagram shows the flow of our project. A portable Glove supports blind people by helping them to navigate their right path. During navigation if any obstacles/barrier is found in front of them they can detect it with the help of ultrasonic sensor which we used in the glove and user will come to know by the frequency of sound. Camera detects the object and conveys message to blind person through voice recognition by an "Impaired Glove" application. With this glove it can recognize different shapes, alphabets and textures of an image. They can also gain their typing ability by taping their fingers in typing mode. In typing mode we embedded the IR sensor and vibrating motor on the finger tip of glove.IR sensor detects the black texture on white paper and vibrating motor vibrates only if we place our finger on black texture.so user can easily detect shapes, textures through vibration .In Impaired glove application there are

two modes caretaker and user mode. In caretaker where caretaker register his/her self and track the blind people. Caretaker may register multiple users. Caretaker will be responsible to track the user's location through GPS tracking system. If user goes anywhere caretaker detects the user's position and gets notified by the application.

A Modules

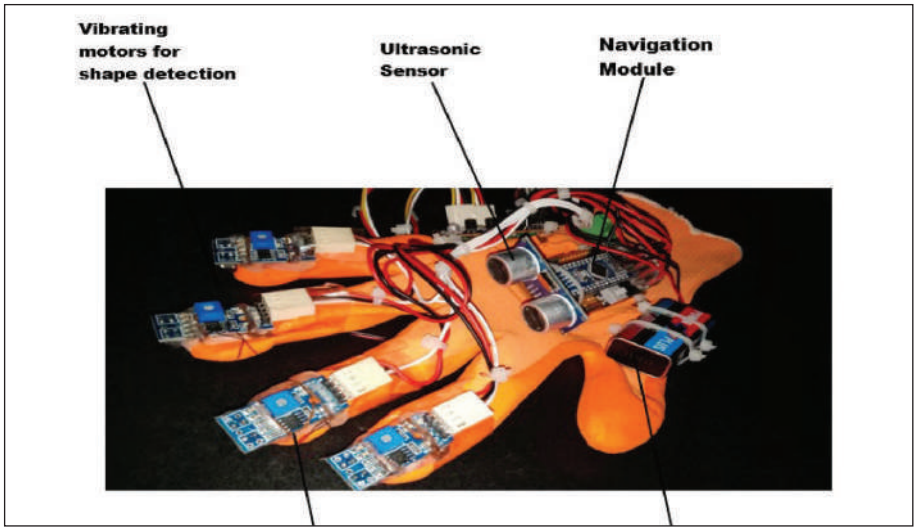

Figure 2: Impaired Glove

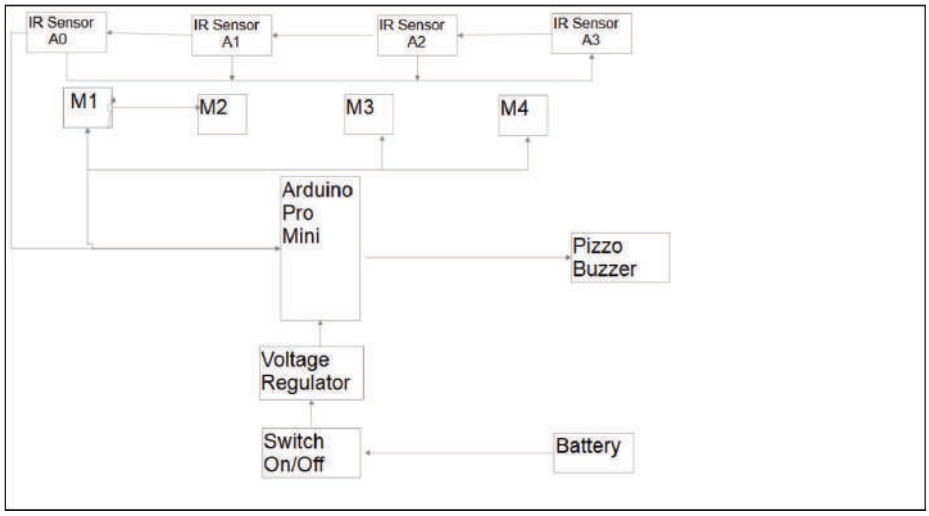

Figure 3: Block Diagram of Glove

1) ***Shape detection:*** In this mode, blind can recognize the edges of the item that is detecting the shapes by using this module for achieving the goals.

2) ***Voice Mail:*** Furthermore, In impaired glove application there is an email mode for blind which person can use by voice or tapping their finger on screen to fill the given fields.

3) ***Navigation:*** This is the main module where blind or impaired person can easily navigate their right path. As they face lots of problems if hurdles or any other obstacles in front of them so that this mode gives the feedback in beep sound.

4) ***Detection:*** In Object detection mode blind can detect the objects by camera through impaired glove's application and gets notified by voice.

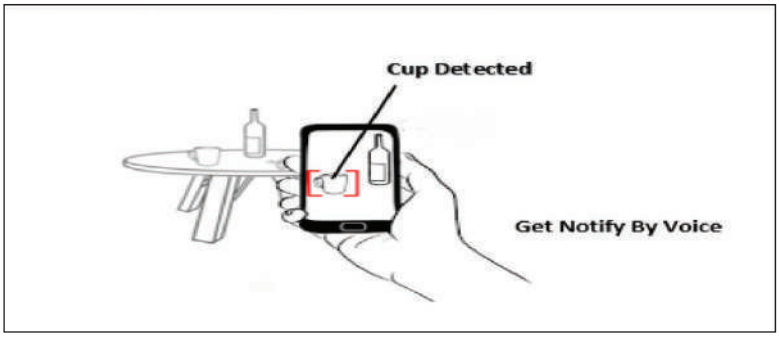

Figure 4: Object Detection

5) ***Tracking:*** In this module, Caretaker will be responsible to track the user's location through GPS tracking system. If the user travels somewhere caretaker can identify the user's position and track them, further caretaker gets notification through the application in case of unusual routes and can track them easily.

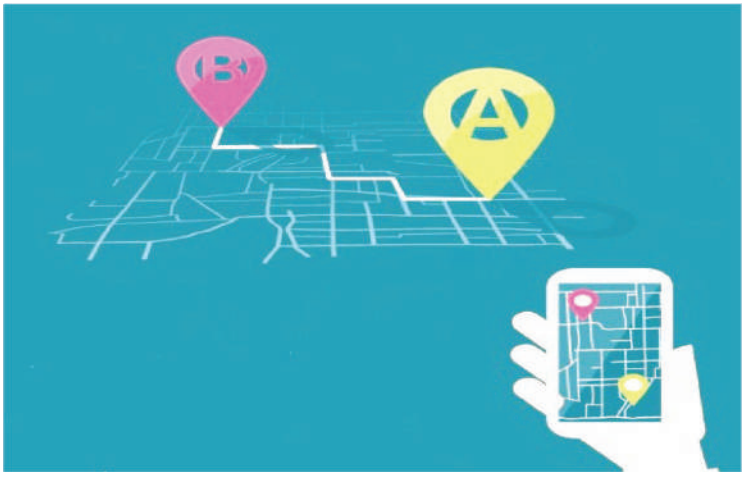

Figure 5: GPS Tracking

6) ***Text Recognition:*** Furthermore text recognition is also one of the features in the application where user can detect the real time text and gets notified by voice through application.

B Technologies used in modules

1) IR sensors

IR sensor technology used in glove which detects the shapes and recognizes the edges.IR sensor detects or senses black texture which is present on white surface.

2) Ultrasonic sensor

It is used in glove to detect the obstacle, measure the distance and give feedback in beep sound.

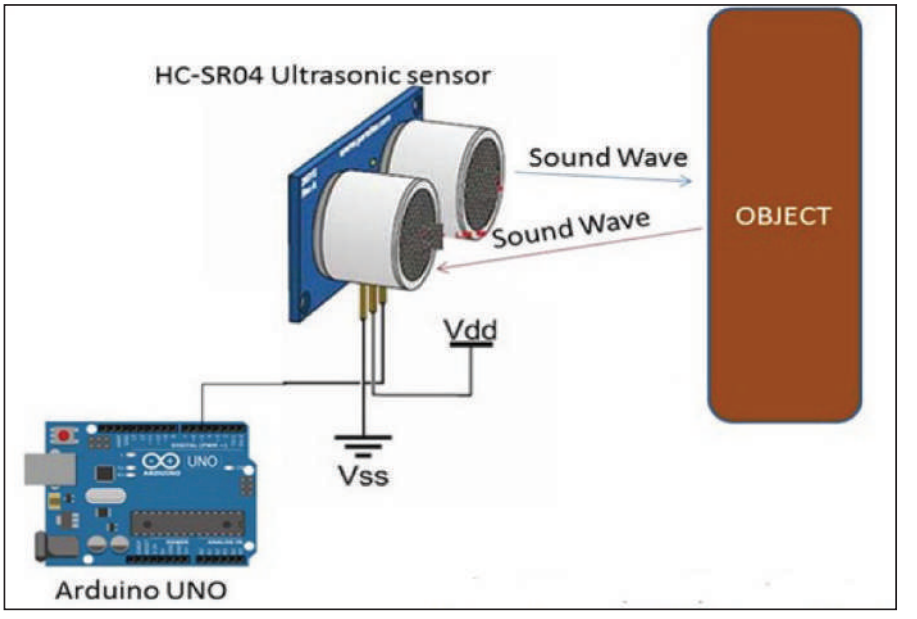

Figure 6: Ultrasonic Sensor working

3) Arduino

This technology was used to embed the coding. It is a single board microcontroller intended to make the application more accessible which are interactive items and its environment.

5 Conclusion

This project's aim is to support blind people who face so many problems due to their disability. A portable glove is designed which focuses to enhance their learning abilities through shape detection and text recognition. A mini camera is provided on the glove for blind to understand what object is in front of them.

Additionally we provide voice email and text recognition features in application. Furthermore, if caretaker wants to know where the blind person is, GPS tracking system facilitates caretaker using impaired glove application through which he can track blind person easily. Hence, this portable glove can be a beneficial application that fulfills a blind's basic needs.

References

[1] Resnikoff, Serge, et al. "Global data on visual impairment in the year 2002." Bulletin of the world health organization 82.11 (2004): 844-851.

[2] Carullo, Alessio, and Marco Parvis. "An ultrasonic sensor for distance measurement in automotive applications." IEEE Sensors journal 1.2 (2001): 143-147.

[3] Morimoto, Carlos Hitoshi, et al. "Pupil detection and tracking using multiple light sources." Image and vision computing 18.4 (2000): 331-335.

[4] Ulrich, Iwan, and Johann Borenstein. "The GuideCane-applying mobile robot technologies to assist the visually impaired." IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 31.2 (2001): 131-136.

[5] Badamasi, Yusuf Abdullahi. "The working principle of an Arduino." Electronics, computer and computation (icecco), 2014 11th international conference on. IEEE, 2014.

[6] Campbell, A. John, et al. "Randomised controlled trial of prevention of falls in people aged≥ 75 with severe visual impairment: the VIP trial." Bmj 331.7520 (2005): 817.

[7] Al-Khalifa, H. S. (2008, July). Utilizing QR code and mobile phones for blinds and visually impaired people. In International Conference on Computers for Handicapped Persons (pp. 1065-1069). Springer, Berlin, Heidelberg.

[8] Ran, L., Helal, S., & Moore, S. (2004, March). Drishti: an integrated indoor/outdoor blind navigation system and service. In Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on(pp. 23-30). IEEE.

[9] Kane, S. K., Wobbrock, J. O., & Ladner, R. E. (2011, May). Usable gestures for blind people: understanding preference and performance. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 413-422). ACM.

[10] Xu, C., Israr, A., Poupyrev, I., Bau, O., & Harrison, C. (2011, May). Tactile display for the visually impaired using TeslaTouch. In CHI'11 Extended Abstracts on Human Factors in Computing Systems (pp. 317-322). ACM.

Multi-Class Emotion Detection (MCED) using Textual Analysis

Hajira Tabbasum [1] Shah Muhammad Emaduddin [2] Aqsa Awan [3] Rafi Ullah [4]

Abstract

Stress and mood disorder is becoming a routine life illness for any human being and it is necessary to find technology based solutions for finding cure and self-treatment of such disorders. In order to find treatment and remedy, it is important to detect ones' emotion before applying mitigation technique. Emotion plays an important role in social interaction and has strong connection with human body and brain signals. Emotions can be stated in many ways like facial expression and body language, speech and by text. Proposed technique is targeting social media platforms for such purpose. As huge textual information is available on social media platforms such as Facebook, Twitter, YouTube etc. in the form of comments, posts etc. Emotion Detection using text is basically a content – based classification problem, connecting ideas from the areas of Natural Language Processing as well as Machine Learning. In this paper we proposed a novel way to detect emotions using Naïve Bayes algorithm by collecting person's browsing history. To find emotions we used Plutchik's Wheel of Classification to check the where the given emotion lies.

Keyword: Multi-Class Emotion Detection, Content Based Classification, Natural Language Processing, Plutchik's Wheel, Keyword Based Emotion Detection

1 Introduction

Emotion plays a vital role in the human life. Emotions are an important element of human nature [30]. Emotions are the strong feelings describing moods, behavior, and sentiments. It can be recognized through body language, facial expressions, heartbeat, voice, movement and text knowledge [33]. Emotion Detection is the process of identifying the spontaneous feeling as distinguished from reasoning or knowledge that is derived from one's current circumstance. On the other hand, technology has become an essential element of our routine lives; one can easily communicate with each other by various means like calls, text messengers and comments on social media. Textual data collected from these communications can play a vital role in detecting mood and emotions of a person. Online searches, browser history and cookies could also be used for the subject.

Textual Analysis is the detailed approach to examine or gather one's information by using their text. As people have directly interacted with a computer or any technology by means of text because now day's maximum of material is accessible on the web in the form of text. Thus, it is useful to extract the feeling for different determination from text. Data mining is used to expert the system because large amount of data is needed for finding the emotions

[1] *Karachi Institute of Economics & Technology, Karachi | hajiratabbasum96@gmail.com*
[2] *Karachi Institute of Economics & Technology, Karachi | shahmuhammademad@gmail.com*
[3] *Karachi Institute of Economics & Technology, Karachi | aqsadilfaraz@gmail.com*
[4] *Karachi Institute of Economics & Technology, Karachi | rafiafridi783@gmail.com*

in textual data. Emotions can be categorized into two types, either positive or negative emotions.

Robert Plutchik created an emotion wheel consist of 8 basic emotions and eight advanced emotions each consist of two basic emotions [32]. Basic emotions are joy, fear, trust, surprise, sadness, disgust, anger, anticipation and the advanced emotions from these basic emotions are optimism, love, submission, awe, disappointment, remorse, contempt and aggressiveness.

The lot of work has been done in the field of emotion detection by different approaches like reacting, body language, face expressions and so on in different papers. But using text needs more improvement.

In order to find and evaluate the intensity of the emotion, Naive Bayes algorithm is applied towards emotionally positive or negative or neutral. It checks the probability of the emotion in which category that emotion lies.

Rest of the paper is divided as follow; Section II is about related work, Section III is the explanation of Emotion Classification Techniques, Section IV is the Methodology Explanation, Section V is Result of Methodology, Section VI us Concluding the Research Work and Section VI is future potential work.

2 Related Work

In literature, some classifications and approaches are presented for emotion to check where the emotion lies through basic emotions classifications [6, 30]. In this work, we consider Plutchik wheel of emotion categorized emotion in 3 categories:

1. Primary
2. Secondary
3. Tertiary.

In literature [6, 30], 3 text approaches are presented. First is keyword based approach, in which emotions can be detected by the keyword used in the given text. Second is learning based approach, where large amount of dataset is trained to the system. Third one is hybrid based approach, this approach uses both keyword based approach and learning based approach for appropriate results.

Robert Plutchik proposed a Plutchik's wheel of emotion theory having eight basic emotions: joy, trust, fear, surprise, sadness disgust, anger and anticipation. This wheel of emotion was influenced by Plutchik's Ten Postulates. It has also twenty four primary, secondary and tertiary dyads (feelings composed of two emotions).this wheel of emotion can categorized in four categories, they are:

1. Primary dyad
2. Secondary dyad
3. Tertiary dyad
4. Opposite emotions

Authors of [1] used different unsupervised machine learning algorithm to classify emotions. For this purpose, authors created a corpus using comments on social media, particularly users' comments in different YouTube videos to train its system. Since the authors used Unsupervised ML techniques are used to classify new entries, emotion labeling is not required in this phase. Limited emotions are the main disadvantage of their research.

Keyword Analysis (KA) and Keyword Negation Analysis (KNA) is another methodology used for emotion detection [2]. Authors try to solve the problem of detecting the emotion in the case of sentence level and emoticon (emojis). KA and KNA is based on a set of proverbs, emoticon, short form of words, exclamatory word. Authors used different keyword based approach like use of basic emotions classifiers e.g. Ekman, Izard or Plutchik, keyword analysis (KA) and keyword negation analysis (KNA). Authors have used paragraph based emotion detection which is a limitation of their research rather than working on sentence structure.

Authors of [3] discuss the idea to detect emotions from input text as well as for training a custom emotion classifier from scratch, based on manually annotated data. Data collected from stack overflow containing 4800 posts was annotated by 12 raters. Limited emotions and described data set were the main disadvantages of this research.

Ekman text is another classification technique used for used for text classification and sentiment analysis [4]. Probabilistic Machine learning algorithm, Support Vector Machine (SVM) has been used for text classification using Hadoop Map Reduce Framework. Limited described emotions are major disadvantage of this research.

Authors of [5] presented an idea with an experiment, which concerned with detection of emotion class at sentence level using an algorithm which calculates the emotion vector of sentence by emotion vector of word. Then on the basis of emotion vector categorized the sentence into appropriate emotion class. Limited emotions, removed stop words from sentences, described dictionary, if the word were not in the described emotion list it would not check and text detection was only for sentences based text, these are the main disadvantages of this research paper.

Authors of [6] evaluated the quality of the emotion lexicons by jointly modeling emotionality and neutrality of words (blogs, news, headlines, and tweets) generated by the proposed method (unigram mixture model) and by state-of-the-art baseline methods on two emotion detection tasks (word-emotion classification and document-emotion ranking). However, their proposed methodology was not working on multi-words.

Authors of [8] uses Naive Bayes algorithm for the word probability checking. However the method has a high time complexity the emotion dictionary was too limited.

Authors of [9] presented a concept of Machine Learning for Emotion Detection using Support Vector Machine. Python based NLTK library was used for word based search, removal of unnecessary words, tagging of words. Emotions are predefined in tabular form which includes the emotions in the first column and the words related to it included in the second & third

column and the last column includes the emotion label set in which that line or paragraph will be added and check emotion. They made the dataset for training and testing, machine which contain extracted tweets in one column and the authors another tweet in 2nd column and another column which contain predefined emotion .Although the paper have limited emotions and only those words which highly show some emotions were considered.

Emotions could also be defined from the context of used word [10]. Authors use NB algorithm which uses word frequency to compute probabilities and makes the Naive assumption that the probability of occurrence of each word is independent of others in a sentence. They use Hand-crafted features, Ontology Model, Statistical Model, and Latent Dirichlet Model for subjectivity detection syntactic method. Although paper have no emotion class defined to find an emotion and work whether the opinions and emotions are positive or negative.

3 Emotion Classification Techniques

Emotion detection Methods are classified in three different classes.

i. Keyword based.
ii. Learning Based.
iii. Hybrid based.

A Keyword Based Approach

In keyword-based approach emotions can be detect by the keyword used in text. It typically involves steps such as pre-processing with a parser and search based on an emotion dictionary.

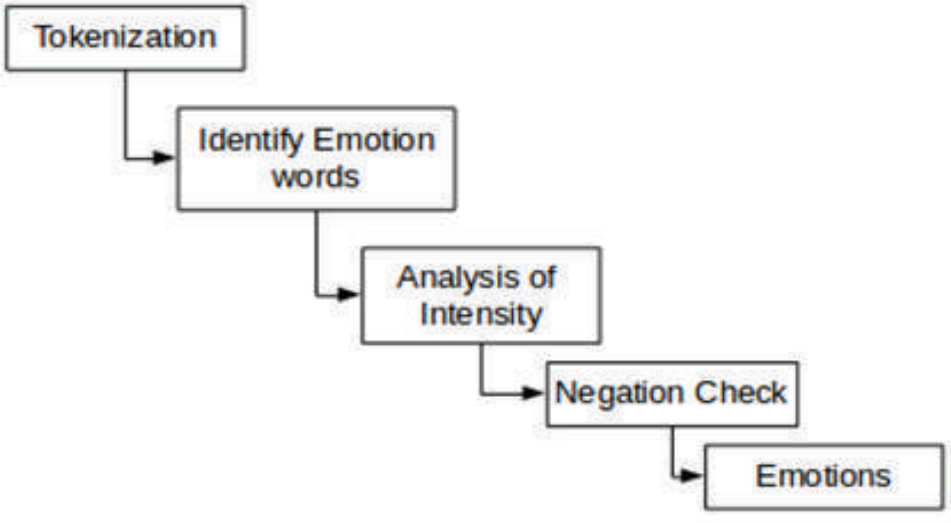

Figure 1: Keyword Spotting Technique

Advantages:

- ✓ Niches and Competitive targeting
- ✓ Search behavior insight
- ✓ Topic cultivation
- ✓ Assumption and bias corrections

Disadvantages:

- × Increased risk of keyword suffering
- × Favoring search engines over users
- × limited ranking data insight
- × Tunnel vision in keyword research

B Learning Based Approach

In machine learning we required large data to train the system. The input text classifies into the suitable emotion class using earlier trained classifier. This is done means of various learning based algorithm such as support vector machine and conditional random field etc. In literature several machine learning algorithms such as Naive Bayes Algorithm, Support Vector Machine and State of the art Neural Networks are used for emotion detection problem. The correctness depends on the training data set. The result of this method is better than keyword-based approach but required a lot of data to train learning algorithm. Advance data pre-processing techniques are required to clean data obtain from sources such as YouTube, Facebook etc. Internet users normally used many languages such for communication and searches. These languages include pure English, roman style languages and other languages. In roman style people normally used different spelled words. Such as given in following comment from YouTube

1. you are reallyyyyyyyy good singer
2. you are gr8 (y)
3. bohat pyara song hain
4. I loveeeeeeee this soooooong :) <3

Sentence 1 contains elongated words which usually people write in case of strong emotions they want to show.

Sentence 2 contains shortened word (usually people used when they sound same, such as great – “eat” in great sounds like “8”, so normally people write “gr8”)

Sentence 3 contains roman Urdu words, where one word usually people write in several form such as “hain”, “hai”, “hay” etc.

Sentence 4 contains emoticon which also shows the strong representation of emotion.

C Hybrid Based Approach

The learning and keyword-based approaches both have some limitations and could not give suitable result. So, to recover the accuracy hybrid-based approach is well. One for hybrid system which uses both learning based approaches to extract semantic and Chinese lexicon ontology to extract quality.

Hybrid Approach is the mixture of Keyword based approach and training datasets. When the system obtains the input and checks that the text has keyword emotion or not. If the input text has one or more emotional keyword then we use keyword approach if word found in the dictionary then we have specific output of the system.

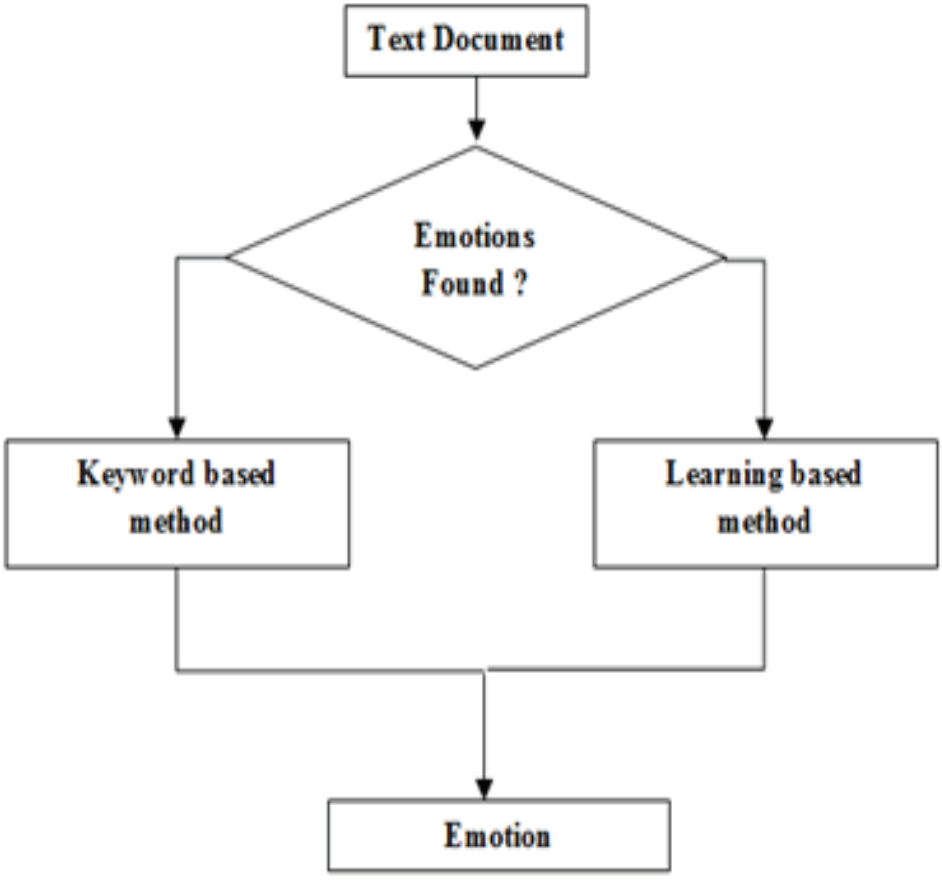

Figure 2: Emotion Class Comparison

Text can be classified form these approaches and from this text depression can be detect by checking probability and intensity level of text.

4 Proposed Methodology

The methodology of this research paper is based on emotion detection by textual analysis with the help of Plutchik's wheel of emotion classification to identify different emotions with the color wheel defined in Plutchik's Wheel. This paper have different classes of emotion compared against different others observed during this research.

Table I: Emotion Class Comparison

Class	P-1 [9]	P-2 [7]	P-3 [2]	P-4 [3,6,7]	P-5 [30]	This Paper
Anger	1	1	1	1	1	1
Disgust	1	1	0	0	1	1
Fear	1	1	0	1	1	1
Joy	1	1	0	1	1	0
Sad	1	1	1	1	1	1
Guilt	0	1	0	0	0	0
Shame	0	1	0	0	0	1
Happy	0	0	1	0	0	1
Love	0	0	0	1	0	0
Surprise	0	0	0	1	1	1
Trust	0	0	0	0	1	1
Anticipation	0	0	0	0	1	1
Advise	0	0	1	0	0	1
Hurt	0	0	1	0	0	0
Confuse	0	0	1	0	0	0

Naive Bayes formula is used to check the probability of the text whether the certain emotion lies in that text or not. Proposed methodology is explained in figure 3.

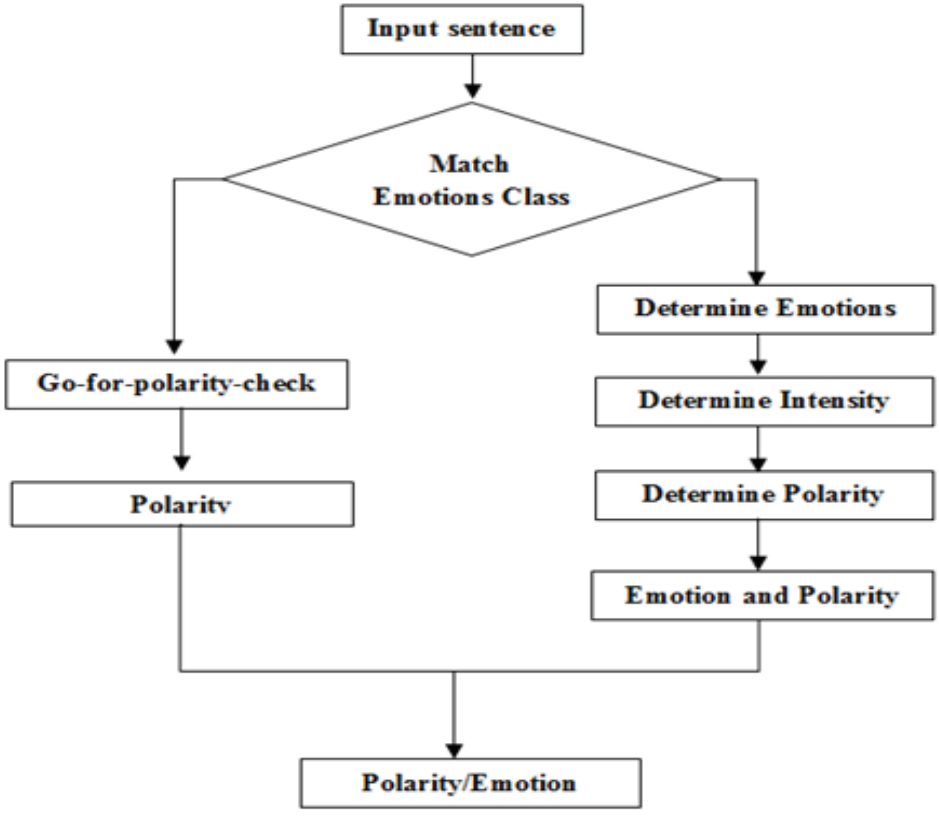

Figure 3: Proposed System Flow Chart

A Extract Text

Extract the text from browser search history of the person. Once a text will extracted it will save in a file then division will takes place. All the basic Natural Language Processing Tasks such as sentence tokenization, word tokenization, stop-word removal, Stemming, Lemmatization and possibly the spell corrections techniques.

B Division

Extract the sentence from the given text. Sentence is basically a set of words that is complete in itself, typically containing a subject and predicate, conveying a statement, question, exclamation, or command, and consisting of a main clause and sometimes one or more subordinate clauses. Sentence will be extracted from text like where there any punctuation mark will detect including dot '.', Exclamation mark '!', question mark '?'. Make a file of sentences.

Advantage: we can break the paragraph into small sentences.

Disadvantage: due to sentence breaker, we will have a problem that if someone put these marks (".", "?", "!") By mistake then our system will break it in sentence and we will not be able to find correct emotion class.

C Breaking

Break a sentence into a word. Remove stop (inappropriate) words which cannot show any type of emotion. Then save these words in a file and then check the intensity of each word which is saved in the file.

Advantage: we can break the sentences into single words, because our system will work on single word.

Disadvantage: may be sometime some words don't lies in that particular text class due to combination of three to four words.

D Check Intensity (for single word)

Check out the intensity of each word to categorize them in different emotions. By using Naive Bayes algorithm check out the probability of each word and then save these words in the file whether the word lies in positive/negative/neutral state. Intensity can be simply calculated as

$$\text{Intensity}={}^{C}/_{R} \quad (1)$$

Where "C" is sum of all intensities of same emotion class and "R" is no of emotional words count.

Advantage: from checking the intensity of the word, we can find the emotion class of that particular word.

Disadvantage: it is time consuming.

E Combination

Combine two consecutive words that are meaningful and are used in a sentence make sense to describe an emotion.

Advantage: due to combine the words, we can tell about the emotion with structure of the given sentence.

Disadvantage: sometime noun and verb combining does not make any sense.

Example: I am, you are, he is, etc.

F Check Intensity (for double word)

Check out the probability of combination of these 2 words and save these words in a new file. Intensity of two words can be calculated by equation 1.

Advantage: from checking the intensity of the word, we can find the emotion class of that particular word.

Disadvantage: it is time consuming.

G Slangs

Check out the word if word is present in slangs or not. If it is presented then take a word from word file or words from a double word file. Search it in a slang words file. Slang words file store slangs according to the location. If slang word is found, check out the meaning of the sentence. Then find the emotion of that meaning. Same as describe for word in a sentence.

Advantage: some words have different meaning which they are showing.
Example: street dogs

Disadvantage: data size will be increase, because slangs meaning is change by location or region. Different terms are used as slang in different locations.

H Idiomatic Word

Check out the word if word is present in idioms or not. Take a word from word file or words from a double word file. Search it in an idioms word file. Idiomatic words file store idioms and their meanings. If slang found, extract the meaning of the sentence. Then find the emotion of that meaning. Same as describe for words in a sentence. Now check the probability of the word whether the word is in high and low intensity level of positive and negative emotions.

Advantage: some words have different meaning which they are showing.
Example: once in a blue moon

Disadvantage: data size will be increase, because idioms meaning is changed by location or region.

I Categorization

Categorize a, b, c category according to the categorized emotions. High intensity emotions, medium intensity emotions and low intensity emotions. Place these emotions in the concerned emotion category.

J Words / Sentenses Checking

Check if the word/words/sentences present in emotion category is/are positive/negative/neutral. Positive/negative/neutral emotions will check with intensity by using Naive Bayes formula.

Advantage: it describe that emotion lies in which category.

Disadvantage: data gathering and time management and data set will be crucial.

5 Results

Results were generated by comparing our own algorithm proposed in Section IV against state of the art technique found in literature review.

Table 2: Selected Sentences Results

Sentence	Polarity	Emotion Class	Emotion Intensity
I am unhappy today	Negative	Sad	60%
He get good marks in his exam	Positive	Happy, Surprise	50% 30%
I will see you	Neutral	–	–
Nothing is impossible	Neutral	–	–
I am getting mad due to happiness	Positive (50%) Negative (50%)	Happy, Angry, Disgust, Surprise	25%, 15%, 15%, 25%
This must be so hard for you	Neutral	–	–
Jump for Joy	Positive	Happy	100%

Any emotion classified as positive or negative can be further classified in to sub-classes (Happy, Disgust, Surprise etc.). In table 2, example sentence "I am getting mad due to happiness", after applying pre-processing steps the only dominant terms that are present in emotion keyword dictionary are "mad" and "happiness", individually "mad" is indicator for negative and "happiness" is for positive. So as per keywords there, the example sentence can be 50% positive and 50% negative. But further classification in to sub-classes will be as follow, 25% happy, 15% Angry, 15% Disgust and 25% surprise as per weight-age and keywords in emotion dictionary and given methodology. For Neutral class, there will be no emotion sub-classes as shown in table 2.

Word	Anger	Anticipation	Disgust	Fear	Joy	Sadness	Surprise	Trust	Negativity	Positivity
abnormal	0	0	1	0	0	0	0	0	1	0
provoking	1	0	1	0	0	0	0	0	1	0
reassure	0	0	0	0	0	0	0	1	0	1
punch	1	0	0	1	0	1	1	0	1	0
muck	0	0	1	0	0	0	0	0	1	0
revolution	1	1	0	1	0	1	1	0	1	1

Figure 4: Word Emotion Lexicon in [7]

In [7] difference emotions classes are categorized on the basis of keywords. For example Anger can be detected when terms like provoking, punch and revolution. There is no words given in the first column that describes joy. Similarity words like abnormal, provoking, punch, muck, revolution and unclean shows negativity. And same as case for other emotion classes such as Anticipation, Disgust, Fear, Sadness, Surprise, Trust and Positivity.

Table 3: Selected Sentences Results

S. No	Text	Polarity	Emotion Class	Emotion Intensity
1	Abnormal	Negative	Disgust	60%
2	Provoking	Negative	Anger Disgust	50% 50%
3	Reassure	Positive	Trust	100%
4	Punch	Negative	Sadness, surprise, anger fear	25%, 15%, 25%, 15%,
5	Muck	Negative	Disgust	100%
6	Revolution	Positive 50% Negative 50%	Sadness, Anticipation, disgust, Anger, Surprise, Fear.	16.66%, 3.33%, 3.33%, 16.66%, 16.66%, 16.66%
7	Unclean	Negative	Ashamed, Disgust, Angry.	6.66%, 33.33%, 6.66%

Words emotions lexicons in [7] are tested against proposed methodology for emotion classes and their intensities as per proposed methodology. Word "punch" is classified as Negative and Sadness with 25%, Surprise with 15%, Anger with 25% and Fear with 15%, which shows the minute details of word punch for emotion classification and this classification become more realistic and clear in real world.

Table 4 represents an example sentence from [7]. This sentence is tokenized as follows.

Table 4: Example from [7] Example of tweets

give	a	Listen	and	a	like	these	guys	are	awesome

After tokenization, stop words has been removed from the sentence as shown in Table 5.

Table 5: After Removing stop-words [7]

give		Listen	and		like				awesome

Table 6 represents the structure of the sentence after tokens have been replaced by emotion class labels as per corpus provided in [7]

Table 6: after replacing with CORPUS [7]

Neutral		Neutral	Neutral		Joy				Neutral

Table 7 represents the same example of table 3 after removing stops word from our customized algorithm.

Table 7: After Removing stop-words

		Listen			like		guys		awesome

Table 4, 5, 6 and 7 shows successive pre-processing steps that will be applied on a sentence. Those pre-processing steps are stop word removal, word replacement and so on.

Table 8: Checking Polarity and Intensity of Example Sentence

Emotion	Intensity	Polarity
Happy	40%	Positive
Surprise	20%	

Polarity of tokens are identified in table 8. Polarity of given sentence is detected as positive, because keywords lies in positive polarity and The intensities of sub-classes "Happy" and "Surprise" are calculated as 40% and 20% respectively These values of intensities are calculated using equation 1.

Table 9: Emotion Category Result of Example Sentence

S no.	Words	Emotion	Intensity	Polarity
1	Give	-	-	Neutral
2	Listen	-	-	Neutral
3	Like	Happy	19.99%	Positive
4	Guys	-	-	Neutral
5	Awesome	Happy	50%	Positive
		Surprise	30%	

After applying pre-processing steps, the remaining sentence as shown in Table 9. **"Give"** is not classified as any emotion class, so polarity is calculated as Neutral. Same as case for **"listen"** and **"guys"**. Like and Awesome with emotion sub-classes Happy (19.99%) and Happy (50%) & Surprise (30%). So overall polarity identified as Positive.

6 Conclusion

Emotions are the most important feature of any human being. There are many methods to find the emotions of a person, however language and communication is one of the basic attribute through which emotion can be determined. Emotions play an important role in finding mood disorder. Some of the emotions cannot be easily recognizable because these emotions are dyads and these dyads can be check through Plutchik's wheel of emotion. Emotions cannot be identified only by handling the polarity of words or sentences but we can also use "bi-word" and "tri-word" combination in order to identify emotions correctly. Context of the word and intensity is also important.

7 Future Work

The potential future work on mentioned problem is very vast and might be need in such digital word. First of all advance natural language pre-processing will be applied to add elongated words and emojis score as well to calculate the emotion class and secondly depression detection will be targeted using similar techniques as discussed for emotion detection using textual analysis as huge amount of data is available on social sites. It is very important to find depression in one's textual information and then do some mitigation accordingly. Mitigation techniques may vary from person to person and from intensity to intensity of depression. Depression is becoming a daily base illness for any human being and it is necessary to find its easy cure with the help of technology. And mixture of straight forward keyword-based technique and machine learning techniques (especially Artificial Neural Network) will be combined along with technique used in this paper to achieve high performance in depression detection and mitigation domain.

References

[1] Hajar, Mousannif. "Using youtube comments for text-based emotion recognition." Procedia Computer Science 83 (2016): 292-299.

[2] Rahman, Romana, Tajul Islam, and Md Humayan Ahmed. "Detecting Emotion from Text and Emoticon." (2017).

[3] Calefato, Fabio, Filippo Lanubile, and Nicole Novielli. "EmoTxt: a toolkit for emotion recognition from text." arXiv preprint arXiv:1708.03892 (2017).

[4] Sruthy, V. V., Amrutha Saju, and AG Hari Narayanan. "Predictive Methodology for Child Behavior from Children Stories."

[5] Tiwari, Sudhanshu Prakash, M. Vijaya Raju, Gurbakash Phonsa, and Deepak Kumar Deepu. "A novel approach for detecting emotion in text." Indian Journal of Science and Technology 9, no. 29 (2016).

[6] Bandhakavi, Anil, Nirmalie Wiratunga, Stewart Massie, and Deepak Padmanabhan. "Lexicon generation for emotion detection from text." IEEE intelligent systems 32, no. 1 (2017): 102-108.

[7] Bandhakavi, Anil. "Domain-specific lexicon generation for emotion detection from text." (2018).

[8] Razak, Zura Izlita, and Sofianita Mutalib. "Web Mining In Classifying Youth Emotions." Malaysian Journal of Computing 3.1 (2018): 1-11.

[9] Ramalingam, V. V., A. Pandian, Abhijeet Jaiswal, and Nikhar Bhatia. "Emotion detection from text." In Journal of Physics: Conference Series, vol. 1000, no. 1, p. 012027. IOP Publishing, 2018.

[10] Chaturvedi, Iti, Erik Cambria, Roy E. Welsch, and Francisco Herrera. "Distinguishing between facts and opinions for sentiment analysis: Survey and challenges." Information Fusion 44 (2018): 65-77.

[11] Shivhare, Shiv Naresh, and Saritha Khethawat. "Emotion detection from text." arXiv preprint arXiv:1205.4944 (2012).

[12] Meo, Rosa, and Emilio Sulis. "Processing affect in social media: a comparison of methods to distinguish emotions in tweets." ACM Transactions on Internet Technology (TOIT) 17, no. 1 (2017): 7.

[13] Skirpan, Michael, and Casey Fiesler. "Ad empathy: A design fiction." In Proceedings of the 2018 ACM Conference on Supporting Groupwork, pp. 267-273. ACM, 2018.

[14] Almashraee, Mohammed, Dagmar Monett Díaz, and Adrian Paschke. "Emotion Level Sentiment Analysis: The Affective Opinion Evaluation." In EMSA-RMed@ ESWC. 2016.

[15] Gentile, Vito, Fabrizio Milazzo, Salvatore Sorce, Antonio Gentile, Agnese Augello, and Giovanni Pilato. "Body gestures and spoken sentences: a novel approach for revealing user's emotions." In Semantic Computing (ICSC), 2017 IEEE 11th International Conference on, pp. 69-72. IEEE, 2017.

[16] Serna, Ainhoa, Jon Kepa Gerrikagoitia, and Unai Bernabé. "Discovery and classification of the underlying emotions in the user generated content (UGC)." In Information and communication technologies in tourism 2016, pp. 225-237. Springer, Cham, 2016.

[17] Sánchez-Rada, J. Fernando, and Björn Schuller. "Emotion and Sentiment Analysis." (2016).

[18] Sharma, Sunny, and Vijay Rana. "Web Personalization through Semantic Annotation System." Advances in Computational Sciences and Technology 10, no. 6 (2017): 1683-1690.

[19] Angiani, Giulio, Stefano Cagnoni, Natalia Chuzhikova, Paolo Fornacciari, Monica Mordonini, and Michele Tomaiuolo. "Flat and hierarchical classifiers for detecting emotion in tweets." In Conference of the Italian Association for Artificial Intelligence, pp. 51-64. Springer, Cham, 2016.

[20] Riahi, Nooshin, and Pegah Safari. "Implicit emotion detection from text with information fusion." Journal of Advances in Computer Research 7, no. 2 (2016): 85-99.

[21] Deyu, Z. H. O. U., Xuan Zhang, Yin Zhou, Quan Zhao, and Xin Geng. "Emotion distribution learning from texts." In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 638-647. 2016.

[22] Ofoghi, Bahadorreza, Meghan Mann, and Karin Verspoor. "Towards early discovery of salient health threats: A social media emotion classification technique." In Biocomputing 2016: Proceedings of the Pacific Symposium, pp. 504-515. 2016.

[23] Fathy, Samar, Nahla El-Haggar, and Mohamed H. Haggag. "A hybrid model for emotion detection from text." International Journal of Information Retrieval Research (IJIRR) 7, no. 1 (2017): 32-48.

[24] Zhang, Yuxiang, Jiamei Fu, Dongyu She, Ying Zhang, Senzhang Wang, and Jufeng Yang. "Text Emotion Distribution Learning via Multi-Task Convolutional Neural Network." In IJCAI, pp. 4595-4601. 2018.

[25] Saad, Mastura Md, Nursuriati Jamil, and Raseeda Hamzah. "Evaluation of Support Vector Machine and Decision Tree for Emotion Recognition of Malay Folklores." Bulletin of Electrical Engineering and Informatics 7, no. 3 (2018): 479-486.

[26] Khairani, Zulia. "Store Environmental Atmosphere on Giant Hypermarket Pekanbaru: Do Effect on Consumers Positive Emotion and Impulse?." In IOP Conference Series: Earth and Environmental Science, vol. 175, no. 1, p. 012046. IOP Publishing, 2018.

[27] Garcia-Garcia, Jose Maria, Víctor MR Penichet, María Dolores Lozano, Juan Enrique Garrido, and Effie Lai-Chong Law. "Multimodal Affective Computing to Enhance the User Experience of Educational Software Applications." Mobile Information Systems 2018 (2018).

[28] Kim, Evgeny, and Roman Klinger. "A Survey on Sentiment and Emotion Analysis for Computational Literary Studies." arXiv preprint arXiv:1808.03137 (2018).

[29] Agrawal, Ameeta, Aijun An, and Manos Papagelis. "Learning Emotion-enriched Word Representations." In Proceedings of the 27th International Conference on Computational Linguistics, pp. 950-961. 2018.

[30] Jeđud, Ivica. "Interdisciplinary Approach to Emotion Detection from Text." (2018): 34-72.

[31] Hayat, Sher. "Emotion Detection through Text: Survey."

[32] Tromp, Erik, and Mykola Pechenizkiy. "Rule-based emotion detection on social media: putting tweets on Plutchik's wheel." arXiv preprint arXiv:1412.4682 (2014).

[33] Cooney, Martin, et al. "Pitfalls of Affective Computing: How can the automatic visual communication of emotions lead to harm, and what can be done to mitigate such risks?." The Web Conference 2018 (WWW'18), Lyon, France, April 23-27, 2018. ACM Publications, (2018).

Code Clone Detection: A Systematic Review

Iqra Yaqub[1] Khubaib Amjad Alam[2]

Abstract

Code cloning in software systems has gained significant development in past few years. Cloning is a general mean of reusing software as existing code snippets can be utilized either by copy and paste methods or by minor modifications in the current code in software systems. However, this may lead to produce bugs and maintenance issues. A plethora of various code clone detection tools and techniques have emerged from last few decades. However, there are no comprehensive studies reviewing all the available techniques since 2013. The aim of this Systematic Literature Review (SLR) is to fill this research gap by systematically reviewing all the available research and extending the research on this particular topic. The main objectives of the study are to identify, categorize and synthesize relevant techniques related to this particular topic. After analyzing initial set of 1181 studies gathered from four large databases, 37 studies relevant to defined research questions were identified by following a systematic and unbiased selection procedure according to standard PRISMA guidelines. This selection process is followed by the data extraction, detailed analysis and reporting of findings. The results of this SLR reveals that different tools and techniques have widely been used for code clone detection, but graph-based and metric-based approaches are most prolific approaches. These approaches have also been used as a part of hybrid approaches. Different match detection techniques are also reported. However, to cope with rapidly evolving clones in software systems, the need is to develop more efficient techniques to improve the state of current research. This study concludes with new recommendations for future research.

Keyword: Software clone, Code clone, Duplicated code, Clone detection, Detection techniques, Reuse, Similarity, Clone detection tools

1 Introduction

In software engineering, the word “abstraction” is used frequently. This technique is widely used by developers to manage the complexity of the software by establishing a level of simplicity. Abstractions at all levels of granularity involves implementation. For doing such implementations, we can start coding from scratch or use some existing code by code cloning [1]. Reusing an existing code is a situation that often occurs during software development process. Existing code can be used as it is if it is fulfilling the requirements or it can be used with minor or even major changes that can be performed at different levels. All of these can be achieved by code cloning which could be of any type that all depends on the programmer’s technique and capability of using the code [2]. Code cloning is a common activity during software development in which existing code snippets can be utilized either by copy and paste methods or by doing minor modifications in the current code in software systems. The pasted code itself

[1]*National University of Computer & Emerging Sciences, Islamabad | f179022@nu.edu.pk*
[2]*National University of Computer & Emerging Sciences, Islamabad | khubaib.amjad@nu.edu.pk*

with or without modifications is called clone of the original. In other words, code clones are like different code fragments that produce similar results on same input.

Code cloning has gained significant importance in our research community. During the development of a software, code cloning can be done intentionally using copy and paste methods by programmers. They can also be introduced unintentionally due to lack of technical knowledge in developers. For example, such accidental code clones are produced due to use of certain design patterns, use of certain APIs, etc. Code cloning has some positive as well as negative effects on the development and maintenance of the software. This activity is adopted or used by the programmers as a common practice to increase productivity, reduce advancement costs and enhance product quality [3, 4, 5].

In software maintenance, duplication of code or reusing code by copy and paste methods with or without modifications is considered a well-known code smell. Although, reusing existing code is a standard practice in modern programming paradigms. But, adapting this approach too much has some negative impact on software systems [2]. It has been observed that code clones have some bad effect on maintenance of software as it increases the chances of bug propagation and produces code that is difficult to maintain. Code clones have bad effect on maintainability and reusability of the software. Software code clones also lack software quality. Considering its harmfulness and to improve the quality of the code, it is important to detect code clones in software systems. So, the negative side of code cloning part needs more attention for detecting code clones and removes them for not becoming a hindrance in the process of software development. However, it is very difficult to identify the original code from copied code after development [6].

Detection of code clones in software systems is very important for avoidance of their side effects. In recent years, many code clone detection methods based on different types of clones have been proposed. Various code clone detection techniques are used according to the characteristics and representation of source code [7]. These code clone detection techniques fall under different categories which will be discussed later.

Recently, it has been investigated that different studies used different tools for detection of different types of code clones having different environment. According to a study [8], there are no general results about the harmfulness of code clones in software systems. It was concluded in the study that "not all code clones make software maintenance more difficult". So, it is unsuitable to remove all the code clones for efficiency of program. However, it is significant to reduce the risk of code clones instead of totally removing them which requires more cost and seems impossible [8].

Code clone detection is a wider field and has gained significant importance from research point of view. Previous research includes different types of code clone detection tools and techniques. However, most of the research is carried out regarding software clone detection in general. There are many surveys and comparative studies in this domain but, there exists no comprehensive review, or systematic study from 2013 to present the state-of-the-art research in this domain. This paper reports a Systematic Literature review (SLR) to fill this research gap

by analyzing and reporting the findings of code clone detection tools and techniques from 2013 to 2018. The purpose of this SLR is to identify, summarize and analyze the existing code clone detection tools and techniques.

The contribution of this SLR involves the taxonomies for understanding the structure of code clone detection tools, techniques and different types of datasets used. Moreover, major findings on code clone detection are uncovered by detailed analysis of the identified solutions. All the studies in this SLR are selected to ensure inclusion criteria and are selected through a quality assessment process. This SLR also considers the overall research productivity of this research field. In this study, section 2 consists of a brief description of related work. Section 3 explains the detailed research method including research questions, study selection process, inclusion and exclusion criteria, quality assessment criteria and data extraction process. Section 4 explains the results, discussion and detailed analysis of findings of the selected primary studies. These selected primary studies consider five research questions. Section 5 describes the conclusion of the study.

2 Background Knowledge

Comparative analysis of different code clone detection techniques observes that text-based techniques can detect Type-1 clones only [6], token-based techniques detect Type-1 and Type-2 clones and tree-based approaches detect Type-1, Type-2 and Type-3 clones. According to a review [5], textual and token-based approaches are good for problem detection. Type-1 and Type-2 are easier to detect than Type-3 and Type-4 clones. PDG-based approach is used to identify Type-3 clones [9]. Graph-based approaches are used more in number than tree-based and metric-based approaches [10].

According to a comprehensive and detailed analysis [11] on software clone detection, the research in this field is increasing day by day. Mainly, semantic clone detection and model-based detection are discussed in this extensive study. This existing SLR is about software clones in general and software clone detection in particular. Different types of clones, different clone detection techniques/tools and their evaluation are discussed. The purpose is to identify the importance of software clone detection techniques. This study identifies that reliable detection of similar code is an open area for research. However, this study does not discuss clone detection techniques/tool from 2013. So, this study fills this research gap by reporting a comprehensive and detailed analysis of the current techniques used since 2013. Neural Networks [17], [18], [19] (CNNs) are feed-forward deep neural networks best suited to solve visual imagery learning problems, e.g., image classification and recognition. They are famous because they eliminate the need to exact image features

3 Research Method

It is necessary to ensure that the search results or analysis must contain all the relevant studies. This can be ensured by performing a systematic literature review (SLR) which is done by identification, interpretation, evaluation and detailed analysis of all the available research associated to a particular domain. A SLR must contain a search plan which is quite fair, free of

biasness and must ensure the completeness of analysis. There is no comprehensive analysis or detailed review of available research on code clone detection since 2013. So, this study aims at conducting a comprehensive SLR on code clone detection by following the SLR guidelines of Kitchenham [12]. This strategy of systematically reviewing has a number of steps to be performed in a systematic way. Development of a review protocol, conduction of systematic review, analysis of results, reporting of results and visualization of results including discussion on findings are the steps of systematic review process.

A Research Questions

This study has a primary research question i.e "What is the state-of-art of code clone detection in software systems?" This main research question is divided into five RQ's. This SLR reports and answers only first two research questions due to shortage of space. The answer of all other research questions will be the part of the extended version of this SLR.

RQ1: What techniques/methods have been used to detect code clones in software systems?

RQ2: What is the overall research productivity in this domain?

RQ3: What type of commercial/open source tools have been used for code clone detection and what are their characteristics?

RQ4: What are the basic types of clones and their taxonomies according to different researchers?

RQ5: Which datasets have been widely used for code clone detection?

B Electronic Databases

Four different electronic databases are used in this process which are enlisted in Table I.

Table 1: Electronic Databases

ED1	ACM	http://dl.acm.org/
ED2	IEEE Xplore	http://ieeexplore.ieee.org/
ED3	Science Direct	http://sciencedirect.com/
ED4	Springer Link	http://link.springer.com/

C Search Terms

A search string was defined by combining different search terms to search all the related articles from the above-mentioned electronic databases enlisted in Table I. Following are the research terms for population, intervention, and outcome.

Population: code, source code, instructions, program, software

Intervention: clone, copy, duplicate, replica, image, dummy

Outcome: detection, recognition, identification, findings, exploration

The main search string includes different inter-related concepts e.g. code clone detection, code clone identification, code duplicate detection etc. All these concepts will be used as a combination.

("code" OR "source-code" OR "source code" OR "sourcecode" OR "program" OR "software") AND ("clone*" OR "cloning" OR "duplicat*" OR "copy*" OR "copies") AND ("detect*" OR "recogni*" OR "identif*")

D Study Selection Procedure

This SLR has a specific study selection procedure which follows the standard PRISMA guidelines for systematic review as visualized in Fig. 1. It has mainly three phases after extraction of results from databases and duplicate removal. Fairness and un-biasness ensured in this process when each phase was done by a detailed consensus meeting.

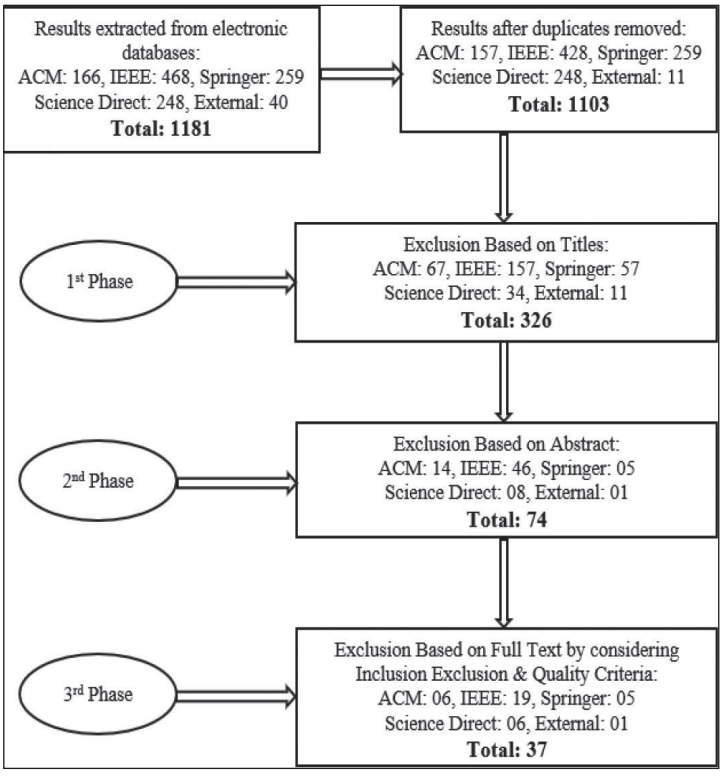

Figure 1: Study Selection Procedure

E Inclusion and Exclusion Criteria

Inclusion and exclusion criteria is defined for selection of relevant studies from databases to

answer the research questions which is listed in Table II.

Table 2: Inclusion and Exclusion Criteria

Inclusion Criteria	
IC1	Any primary study related to code clone detection
IC2	Studies published in 2013-2018
IC3	Only peer reviewed articles should be included
IC4	Only those articles are considered for which full texts are retrievable
Exclusion Criteria	
EC1	Studies other than English language
EC2	Studies with no validation of proposed techniques or comparative evaluation
EC3	Data from editorials, short papers, posters, extended abstracts, blogs or Wikipedia should not be included
EC4	Studies with ambiguous results or findings

These criteria ensured that the studies from 2013 to 2018 were included to fill the research gap of no comprehensive study on code clone detection since 2013. These criteria were applied to all the results in different stages of study selection procedure (Fig. 1). These criteria are mainly applied to 2nd stage for exclusion based on abstracts and 3rd stage for exclusion based on full-text articles considering inclusion & exclusion criteria and quality attributes. Initial set of studies were 1181. Final set of studies after filtration were 37. Fig. 2 depicts the proportion of selected studies.

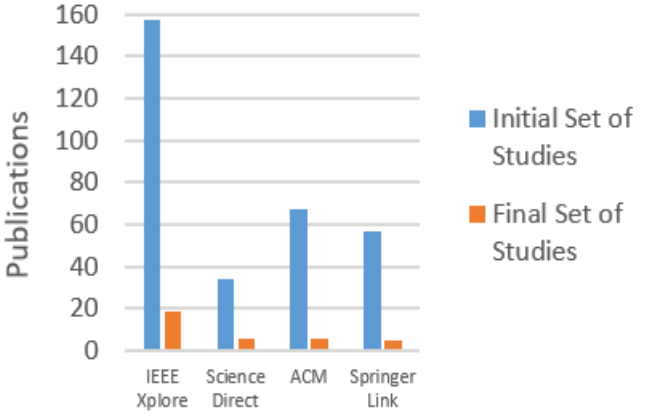

Figure 2: Proportion of selected studies

F Quality Assessment Criteria

Quality assessment was considered for exclusion of full-text articles in third phase of study selection process (Fig. 1). As, 74 articles were filtered out in stage 2. So, quality assessment criteria were considered for these 74 studies which were filtered based on abstracts. Fairness

and un-biasness were accomplished by reviewing each article by 2 reviewers. Scale of three (Fully, Partially, No) was used for conformance to quality ranking. This procedure retrieved 37 studies satisfying the quality attributes. Quality assessment criteria is given in Table III.

Table 3: Quality Assessment Criteria

Quality Assessment Criteria	
QC1	Primary studies must have proper validation
QC2	Primary studies must have clearly defined goals and objectives
QC3	Primary studies must include limitations
QC4	Are the methods used in the studies well defined?

4 Discussion and Results

The similarity of code can be occurred in the form of clone pairs or clone classes. Two code fragments can be similar due to textual similarity or on the basis of resemblance of their functionalities [5]. Textual similarity can be in terms of syntax and functional similarity in terms of functions or semantics of two or more code fragments. Textual similarity is further divided into Type-1, 2 and 3 clones and functional similarity as Type-4 clones [13]. Type-1 are the similar code fragments having some variations in comments, whitespaces and layouts. Type-2 are the similar code fragments having different identifiers, literals, layout and comments. Type-3 clones are the similar code fragments which are further modified by adding or changing statements. Type-4 clones are semantically similar performing same computations [5].

Clone detection process have several steps for finding clone pairs or classes. It has a proper mechanism and requires speedy computational results. Pre-processing, transformation, match detection process, formatting, post processing and aggregation are the phases of a generic work flow of clone detection process [7]. A clone detection technique can focus on one or more of the phases of generic clone detection process [9].

The first step of clone detection process involves pre-processing of code base for elimination of uninterested parts. In this phase, segmentation is performed on source code and then, area of comparison is figured out. Second step of clone detection process is transformation in which preprocessed code is converted into intermediate representations. Extraction, tokenization and parsing are performed to get transformed code [3]. Every transformed fragment is compared to all other fragments to find similarity using comparison algorithm [9]. A set of clones on transformed code are obtained in this phase. The next phase of clone detection involves formatting. In this phase, the code acquired in previous phase is further converted to some new clone pairs or classes related to the original source code. Post processing or filtration is performed in the next phase which can be done manually or by some automated heuristic. Next phase is aggregation which is considered as an optional phase. In this step, clone pairs

extracted from previous phase are aggregated into groups, sets or classes to reduce the amount of data [3]. A generic work flow of clone detection process is visualized in Fig. 3.

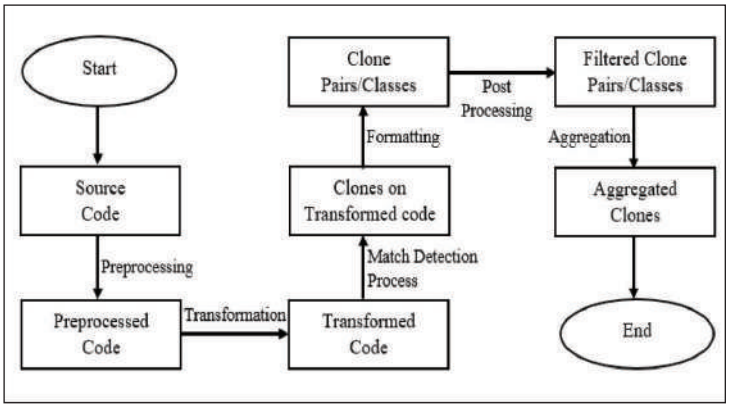

Figure 3: Work Flow of Clone Detection Process

The basic work flow of clone detection process and the idea of clone similarity detection are considered for answering top two previously defined research questions.

RQ1: What techniques/methods have been used to detect code clones in software systems?

This question is considered as one of the main questions of this study as the main thing in a clone detection are the techniques or methods that a clone detection process uses. There are different types of clones that can be detected by different approaches. Different types of techniques or methods for code clone detection include text-based, token-based, abstract syntax tree-based, program dependency graph-based, metric-based and hybrid approaches.

Text-based approach is one of the simple and fastest approach. It is used to detect Type-1 clones. Comparison is performed line by line on two code fragments and, in this way, similarity on the basis of text is detected as clones. Token-based approach converts code fragments into tokens and these tokens are compared by using matching algorithm to find similarity. It can detect both Type-1 and Type-2 clones. Tree-based approach converts source code into Abstract Syntax Tree (AST) and similar trees are identified using tree matching algorithm. Graph-based approach converts code fragments into Program Dependency Graph (PDG) which contains the semantic information of code fragments. Similar subgraphs are identified by using some sub-graph matching algorithm. This approach can detect Type-4 clones. In metric-based approach, different metrices are computed and values of metrices are compared to find similarity. Moreover, Hybrid approach can use these approaches as a combination to give the better results of similarity [3].

Different types of techniques have been used by different researchers depending on the nature of clones. Table IV provides an overview of all the techniques used in selected primary studies since 2013.

Table 4: Overview of Techniques in Selected Studies

Ref.	Technique/Method	Approach
[14]	Feature extraction from BDG, PDG, AST	Framework
[15]	Suffix array, token substrings	Method+Tool
[16]	Token-based approach, Filtering heuristic	Method
[17]	Count matrix clone detection (CMCD), AST	Method
[18]	Token-based approach, deep learning	Method
[19]	Formal methods, CCS transformation	Method+Tool
[20]	Tree-based (AST) + Token-based methods	Method
[21]	Coarse-grained + Fine-grained methods, Hash values + Levenshtein distance	Method+Tool
[22]	Dynamic dependence graphs	Method+Tool
[23]	Token-based + ASTs, computing LCPs	Method+Tool
[24]	Interface information + PDG	Method
[25]	PDG, Plan calculus to represent programs	Method+Tool
[26]	PDG, Approximate Subgraph Matching	Method
[27]	Concolic Analysis, Levenshtein distance	Method
[28]	Static data flow analysis, I/O profiles	Method+Tool
[29]	Metric Collection, Pairwise comparisons	Model
[30]	Method Interface Similarities, Jaccard similarity measure	Method
[31]	Concolic Analysis	Method+Tool
[32]	PDG generation, PDG's merging	Framework
[33]	Textual analysis (Island-drivern parsing approach) + Metrices	Method+Tool
[34]	Smith-Waterman algorithm, Fine-grained	Method+Tool
[35]	Smith-Waterman algorithm	Method+Tool
[36]	PDG, Spatial-based+graph based pattern mining	Framework
[37]	PDG, Method trials	Method
[38]	Hybrid (Metric-based + Token-based)	Model+Tool
[39]	PDG, Slice-based algorithm	Method+Tool
[40]	Token-based approach, Heuristics (prefix filtering + token position filtering + adaptive prefix filtering)	Method
[41]	Token Matching, Jaccard similarity	Model
[42]	AST + PDG, Vector representation	Method+Tool
[43]	Feature extraction, DBSCAN Clustering	Method
[44]	Token-based, Partial Index Creation	Method+Tool
[45]	Metric-based appoach, Distance Matrix	Method
[46]	Metric-based method, Metric comparison	Method

[47]	Partial indexes, Jaccard similarity metric	Method+Tool
[48]	K-means Clustering, Friedman method	Method
[49]	Hybrid (PDG + Metric-based) approach	Method
[1]	AST, Greedy method	Method

The overview of all the techniques in selected primary studies from Table IV shows that after 2013, mostly studies use hybrid approaches using two or more basic detection techniques. It has been observed that numerous studies used Graph-based [14, 22, 24-26, 32, 36, 37, 39, 42, 49] and Metric-based [24, 29, 30, 33, 38, 43, 45-47, 49] approaches to find similarity for code clone detection. Moreover, some studies also used Token-based [15, 16, 18, 20, 23, 38, 40, 41, 44], Hybrid [14, 20, 21, 23, 24, 33, 38, 42, 49] and Tree-based [1, 14, 17, 20, 23, 42] approaches. However, only a few studies included Textual analysis [33]. Some other detection techniques used in some studies are Formal methods [19], K-means clustering [48], Smith-Waterman algorithm [34, 35], Static flow analysis [28], and Concolic analysis [27, 31]. However, many studies used combination of different detection techniques to improve the efficiency of similarity detection of code clones.

RQ2: What is the overall research productivity in this domain?

The purpose of this research question is to identify the overall research productivity in code clone detection. This can be done by analyzing Chronological distribution of selected studies, most influential studies of the domain and potentially relevant publication sources.

Chronological distribution is used for the demonstration of increasing research interest in a particular field. Based on this, further research or future work can be conducted. This study basically fills the research gap of having no comprehensive systematic review since 2013. So, studies from 2013 to 2018 were selected accordingly. This distribution visualizes that the research on clone detection was on peak in 2015 and 2017. Fig. 4 visualizes the Chronological distribution of selected studies.

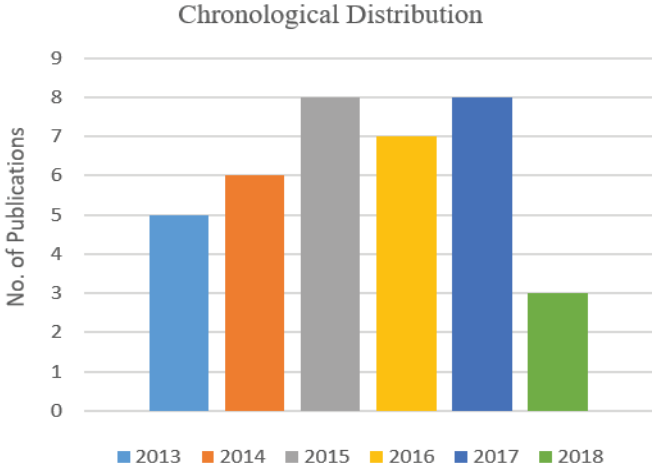

Figure 4: Chronological Distribution of Selected Studies

As, this SLR extracted studies from four main electronic databases which were IEEE Xplore, Springer Link, Science Direct and ACM. Most of the studies which were retrieved after selection procedure were from IEEE Xplore. However, full texts were retrieved from all of the four databases. Most influential studies of code clone detection having greatest count of citations are enlisted in Table V. Table shows top 10 studies in terms of citations having greatest citation count of [44].

Table 5: Most Influential Primary Studies

Ref.	Title	Citation Count
[44]	SourcererCC: Scaling Code Clone Detection to Big-Code	61
[1]	Deep Learning Code Fragments for Code Clone Detection	45
[34]	Gapped Code Clone Detection with Lightweight Source Code Analysis	34
[16]	A parallel and efficient approach to large scale clone detection	19
[36]	Pattern mining of cloned codes in software systems	16
[48]	Threshold-free code clone detection for a large-scale heterogeneous Java repository	14
[31]	CCCD: Concolic Code Clone Detection	13
[25]	Detecting Refactored Clones	13
[41]	SeByte: Scalable clone and similarity search for bytecode	10
[22]	Code Relatives: Detecting Similarly Behaving Software	09

Studies which were extracted from different databases have different publication venues. The overall research productivity can be found out by analysis of distribution of primary studies in these publication venues. Table VI enlists the distribution of primary studies along journals and conferences which clearly shows that the ratio of primary studies along conference proceedings are more than the journal articles. However, table shows that 22nd International conference on SANER has maximum count while the count of journals is same for all journal articles.

Table 6: Distribution of Primary Studies Along Journals and Conferences

Journals	#
Expert Systems with Applications	1
Journal of Software: Evolution and Process	1
Science of Computer Programming	1
Journal of Software Engineering and Applications	1
Computational Science and its applications	1
Journal of Software Engineering Research and Development	1
Information Sciences	1
Programming and Computer Software	1
Journal of Systems and Software	1
Science of Computer Programming	1
Procedia Computer Science	1

Conferences	#
Proceedings of the IEEE International Conference on Software Engineering and Service Sciences	1
Proceedings of the Thirty-Seventh Australasian Computer Science Conference (ACSC 2014), Auckland	1
Proceedings - 2017 IEEE International Conference on Software Maintenance and Evolution	1
Seventh International Conference on Intelligent Computing and Information Systems	1
Foundation of Software Engineering Conference	
Proceedings of the 2017 ACM SSIGPLAN workshop on partial evaluation and program manipulation	1
IEEE 12th International Workshop on Software Clones	1
European Conference on Object-Oriented Programming	1
11th IEEE International Workshop on Software Clones, co-located with SANER	1
Proceedings of the 30th annual ACM symposium on Applied computing	1
IEEE International Conference on Program Comprehension	1
24th Asia-Pacific Software Engineering Conference	1
Proceedings- Working conference on Reverse Engineering, WCRE	1
CSIT 2015 - 10th International Conference on Computer Science and Information Technologies	1
21st International Conference on Program Comprehension (ICPC)	
22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)	1
International Conference on Data and Software Engineering (ICODSE)	1
Confluence 2013: The Next Generation Information Technology Summit (4th International Conference)	1
15th IEEE International Conference on Machine Learning and Applications (ICMLA)	1
International Conference on Neural Information Processing (ICONIP)	1
38th IEEE International Conference on Software Engineering	1
2nd International Conference on Contemporary Computing and Informatics (IC3I)	1
IEEE International Conference on Software Engineering Companion	1
International Conference on Intelligent Computing and Control Systems ICICCS	1
International Conference on Automated Software Engineering	

5 Conclusion & Future Work

Code clone detection has emerged as a most dominant area of research. The detection of clones is necessary for improving the quality and maintenance of software systems. This SLR provides a comprehensive systematic review of all the existing research on code clone detection since 2013. After a detailed analysis, 37 primary studies were selected having different techniques to detect code clones which include text-based, token-based, tree-based, graph-based, metric-based and hybrid approaches. Results of this study reveals that PDGs and metric-based approaches are the mostly commonly used techniques to detect code clones. Although, many efficient hybrid approaches have been developed but still, the need is to improve the techniques in terms of accuracy and efficiency. Overall research productivity in code clone detection is defined by chronological distribution which visualizes the increasing research interest towards code clone detection in past few years. Lastly, this study presents preliminary results relevant to the two selective research questions. The extended version of this study will provide comprehensive discussion related to all defined research questions.

Acknowledgement

This research is supported by FAST-National University of Computer & Emerging Sciences (NUCES), Islamabad, Pakistan.

References

[1] M. White, M. Tufano, C. Vendome, and D. Poshyvanyk, "Deep learning code fragments for code clone detection," Proc. 31st IEEE/ACM Int. Conf. Autom. Softw. Eng. - ASE 2016, pp. 87–98, 2016.

[2] M. Tech, C. S. Engg, and M. Gobindgarh, "Hybrid Approach for Efficient Software Clone Detection," IRACST–Engineering Sci. Technol. An Int. J., vol. 3, no. 2, pp. 2250–3498, 2013.

[3] G. Chatley, S. Kaur, and B. Sohal, "Software clone detection: A review," Int. J. Control Theory Appl., vol. 9, no. 41, pp. 555–563, 2016.

[4] C. K. Roy and R. Koschke, "The Vision of Software Clone Management : Past , Present , and Future (Keynote Paper)," Softw. Maintenance, Reengineering Reverse Eng. (CSMR-WCRE), 2014 Softw. Evol. Week-IEEE Conf. on. IEEE, pp. 18–33, 2014.

[5] P. Prem, "A Review on Code Clone Analysis and Code Clone Detection," Int. J. Eng. Innov. Technol., vol. 2, no. 12, pp. 43–46, 2013.

[6] K. Kaur and R. Maini, "A Comprehensive Review of Code Clone Detection Techniques," vol. IV, no. Xii, pp. 43–47, 2015.

[7] M. Kapdan, M. Aktas, and M. Yigit, "On the Structural Code Clone Detection Problem: A Survey and Software Metric Based Approach," Iccsa 2014, vol. 8583 LNCS, no. PART 5, pp. 492–507, 2014.

[8] H. Murakami, K. Hotta, Y. Higo, H. Igaki, and S. Kusumoto, "Gapped code clone detection with lightweight source code analysis," IEEE Int. Conf. Progr. Compr., pp. 93–102, 2013.

[9] A. Sheneamer and J. Kalita, "A Survey of Software Clone Detection Techniques," Int. J. Comput. Appl., vol. 137, no. 10, pp. 1–21, 2016.

[10] K. Solanki and S. Kumari, "Comparative study of software clone detection techniques," 2016 Manag. Innov. Technol. Int. Conf., no. 1995, p. MIT-152-MIT-156, 2016.

[11] D. Rattan, R. Bhatia, and M. Singh, "Software clone detection: A systematic review," Inf. Softw. Technol., vol. 55, no. 7, pp. 1165-1199, 2013.

[12] S. E. Group, "Guidelines for performing Systematic Literature Reviews in Software Engineering," 2007.

[13] A. Gupta and B. Suri, "A Survey on Code Clone, Its Behavior and Applications," Netw. Commun. Data Knowl. Eng., vol. 27–39, 2018.

[14] A. Sheneamer, S. Roy, and J. Kalita, "A detection framework for semantic code clones and obfuscated code," Expert Syst. Appl., vol. 97, pp. 405–420, 2018.

[15] Q. Q. Shi, L. P. Zhang, F. J. Meng, and D. S. Liu, "A novel detection approach for statement clones," Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS, pp. 27–30, 2013.

[16] M. Zanoni, F. Perin, F. A. Fontana, and G. Viscusi, "A parallel and efficient approach to large scale clone detection," J. Softw. Evol. Process, vol. 26, no. 12, pp. 1172–1192, 2014.

[17] X. Chen, A. Y. Wang, and E. Tempero, "A replication and reproduction of code clone detection studies," Conf. Res. Pract. Inf. Technol. Ser., vol. 147, no. ACSC, pp. 105–114, 2014.

[18] L. Li, H. Feng, W. Zhuang, N. Meng, and B. Ryder, "CCLearner: A deep learning-based clone detection approach," Proc. - 2017 IEEE Int. Conf. Softw. Maint. Evol. ICSME 2017, pp. 249–260, 2017.

[19] A. Cuomo, A. Santone, and U. Villano, "CD-Form: A clone detector based on formal methods," Sci. Comput. Program., vol. 95, pp. 390–405, 2014.

[20] R. Ami and H. Haga, "Code Clone Detection Method Based on the Combination of Tree-Based and Token-Based Methods," J. Softw. Eng. Appl., vol. 10, no. 13, pp. 891–906, 2017.

[21] A. Sheneamer and J. Kalita, "Code clone detection using coarse and fine-grained hybrid approaches," Intell. Comput. Inf. Syst. (ICICIS), 2015 IEEE Seventh Int. Conf. on. IEEE, pp. 472–480, 2015.

[22] F.-H. Su, J. Bell, K. Harvey, S. Sethumadhavan, G. Kaiser, and T. Jebara, "Code relatives: detecting similarly behaving software," Proc. 2016 24th ACM SIGSOFT Int. Symp. Found. Softw. Eng. - FSE 2016, pp. 702–714, 2016.

[23] T. Matsushita, "Detecting Code Clones with Gaps by Function Applications," Proc. 2017 ACM SIGPLAN Work. Partial Eval. Progr. Manip., pp. 12–22, 2017.

[24] R. Tajima, "Detecting Functionally Similar Code within the Same Project," Softw. Clones (IWSC), 2018 IEEE 12th Int. Work. IEEE, pp. 51–57, 2018.

[25] M. Shomrat and Y. A. Feldman, "Detecting refactored clones," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7920 LNCS, pp. 502–526, 2013.

[26] C. M. Kamalpriya and P. Singh, "Enhancing program dependency graph based clone detection using approximate subgraph matching," IWSC 2017 - 11th IEEE Int. Work. Softw. Clones, co-located with SANER 2017, pp. 61–67, 2017.

[27] D. E. Krutz, S. A. Malachowsky, and E. Shihab, "Examining the effectiveness of using concolic analysis to detect code clones," Proc. 30th Annu. ACM Symp. Appl. Comput. - SAC '15, pp. 1610–1615, 2015.

[28] F. H. Su, J. Bell, G. Kaiser, and S. Sethumadhavan, "Identifying functionally similar code in complex codebases," IEEE Int. Conf. Progr. Compr., vol. 2016–July, pp. 1–10, 2016.

[29] M. S. Aktas and M. Kapdan, "Implementation of Analytical Hierarchy Process in Detecting Structural Code Clones," Int. Conf. Comput. Sci. Its Appl., vol. 2, pp. 652–664, 2017.

[30] R. H. Misu and K. Sakib, "Interface Driven Code Clone Detection," Asia-Pacific Softw. Eng. Conf. (APSEC), 2017 24th. IEEE, 2017.

[31] D. E. Krutz and E. Shihab, "CCCD: Concolic code clone detection," Proc. - Work. Conf. Reverse Eng. WCRE, pp. 489–490, 2013.

[32] A. Avetisyan, S. Kurmangaleev, S. Sargsyan, M. Arutunian, and A. Belevantsev, "LLVM-based code clone detection framework," CSIT 2015 - 10th Int. Conf. Comput. Sci. Inf. Technol., pp. 100–104, 2015.

[33] E. Kodhai and S. Kanmani, "Method-level code clone detection through LWH (Light Weight Hybrid) approach," J. Softw. Eng. Res. Dev., vol. 2, no. 1, p. 12, 2014.

[34] H. Murakami, K. Hotta, Y. Higo, H. Igaki, and S. Kusumoto, "Gapped code clone detection with lightweight source code analysis," IEEE Int. Conf. Progr. Compr., pp. 93–102, 2013.

[35] H. Murakami, Y. Higo, and S. Kusumoto, "ClonePacker: A tool for clone set visualization," 2015 IEEE 22nd Int. Conf. Softw. Anal. Evol. Reengineering, SANER 2015 - Proc., pp. 474–478, 2015.

[36] W. Qu, Y. Jia, and M. Jiang, "Pattern mining of cloned codes in software systems," Inf. Sci. (Ny)., vol. 259, pp. 544–554, 2014.

[37] B. Priyambadha and S. Rochimah, "Case study on semantic clone detection based on code behavior," 2014 Int. Conf. Data Softw. Eng., pp. 1–6, 2014.

[38] K. Raheja and R. K. Tekchandani, "An efficient code clone detection model on Java byte code using hybrid approach," 4th Int. Conf. Next Gener. Inf. Technol. Summit, Conflu. 2013, vol. 2013, no. 647 CP, pp. 16–21, 2013.

[39] S. Sargsyan, S. Kurmangaleev, A. Belevantsev, and A. Avetisyan, "Scalable and accurate detection of code clones," Program. Comput. Softw., vol. 42, no. 1, pp. 27–33, 2016.

[40] M. A. Nishi and K. Damevski, "Scalable code clone detection and search based on adaptive prefix filtering," J. Syst. Softw., vol. 137, pp. 130–142, 2018.

[41] I. Keivanloo, C. K. Roy, and J. Rilling, "SeByte: Scalable clone and similarity search for bytecode," Sci. Comput. Program., vol. 95, pp. 426–444, 2014.

[42] A. Sheneamer and J. Kalita, "Semantic Clone Detection Using Machine Learning," 2016 15th IEEE Int. Conf. Mach. Learn. Appl., pp. 1024–1028, 2016.

[43] M. Chawla and K. P. Miyapuram, "Software Clone Detection Using Clustering Approach," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9490, pp. 467–474, 2015.

[44] H. Sajnani, V. Saini, J. Svajlenko, C. K. Roy, and C. V. Lopes, "SourcererCC: Scaling Code Clone Detection to Big Code," Softw. Eng. (ICSE), 2016 IEEE/ACM 38th Int. Conf. on. IEEE, no. 1, pp. 1157–1168, 2016.

[45] M. Sudhamani and L. Rangarajan, "Structural similarity detection using structure of control statements," Procedia Comput. Sci., vol. 46, no. Icict 2014, pp. 892–899, 2015.

[46] M. Sudhamani, "Code clone detection based on order and content of control statements," Contemp. Comput. Informatics (IC3I), 2016 2nd Int. Conf. on. IEEE, pp. 59–64, 2016.

[47] J. Svajlenko and C. K. Roy, "Fast and flexible large-scale clone detection with cloneworks," Proc. - 2017 IEEE/ACM 39th Int. Conf. Softw. Eng. Companion, ICSE-C 2017, pp. 27–30, 2017.

[48] I. Keivanloo, F. Zhang, and Y. Zou, "Threshold-free code clone detection for a large-scale heterogeneous Java repository," 2015 IEEE 22nd Int. Conf. Softw. Anal. Evol. Reengineering, SANER 2015 - Proc., pp. 201–210, 2015.

[49] G. Singh, "To Enhance the Code Clone Detection Algorithm by using Hybrid Approach for detection of code clones," Intell. Comput. Control Syst. (ICICCS), 2017 Int. Conf. on. IEEE, pp. 192–198, 2017.

Supervised Learning Algorithm of Classification on Basis of Ranges

Ahmer Hasan[1] Usman Khan[2]

Abstract

A supervised learning algorithm of classification which is implemented on linear data. Ranges/ Boundaries of each group are calculated. As the data is linear the number of ranges will be equal to a number of classes. New input data will be tested by in which range/boundary it lies. And the range in which it lies, the class of that specific range will be assigned to the new data. The experimental results show that the accuracy depends on the linearity of the data. Therefore our model can show accuracy up to 99.99% also if the data is completely linear in nature.

Keyword: Classification,Linear Data,Supervised Learning,Unsupervised Learning, Ranges, Algorithms, Clustering, Decision Tree

1 Introduction

Machine learning [15] enriches us with many of its learning techniques, some of which includes the technique of classification [1]. Supervised learning [2] [17] includes data which is classified or you can say that each data is labeled with a class instance. Unsupervised learning [2] [17] includes data which is not classified or unlabeled data. Data which is classified is trained for predicting the class of the new incoming inputs. And the data which is not classified is grouped/ clustered [5] [17] in unsupervised learning.

There are many supervised and unsupervised learning classification algorithms which include K-Nearest Neighbor [3] for supervised learning and K-Means [4] [17] for unsupervised learning.

K-Nearest Neighbor is an example of a supervised learning algorithm for classification. The new features checks 'k' number of data points (neighbors) closest to them by calculating their distances (distances are usually calculated by Manhattan [6] or Euclidean [6] methods). The most repeated classes among the closest 'k' number of neighbors are the predicted class for the new input.

K-Means is an example of an unsupervised learning algorithm for clustering. The number of 'k' decides the number of centroids/means/clusters of the data. The 'k' no of means are selected from the data randomly. Each mean point or centroid has a distance with each other data point. The data point which is most near among all the centroids is thenconsidered to be the part of the cluster or group of the respective centroid/mean point. New means are calculated again but this time it is among the groups, i.e each group will a have mean (a new centroid). Now the distance is calculated again of each data point with the new centroids, and each data point is assigned to the nearest centroid. This process continues until any data point does not change the group.

[1]*Karachi Institute of Economics & Technology, Karachi |ahmerhasan123@yahoo.com*
[2]*Karachi Institute of Economics & Technology, Karachi |usman@pafkiet.edu.pk*

The structure of the paper is as follows. In section 2, literature review; in section 3, explanation of the data and feature selection; in section 4, we propose our classification algorithm; section 5, clustering of an unsupervised data using K-Means; section 6, practical implementation and the results; Finally, the conclusion of the paper is shown in section 7.

2 Literature Review

A Machine Learning

It is one of the types of Artificial Intelligence [8], which works on the development and designing of algorithms which uses past data that provide computers ability to predict or make decisions by the use of statistical/mathematical methods [9].

The historical data is used for the training part of the algorithms, which is actually making the computer able to learn from data. The new inputs are tested on behalf of the empirical data, these new inputs are called testing data.

The measures of accuracy [10] of any learning algorithm depend on many factors which include features extraction [11], features normalization [12], selecting a suitable algorithm according to the nature of the data [11], etc.

Following are four types of learning algorithms which depend on the nature of the data:

1) Supervised Learning: It consists of labeled data. Predictions are made by regression models [17] and classifications are made by classifiers.

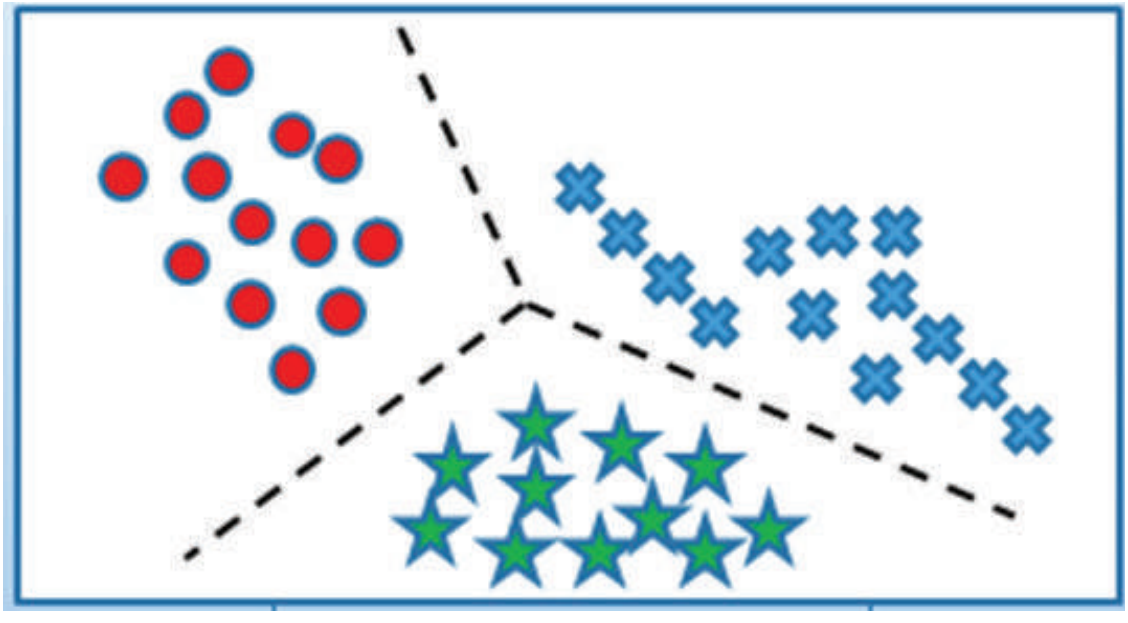

Figure 1: Graphical View of Supervised Learning Data

2) ***Unsupervised Learning:*** It consists of unlabeled data. Clustering, probability distribution estimation, finding an association (in features) and dimension reductions are used in unsupervised learning.

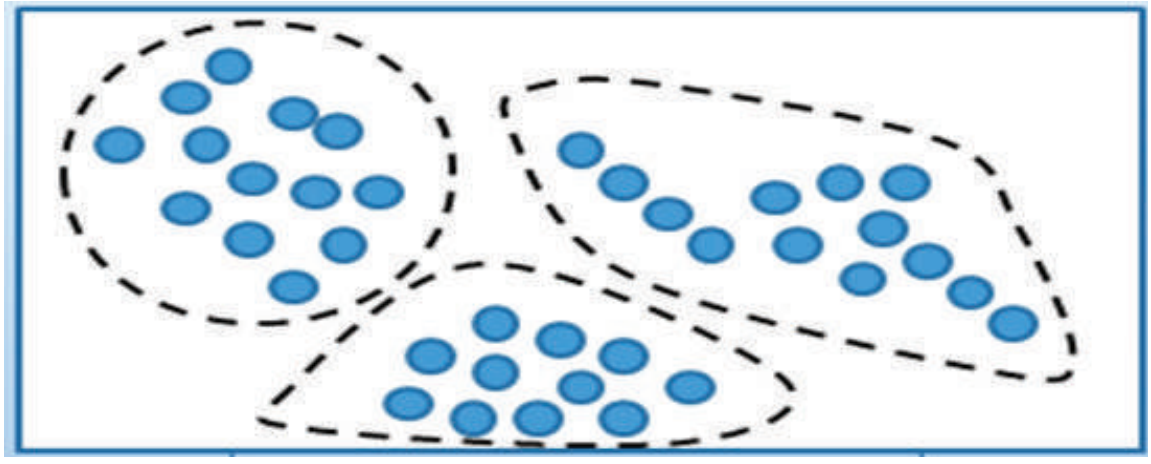

Figure 2: Graphical View of Unsupervised Learning Data

3) ***Semi-Supervised Learning:*** The data is partially labeled in nature. Semi-supervised learning [13] is a mixture of supervised and unsupervised learning.

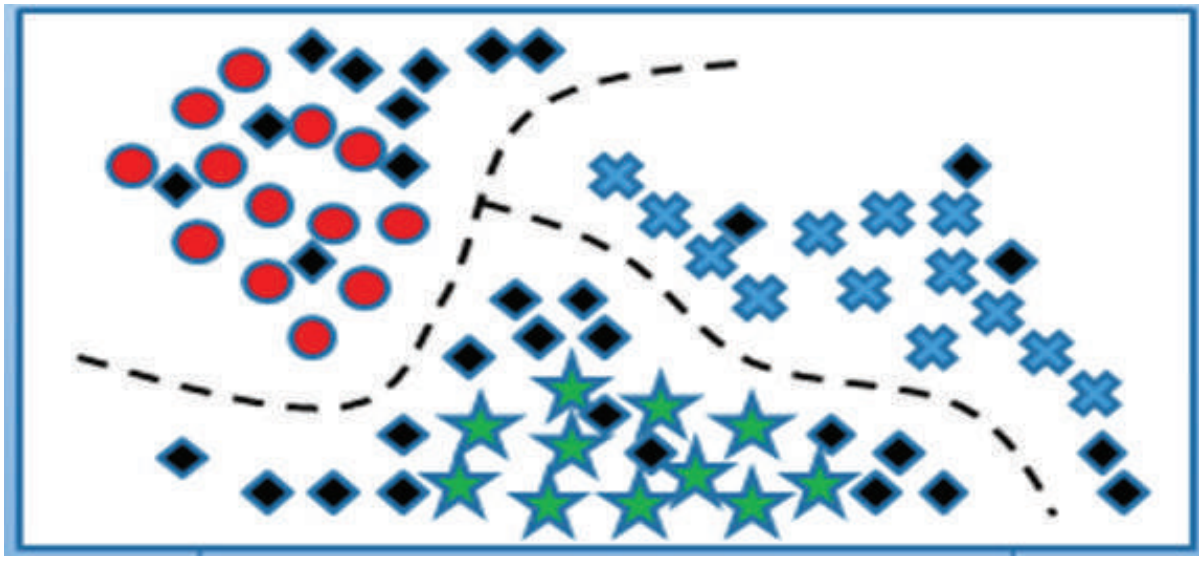

Figure 3: Graphical View of Semi-Supervised Learning Data

4) ***Reinforcement Learning [14][17]:*** It is used in decision making. For example chess game or the ability of a robot for making efficient actions/decisions.

B Supervised Learning

This is one of the four types of machine learning. In supervised learning, the nature of the data defines output on behalf of the feature/s. These outputs vary into two types 1)Classes(discrete

labels) 2)Regression(Real values). The data in which there are classes assigned for every input is known as classified data i.e the data is in classes/groups, therefore classification algorithms [16][17] are implemented on this kind of data. The data which has output as decimal values are handled by regression algorithms [17] for predictions.

Table 1: Classification Data

Return July	Return Aug	Return Sep	Return Oct	Return Nov	Positive Dec
-0.020517029	0.024675868	-0.020408163	-0.173317684	-0.025385313	0
-0.025321312	0.211290002	-0.580003262	-0.267141251	-0.151234568	0
-0.135384615	0.033391916	0	0.091695502	-0.059561129	0
-0.094	0.095290252	0.056680162	-0.096339114	-0.40511727	1
0.35530086	0.056842105	0.033602151	0.03626943	-0.085305106	1
0.274446938	0.538343949	0.127068167	-0.171428571	-0.195374525	1

The “PositiveDec” column denotes the binary classes [17]of the features.

Table 2: Regression Data

PT08.S2(NMHC)	NOX(GT)	PT03.S3(NOX)	NO2(GT)	PT08.S4(NO2)	PT08.S5(03)	T	RH	AH
1046	166	1056	113	1692	1268	13.6	48.9	0.7578
955	103	1174	92	1559	972	13.3	47.7	0.7255
939	131	1140	114	1555	1074	11.9	54.0	0.7502
948	172	1032	122	1584	1203	11.0	60.0	0.7867
836	131	1205	116	1490	1110	11.2	59.6	0.7888
750	89	1337	96	1393	494	11.2	59.2	0.7848
690	62	1462	77	1333	733	11.3	56.8	0.7603

The “AH” column is the output on behalf of the values of the rest of the columns in that single row. There are two types of regression models namely simple and multiple which comprise of two types of nature of data, linear and non-linear.

The data which is in textual format, algorithms like decision tree [17] is implemented on them. This nature of data lies in the context of classification but the classes, as well as the features, are in a textual format as shown on next page:

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Figure 4: Textual Data

The "Play" column shows the decisions on behalf of the features. A decision tree for the above data is shown below:

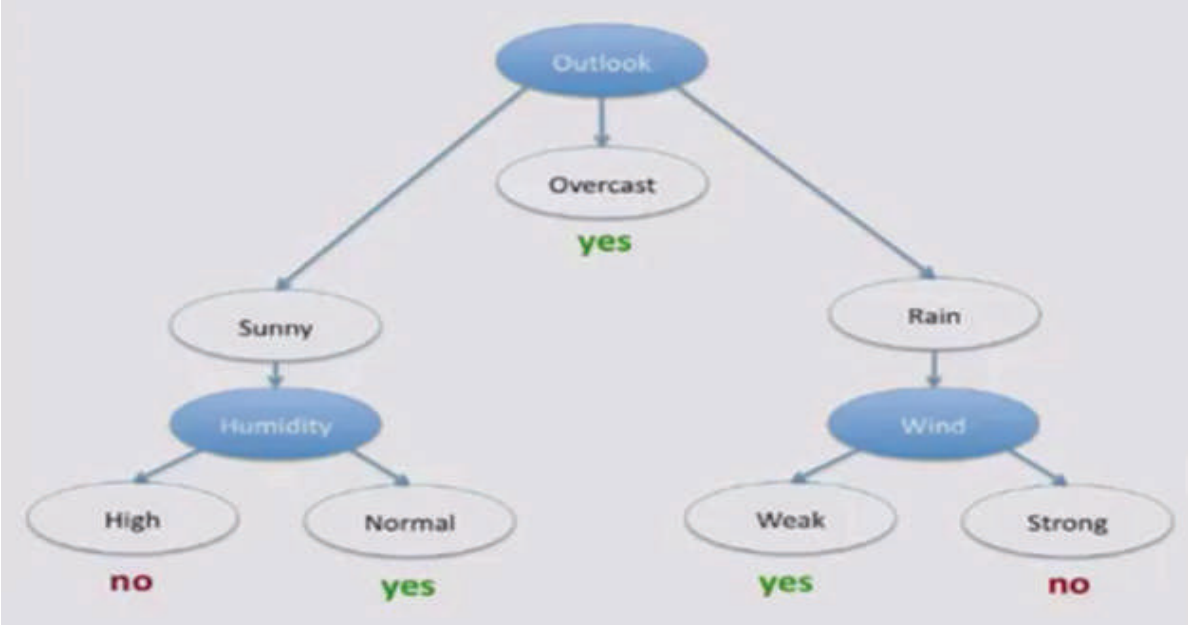

Figure 5: The Decision Tree of theTextual Data

3 Explanation & Filtering of Data

A Explanation of Data

This is the 5 years past data of Newyork (NYSE: TheNewyork Stock Exchange) stock market from date 2/2013 to 1/2018. There are 505 different stocks. Features include Date (current date of the stock price), Open (price of the stock at market open), High (the highest price reached in that day), Low (lowest price of that day), Close (price of the last trade of the stock), Volume (number of stocks traded) & Name (stock's ticker name). We are going to use the first 5 stocks from the data.

B Feature Selection

There are features which need to filter out as they contribute no benefit to our training, some of them will disturb the linear nature of the data if they are not filtered. We are going to remove the features of 'Volume', 'Date', 'Name' from the data.

C Nature of Data

To check the linearity of the data plotting is the optimal way of representing. Hence we check all 5 stocks data nature by plotting them in Figure 6.

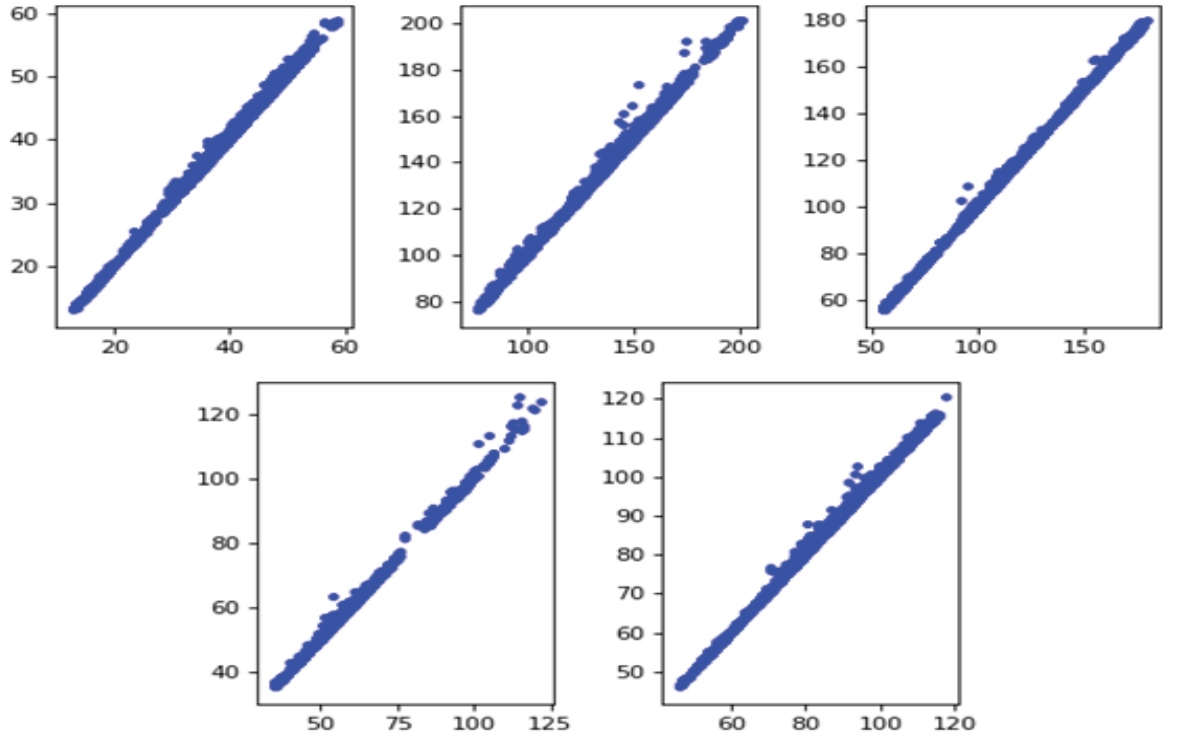

Figure 6: Graphs of 5 different stocks data

The above plots illustrate the linear nature of the data.

4 Classification Algorithm on Basis of Ranges

The classification algorithm is implemented in a supervised learning environment on linear data. Calculate the minimum and the maximum value of each group from any one of the features (as the data is linear therefore it can be calculated from any of the features). These minimum and maximum values will define the range of each group. Now that we have our ranges we can predict the class of an incoming new input/s. We will pick the same feature's value, which we used for calculating our ranges, to check the predicted class. The value lies between one of the ranges and the class of that specific range is assigned as the class of the new input. If the input value exceeds the maximum value of the highest range, then the class of the highest maximum value is assigned to the new input and the maximum value of the highest range is updated with the new input value. If the input value falls short of the lowest minimum value of the smallest range, then the class ofthe lowest minimum value among all the ranges is assigned as the new class and the minimum value of the smallest range is replaced by the new input value.

5 Clustering

The data of all the stocks are unlabeled, that means it is not classified into groups/clusters. There are many clustering algorithms for unsupervised learning. We have used the K-Means algorithm to cluster the data of each stock into 3 groups. After clustering, the plotting of all 5 stocks data is shown in Figure 7.

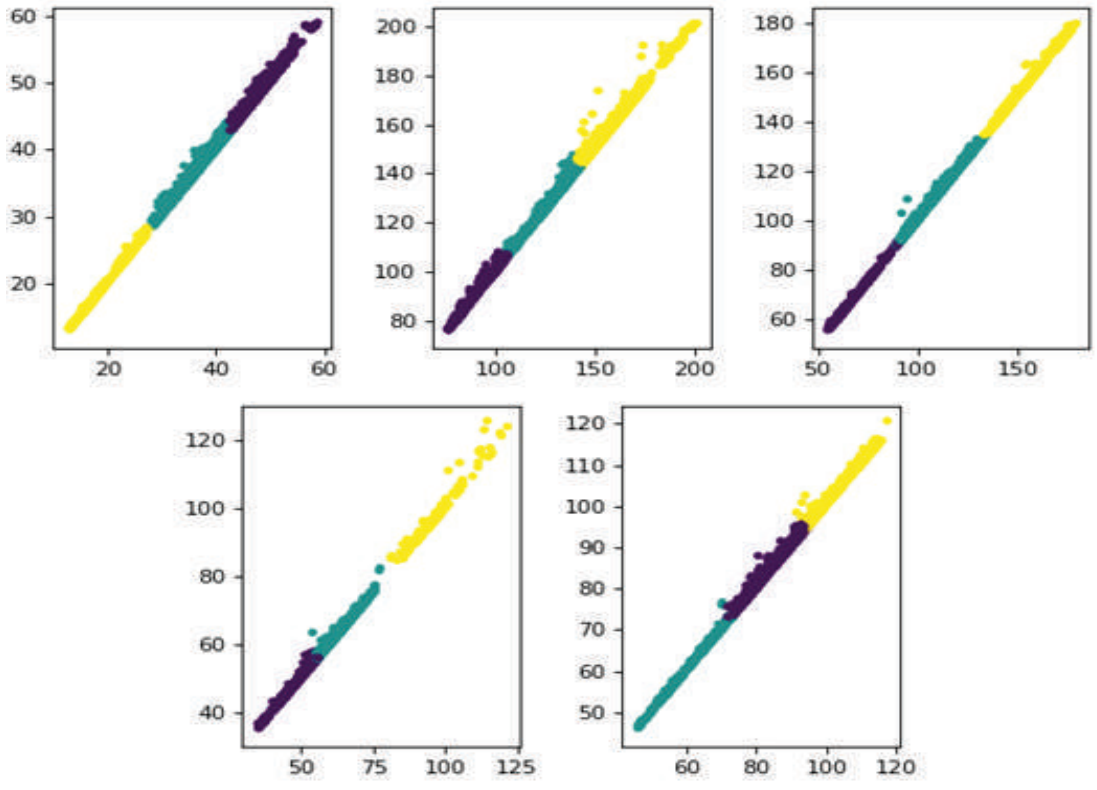

Figure 7: Linear data after K-Means Clustering

Now each stock is classified into 3 groups. K-Means classifies groups into numeric values. For instance, the lower part of the graph shown in each subplot is group '0', the middle part of the graph in each subplot is grouped as '1' and the last part as '2'. So there are 3 groups namely (0,1,2).

6 Experiments and Results

A Nature of Data

After features extraction [7] and classification, we have 4 features and their class, namely (Open, High, Low, Close, Groups). A sample image of the data is shown below in Figure 8 to explain the structure of the features:

Open	High	Low	Close	Group
153.88	154.770	153.31	154.08	0

Figure 8: Stock Features

We have split the data into 2 parts, 90% for training and 10% for the testing. The data is randomly selected for training and testing.

Now we select one feature from where the minimum and maximum values will be calculated. In our case, we selected the "close" feature for the calculation. For each group minimum and maximum values will be calculated from the "close" feature. The Minimum and Maximum value for each group of stock data are shown below in Figure 9:

For group 0 Min Max Values are : (152.05, 220.37)
For group 1 Min Max Values are : (216.05, 266.2)
For group 2 Min Max Values are : (258.07,309.91)

Figure 9: MinMax ranges of each cluster/group

We have our ranges/boundaries which will predict the class of the new inputs. The value of the "close" feature in the new input will be tested, that in which range or boundary it fits and then the class of the specific range is the new class for the input.

B Results

The best accuracy achieved up till now is 99.76%.The accuracy of the test data of the first stock is shown below in Figure 10.

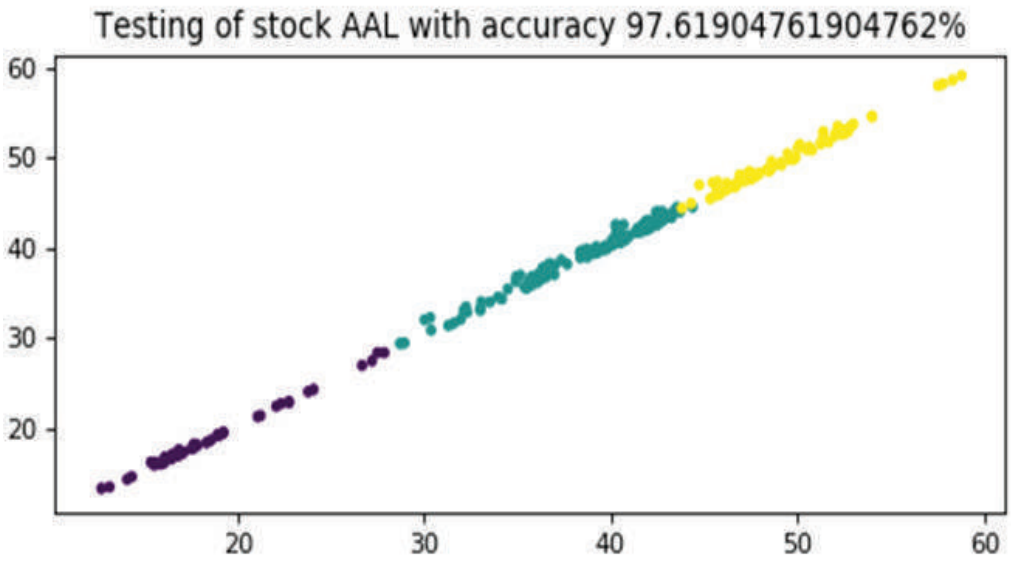

Figure 10: Stock 'AAL' accuracy 97.62%

For the second, third, fourth and fifth datasets of stocks, graphs are listed below:

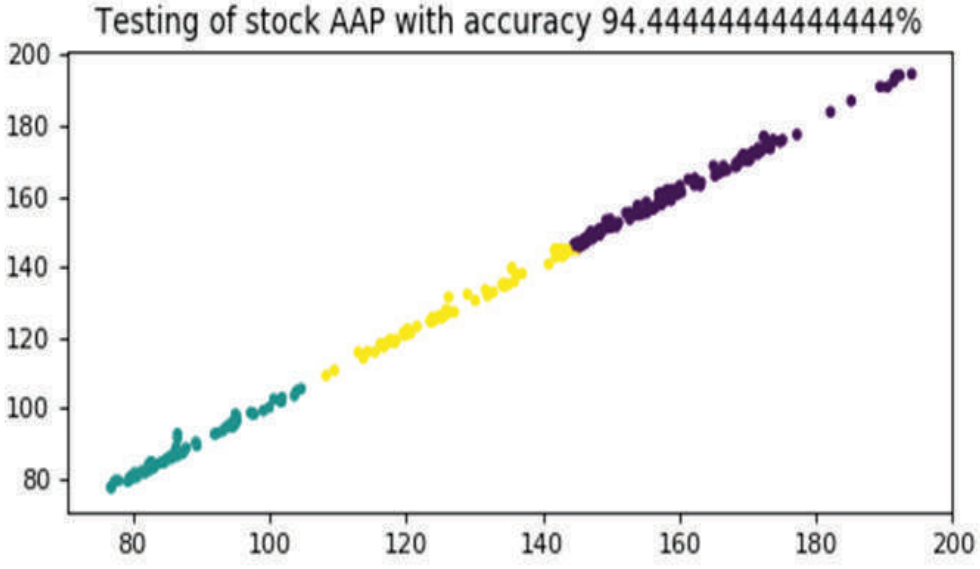

Figure 11: Stock 'AAP' accuracy 94.44%

The images of the above plots show that the accuracy is directly proportional to the linear nature of the data:

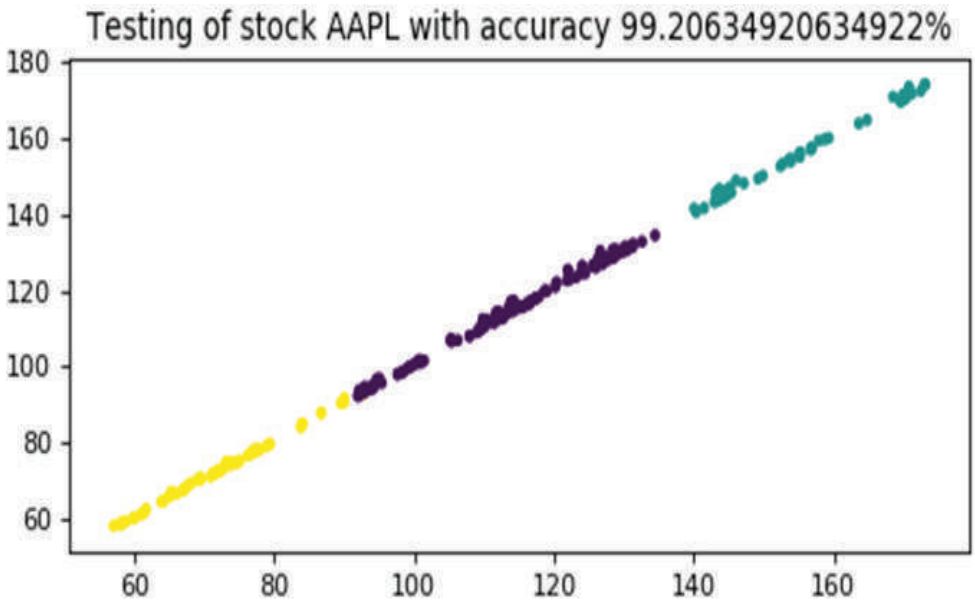

Figure 12: Stock 'AAPL' accuracy 99.21% (Highest Accuracy)

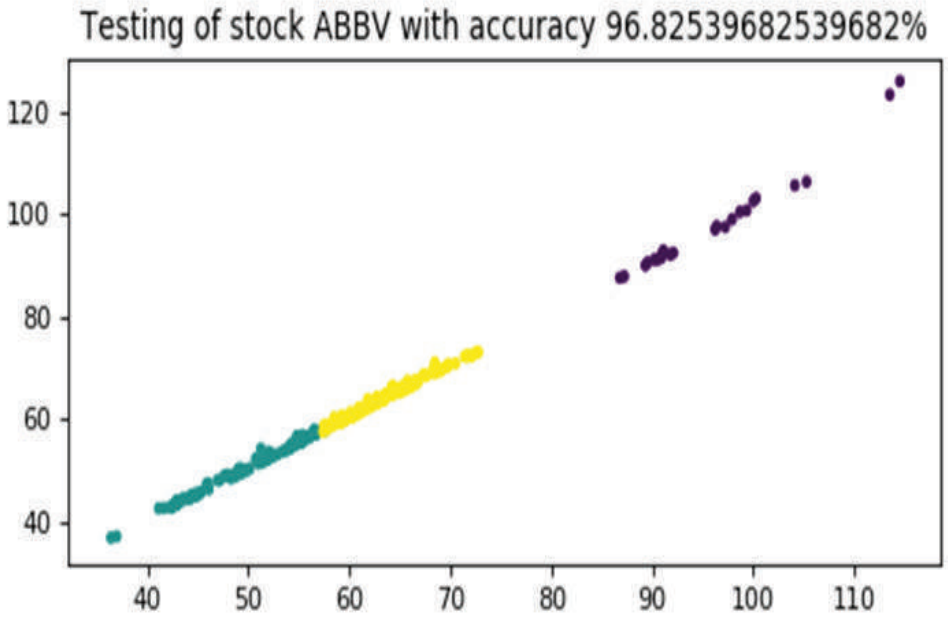

Figure 13: Stock 'ABBV' accuracy 96.82%

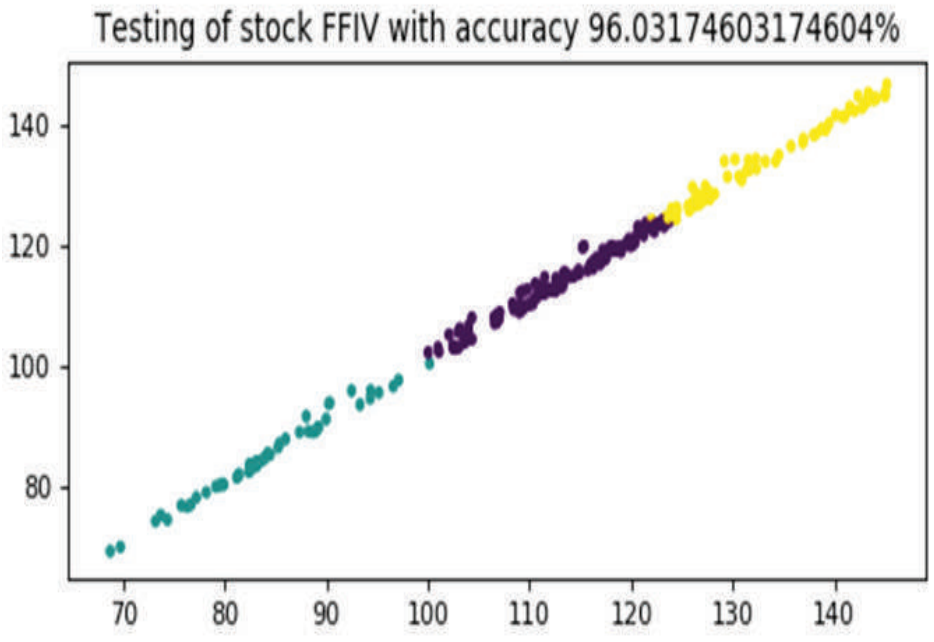

Figure 14: Stock 'FFIV' accuracy 96.03%

The images of the above plots show that the accuracy is directly proportional to the linear nature of the data:

Accuracy Nature of Data

C Experimental Environments

Table 3: PC Combination

CPU	Intel(R) Core(TM) i5-3340M CPU @ 2.70GHz 2.70 GHz
RAM	DDR3 8GB Ram
OS	Windows 7 Ultimate
TOOLS	Pycharm (python, scikit-learn, pandas, matplotlib, scipy)

7 Experiments and Results

This paper introduces a classification algorithm for supervised learning on linear data. On 5 stocks data which consists of Open, High, Low and Close as their features. The data is split into two parts, 90% for the training part and 10% for testing. As for the results, the accuracy percentages are all above 90%, giving a maximum accuracy of 99.20%. The advantage of implementing this model is efficiency, accuracy, and simplicity of the algorithm as other classification algorithms maybe costly on linear data. However, the restriction of the data being linear is a drawback of this model. So, the improvement of this algorithm efficiency on non-linear data is needed soon. For these improvements, we can develop a way of extracting optimal ranges from non-linear classification data as well.

References

[1] Thomas G. Dietterich, "Ensemble Methods in Machine Learning" in 2000. International Workshop on Multiple Classifier Systems, On 2000. LNCS 1857, pp 1–15.

[2] TaiwoOladipupoAyodele (2010)." Types of Machine Learning Algorithms", New Advances in Machine Learning, Yagang Zhang (Ed.), ISBN: 978-953-307-034-6, InTech

[3] Sahibsingh A. Dudani, "The Distance-Weighted K-Nearest-Neighbor Rule" in1976, IEEE Transactions on Systems, Man, and Cybernetics (Volume: SMC-6, Issue: 4), pp 325-327.

[4] J. A. Hartigan and M. A. Wong, "A K-Means Clustering Algorithm", in 1979, Journal of the Royal Statistical Society, Series C (Applied Science), Wiley for the Royal Statistical Society, pp 100-108

[5] Gleen W. Milligan and Martha C.Cooper, "Methodology Review: Clustering Methods ", in 1987, Ohio State University, Volume: 11 Issue: 4, pp 329-354.

[6] T. SoniMadhulatha, "An Overview On Clustering Methods", in 2012, IOSR Journal of Engineering, Vol. 2(4), pp: 719-725.

[7] Shigeo Abe, "Feature Selection and Extraction", in 2010, Support Vector Machine for Pattern Classification, pp 331-341.

[8] Eugene Charniak, "Introduction to Artificial Intelligence", in 1985, Brown University.

[9] Makrufa S. Hajirahimova and Aybeniz S. Aliyeva, "Review of Statistical Analysis Methods of Large-Scale Data", in 2015, 9th IEEE International Conference on Application of Information and Communication (AICT), pp 67-71.

[10] Jin Huang and Charles X. Ling, "Using AUC and Accuracy in Evaluating Learning Algorithms", in 2005, IEEE Transaction on Knowledge and Data Engineering (Volume: 17, Issue: 3), pp 299-310.

[11] Foram P. Shah and Vibha Patel, "A Review on Feature Selection and Feature Extraction for Text Classification", in 2016, IEEE International Conference on Wireless Communications, Signal Processing and Networking(WiSPNET), pp 2264-2268.

[12] C. Barras and J. –L. Gauvain, "Feature and Score Normalization for Speaker Verification of Cellular Data", in 2003, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03), Volume: 2, pp II-49.

[13] O. Chapelle, B. Scholkopf and A. Zien, Eds, "Semi-Supervised Learning (Chapelle, O. et al., Eds.;2006)[Book Revies]", in 2009, IEEE Transactions on Neural Networks (Volume: 20, Issue: 3), pp 542-542.

[14] Leslie Pack Kaelbling, Michael L. Littman and Andrew.W. Moore, "Reinforcement Learning: A Survey", in 1996, Journal of Artificial Intelligence Research 4, Volume: 4, pp 237-285.

[15] Jaime G. Carbonell, Ryszard S. Michalski, and Tom M. Mitchell, "Machine Learning: An Artificial Intelligence Approach", in 2013.

[16] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine Learning: A Review of Classification and Combining Techniques", in 2006, Artificial Intelligence Review, Volume: 26, Issue: 3, pp 159-190.

[17] Giuseppe Bonaccorso, "Machine Learning Algorithms", in 2017.

3D Printing using Fused Filament Fabrication

A. M. Khan[1] M. T. Khan[2] M. Faisal[3]

S. Tahir[4] G. Mustafa[5] Kalbeabbas[6]

Abstract

Fused Filament Fabrication (FFF) 3D Printing is a technique to directly obtain a real object, designed using CAD, by fusing the thermoplastic material at high temperature and movement of stepper motors as directed by the controller. 3D Printing prototypers and manufacturers face many problems while printing objects. They face the problem of not having 3D Printer repairing service (especially in less developed areas) and thus 3D printers become useless. Another major problem 3D Printing firms face is the transference of object file among many of 3D Printers. Moreover, last one its post-processing, especially re-colouring of the product in different colours and connectivity with a remote location. Thus, this study is carried out to overcome the majors' problems in the 3D printing manufacturing. This study project utilized an open source Marlin Firmware to design and make 3D Printer controller, ESP8266 to provide Wireless connectivity, and dual Extruder setup tailored to make dual colours to make 3D Printer a colourful. The study shows that the accuracy comparison of proposed printed Cube with respect to CAD Cube is about 96.66%.

Keyword: Thermoplastic material, CAD, Arduino Mega, Multi Colour, Wi-Fi ESP8266

1 Introduction

Additive manufacturing, colloquially known as 3D printing, is an umbrella term for technologies that allow the production of physical goods from the ground up. In the case of the usual creation of equipment - machines, saws, drills and rotary motors - taking a piece of raw material and cutting it into a frame shape (manufacture by subtraction), the 3D printer is the opposite. As the terminology suggests, a 3D printer will include raw materials in a small layer at any time, creating the entire project. Unlike plastic infusion molding, 3D printers do not require excessively tall shapes, but only high-level structural records containing ideal project data [1]. The first additions to materials and material-making materials were made in the 1980s. In 1981, Hideo Kodama, of the Nagoya Industrial Research Institute, considered two additive technologies for the manufacture of three-dimensional plastic models with polymers Photo-curable thermosets. The innovations that have been used so far in most 3D printers - especially professional and buyer-designed models - demonstrate the extraordinary use of plastic profiles created by S. Scott Crump in 1988 and promoted by his Stratasys organization. Promote its first FDM machine in 1992 [2].

[1,3,4,5,6]*Sir Syed University of Engineering & Technology, Karachi | abidmk@ssuet.edu.pk*
[2]*Indus University, Karachi | mtk.masters@gmail.com*

The remainder of this paper is organized as follows: Section 2 presents the Conventional 3D Printers. Section 3 explains the types of FFF 3D Printers. Proposed Methodology is presented in Section 4. Section 5 covers the results and discussion. Conclusion of the work is presented in Section 6.

2 Conventional 3D Printers

Recently referenced 3D printing is a generic term that includes unique advances. As part of more than a dozen 3D printing technologies, it can be divided into three main categories, each with its own advantages and disadvantages. The advancement of the main category depends on the liquid or semi-fluid material extruded through the nozzle of the print head into the desired shape. This classification is called material extrusion and most buyer-level gadgets are available. Although any material that can be ejected through a syringe and retains its shape later can be printed along these 3D lines, most material extrusion printers use thermoplastics as fibers. Some of the 3D printers in this category have print heads that can use a variety of materials in one location.

In the second category, called photo polymerization, 3D printers use lasers or other light sources to define or solidify a fluid called a photopolymer. In some procedures, the light source follows the ideal shape in a photopolymer tar box, although it is suitable for the violin because the entire layer must be on the ground and glued without delay. In some applications, the photopolymer is splashed and released directly into the desired shape. Some buyer-level 3D printers use photo aggregation, but such 3D printers are typically used by experts.

Finally, there is a combination of granular materials. Here, the 3D printer uses a laser or blanket to join the fine powders to make an article. Printers in this category usually have a large amount of powder contiguous to the actual printing platform. The roller applies a thin layer of powder to the printing platform and then follows the desired shape with a laser or a fastener. When the main layer is completed, the platform lowers and re-joins the next layer of powder. Printers in this category may use plastic, metal, pottery or even glass as printing material. Currently, no customer-level 3D printer uses binding of granular material [3]. The study project is based on innovative manufacture of fused filaments to create 3D objects layer by layer. The FFF process is described in detail below.

A Fused Filament Fabrication

FFF is a solid-based additive manufacturing (AM) technology. The FFF system builds parts layer-by-layer by depositing semisolid molten polymeric materials in the shape of the thin filament (or road/bead) via a computer-controlled robotic extruder. Figure 1 shows the process of fusion and extrusion of the filament. Heater block provides enough temperature to melt the filament, and the motor is used to push the filament downwards, thereby filament is continuously melted and re-solidifies on the build surface. FFF 3D Printers make motions using stepper motors. Four stepper motors are used basically for the motion, motion along the x-axis, the y-axis, the z-axis and for providing extrusion force. Table I compares characteristics of FDM, SLA and SLS 3D Printing technology.

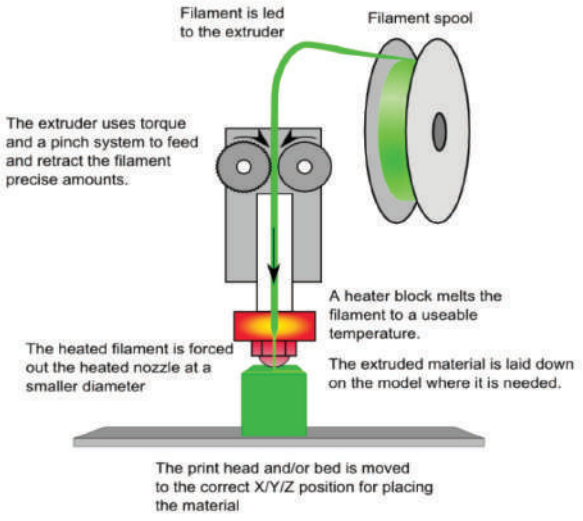

Figure 1: Illustration of Fused Filament Fabrication process

Table 1: Context of our Search Queries

	Fused Filament Fabrication (FFF)	Stereo Lithography (SLA)	Selective Laser Sintering (SLS)
Resolution	★★☆☆☆	★★★★★	★★★★☆
Accuracy	★★★★☆	★★★★★	★★★★★
Surface Finish	★★☆☆☆	★★★★★	★★★★☆
Complex Designs	★★★☆☆	★★★★☆	★★★★★
Ease of Use	★★★★★	★★★★★	★★★★☆
Pros	Fast Low-cost consumer machines and materials	Great value High accuracy Smooth surface finish Range of functional applications	Strong functional parts Design freedom No need for support structures
Cons	Low accuracy Low details Limited design compatibility	Average build volume Sensitive to long exposure to UV light	Rough surface finish Limited material options
Applications	Low-cost rapid prototyping Basic proof-of-concept models	Functional prototyping Dental applications Jewelry prototyping and casting Model making	Functional prototyping Short-run, bridge, or custom manufacturing
Print Volume	Up to ~200 x 200 x 300 mm (desktop 3D printers) printers)	Up to 145 x 145 x 175 mm (desktop 3D printers)	Up to 165 x 165 x 320 mm (bench top 3D printers)

3 Types of Fused Filament Fabrication (FFF) 3D Printers

There are variants of FFF type 3D Printer. They differ by their working mechanism, mathematics and machine parts. There are four basic types of FFF 3D Printers:

A) Cartesian FDM 3D Printers.

B) Delta FDM Printers

C) Polar 3D FDM Printers.

D) FDM 3D Printing with Robotic arms.

A Cartesian FDM 3D Printers

The Descartes 3D printer is currently the best known 3D FDM printer. Given the Cartesian arrangement of arithmetic, the innovation uses three axes: X, Y, and Z to determine the correct position and alignment of the print head. With this type of printer, the print tray generally moves only on the Z axis and the print head operates in both dimensions on the X-Y plane [5].

B Delta FDM Printers

These printers are gaining increasing attention in the FDM 3D printing industry and two research institutes in Switzerland are developing, including six-axis 3D printers based on Delta's innovation. These machines use the Cartesian direction. This includes a circular plate that is attached to the extruder and the extruder is attached to three triangular focal points. Each of the three points on this point goes up and down, which determines the position and orientation of the print head. Delta printers are designed to speed up the printing process. In any case, many people think that this printer is not as accurate as a traditional Cartesian printer. The Descartes and Delta 3D printers are not very unique. It is important that each component can be moved on a print bed. In a Cartesian 3D printer, each component can move in one direction, while in a Delta 3D printer, the print head can move in any path, but the print board does not move [6]. The difference between Cartesian 3D printing mechanisms and Delta FFF is illustrated in Figure 2.

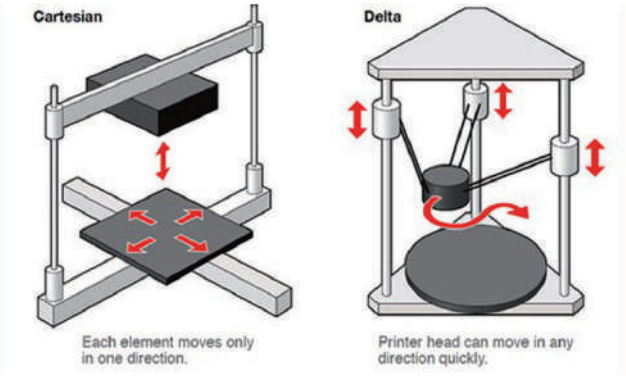

Figure 2: Cartesian vs. Delta printers

C Polar 3D FDM Printers

The positioning of the polar 3D printer is not determined by the organization X, Y, and Z, but by the angle and the length. This means that during this time, when the extruder goes up and down, the plate turns and moves. The main advantage of Polar FDM 3D printers is that they only have two motors, whereas Cartesian printers require three. In the long term, Polar printers have more vitality and can make larger documents while taking up less space [7].

D FDM 3D Printing with Robotic Arms

Robotic arms are often known to accumulate parts on industrial lines, especially in large automobile factories. Although 3D printing has begun to add robotic arms to their build process, it is particularly important in 3D printing of houses and structures, and this innovation is being improved [8]. Although the printing process is not often used, this FDM printing technology begins to see the expansion used. Indeed, the program does not work with the plates to make them more portable. In addition, due to the adaptability when positioning the 3D FDM print head, the requirements for the manufacture of complex structures are low. It should be noted, however, that the final print quality is comparable to that of traditional Cartesian printers.

4 Types of Fused Filament Fabrication (FFF) 3D Printers

With advancement in 3D Printer technologies engineers are working on the Multicolour printed object since there is a need for designers, entrepreneurs and for teaching aids too. There are many techniques to print a multi-colored object, which varies with complexity, ease of use cost and time, with each, is associated is many advantages and disadvantages. With the simplest and cost efficient is printing objects separately. The following section discusses the many possible ways to do it. The functional block diagram of the proposed 3D Printer working is shown in Figure 3.

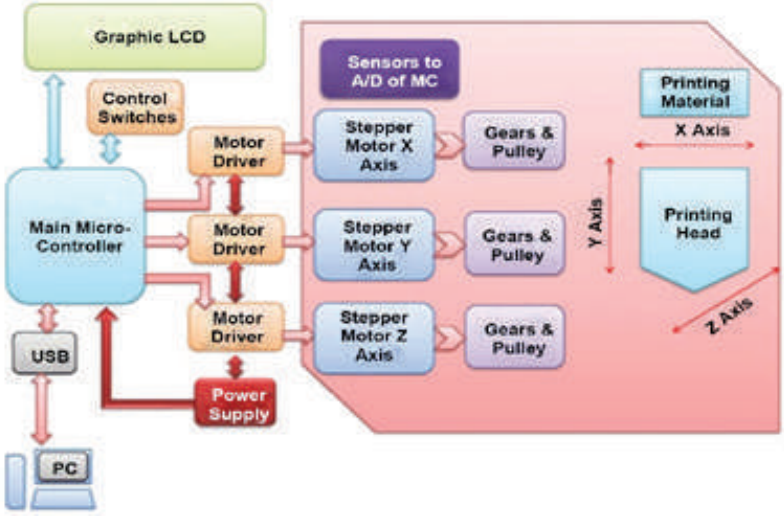

Figure 3: Functional block diagram of the proposed 3D printer

A Printing each Colour Objects Separately

In this technique, objects are first pre-processed using CAD software which will be breakdown the 3D model into different colour models. With 3D models require great labour to break the model into colours. There are many advantages and disadvantages to it. The biggest advantage is that it is cost efficient, that anyone having 3D Printer with single extruder can do it without any further hardware advancements. The disadvantage with these techniques is that it requires pre- and post-processing which is a delicate task [9].

B Pausing Mid-Print

In this technique, the printing is paused at some interval and filament is changed, and then resumed, thus the multicolour object is obtained. It is suitable where layered colour 3D model is printed, and is not suitable where colours are mixed at different Z-axis points. It is advantageous than the previous technique discussed in that it requires lesser pre-processing and object is obtained as a single [10].

C Multi-Extruder

Multi-Extruder comes Multiple Extruders with each extruding different colour, and there is no limit to the number of colours except space which is required for the extruder. The G-code is responsible for each colour change. Multi-Extruder has advantages over the other two discussed so far, that in it there is no need of worry to change the filament and then resume, but just begin and the end product will be ready soon. However, it has many disadvantages too. It reduces the size of Print bed that when, suppose a two-extruder head is used, the Extruder at right will not reach at left corner unless the left extruder is moved outside the print bed, thus the size of bed equivalent to the width of the extruder must be reduced and vice versa for the right corner, reducing the width of the print two times the width of extruder head. It is also unsuitable to create gradient colours [11].

D Multi-Spool Single-Extruder

In this technique, multiple spools are used and each spool holds a filament. Attached with each spool is an extruder motor that pushes the filament to the single nozzle when detected by the software. Advantages of both multiple extruder techniques and of the classic are achieved through it, with no space shrinkage and using a single nozzle to print multiple filaments. But like others it has some disadvantages too it wastes filament when changing the filament colour, thus there is more waste in it, thus it increases prototyping cost [12].

E Wi-Fi Connectivity

Using Wi-Fi Connectivity 3D Model may be transmitted wirelessly instead of using SD card and USB cable reducing space. Many a company produces 3D printed product, it works 24/7 nonstop, which requires many USB ports which a single PC cannot handle. The solution is to implement Wi-Fi connectivity. The Wi-Fi may be implemented using Wi-Fi modules such as ESP8266.

ESP8266 is a Wi-Fi microchip with full TCP/IP stack and microcontroller capability module as shown in Figure 4. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections.

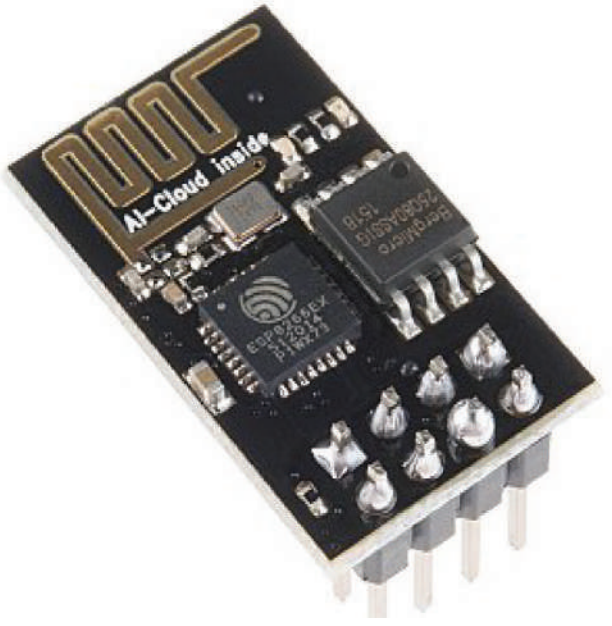

Figure 4: ESP8266 a Wi-Fi Module

F Schematics of wiring ESP8266 with RAMPS

Figure 5 and 6 shows pin-out of RAMPS and ESP8266. The ESP8266 TX pin is to be connected to TX pin of RAMPS and RX pin to be connected to RX pin of RAMPS.

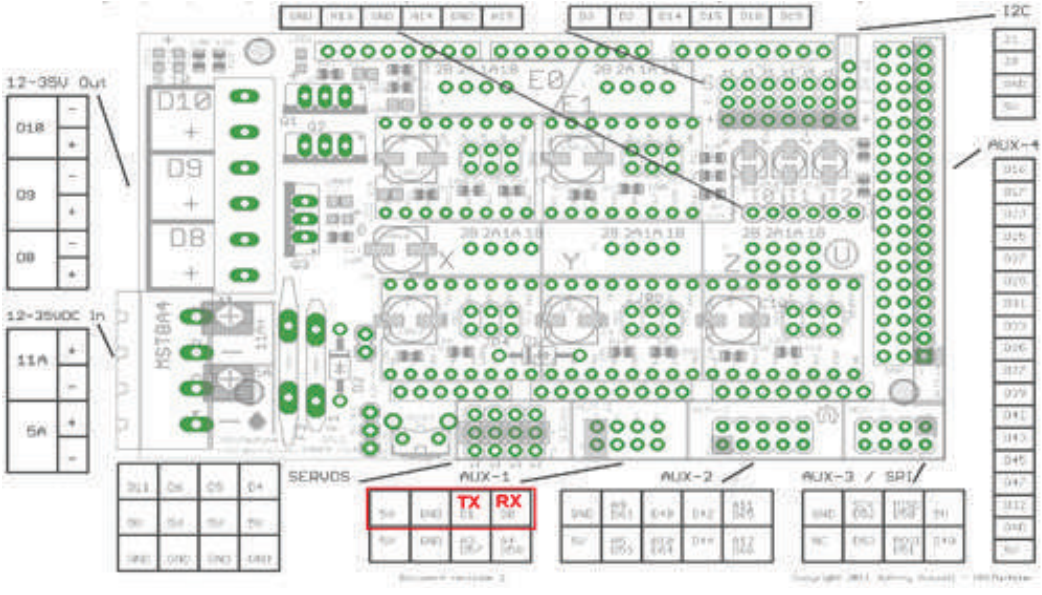

Figure 5: Pin assignment of RAMPS board for Wi-Fi

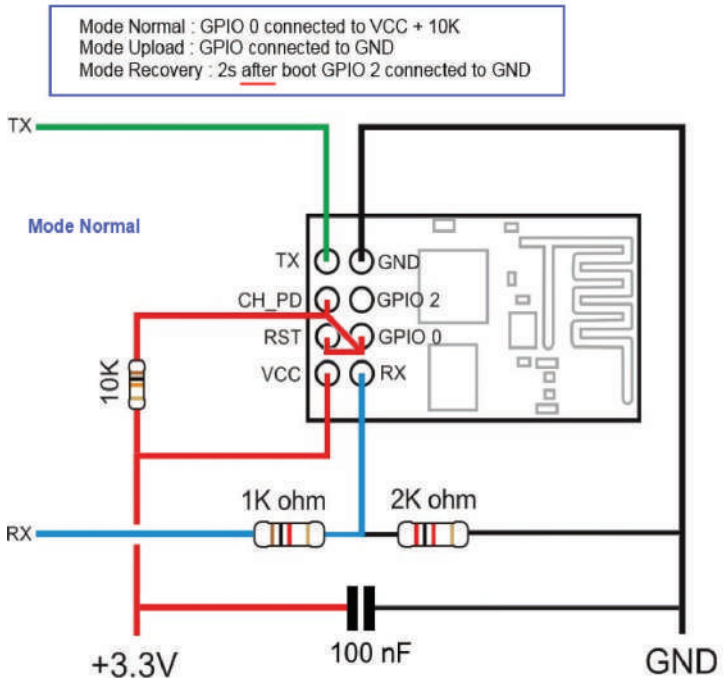

Figure 6: ESP8266 connections with RAMPS board

5 Results and Discussion

This section discusses the three major parameters of 3D printed object, which are as follows:

A Printing at different Infill ratios

Printing with different Infill ratios increases the weight of the object, increasing the time to print. 3D Prints with lesser infill ratios prints faster than higher infill ratio. Usually, 20-25% is sufficient for the object to be of good Quality. 100% infill completely fills the object using much filament to produce. Figure 7 shows the print with different print ratios.

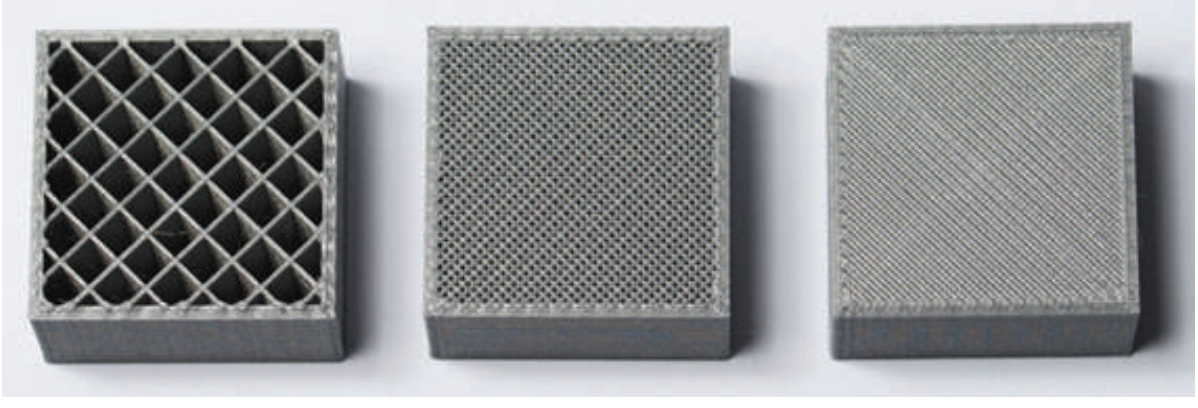

Figure 7: 3D Printing Cube with different Infill ratios

B Printing Time

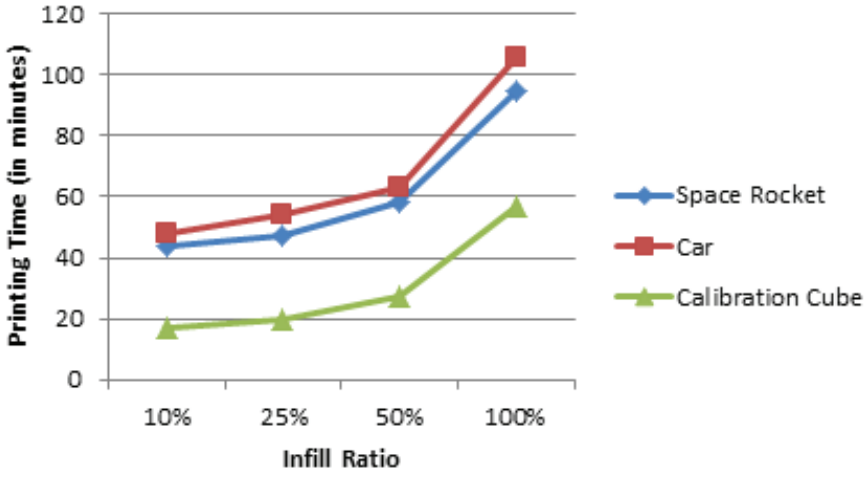

Figure 8: Printing Time vs. Infill ratio

Figure 8 shows the different printing time against infill ratio for three different 3D Printed models i.e. Space Rocket, Car and Calibration Cube. The graph shows the higher infill ratio tends to increase printing time and weight of the printing object. The actual 3D printed objects can be seen in Figure 9.

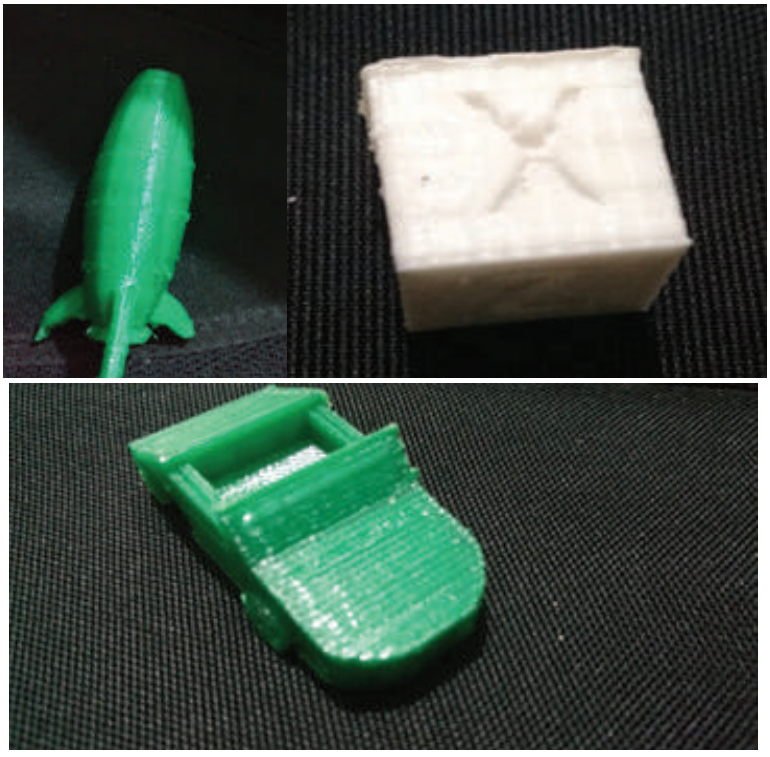

Figure 9: Calibration Cube, Space Rocket, Car (Top-left to bottom).

C Accuracy

A 20x20x20 mm cube is printed with the parameters given below for quality analysis.

Extruder Temperature: 190 C

Flow: 100%

Print speed: 50 mm/s

Infill speed: 60 mm/s

Travel speed: 100 mm/s

Wall thickness: 0.8 mm

Top/bottom thickness: 0.8 mm

Layer height: 0.1 mm

Infill Density: 20%

Build plate adhesion type: Brim

Table II displays the accuracy of the printed Cube against 3D CAD Cube model. The comparison in three different axis shows that the average accuracy of the proposed method is about 96.66% for 3D Cube printing.

Table 2: Accuracy comparison of printed Cube to CAD Cube model

FFT	X-measurement	Y-measurement	Z-measurement
Input Test Cube	20 mm	20 mm	20 mm
Output Test Cube	19 mm	19.5 mm	19.5 mm

6 Conclusion

In this paper, the design of general 3D Printer main-board using RAMPS board and Arduino mega discussed. The design and analysis suggest that a higher infill ratio tends to increase printing time and weight of the printing object. The accuracy comparison of printed Cube with respect to CAD Cube is about 96.66%. Moreover, the addition of Wi-Fi to the RAMPS and methodology of printing with multicolour/multi-material capability has enhanced the performance of this study project. In future work, the quality of 3D printing can be further improved.

Acknowledgment

The authors would like to thank Sir Syed University of Engineering & Technology, Karachi for providing research facility in this paper.

References

[1] Ngo, Tuan D., et al. "Additive manufacturing (3D printing): A review of materials, methods, applications and challenges," Composites Part B: Engineering vol. 143, 2018.

[2] Nale, Swati B., and A. G. Kalbande. "A Review on 3D Printing Technology," International Journal of Innovative and Emerging Research in Engineering vol. 2, issue 9, 2015.

[3] Mark, Gregory Thomas, et al. "Multilayer fiber reinforcement design for 3D printing." U.S. Patent No. 9,688,028. 27 Jun, 2017.

[4] Goyanes, Alvaro, et al. "3D printing of medicines: engineering novel oral devices with unique design and drug release characteristics," Molecular pharmaceutics vol. 12, issue 11, 2015.

[5] Shirazi, Seyed Farid Seyed, et al. "A review on powder-based additive manufacturing for tissue engineering: selective laser sintering and inkjet 3D printing," Science and Technology of Advanced Materials vol. 16, issue 3, 2015.

[6] Taufik, Mohammad, and Prashant K. Jain. "A study of build edge profile for prediction of surface roughness in fused deposition modelling," Journal of Manufacturing Science and Engineering", vol. 138, issue 6, 2016.

[7] Dudek, P. F. D. M. "FDM 3D printing technology in manufacturing composite elements," Archives of Metallurgy and Materials, vol. 58, issue 4, 2013.

[8] Mueller, Stefanie, et al. "Wire Print: 3D printed previews for fast prototyping," Proceedings of the 27th annual ACM symposium on User interface software and technology. ACM, 2014.

[9] Meisel, Nicholas A., Amelia M. Elliott, and Christopher B. Williams. "A procedure for creating actuated joints via embedding shape memory alloys in PolyJet 3D printing," Journal of Intelligent Material Systems and Structures, vol. 26, issue 12, 2015.

[10] Hickman, Mark S. "Printing of pixel locations by an ink jet printer using multiple nozzles for each pixel or pixel row." U.S. Patent No. 4,963,882. 16 Oct., 1990.

[11] Andy, (2015), One nozzle with multiple spools of filament, [Online]. Available: https://www.scan2cad.com/cad/multicolor-3d-printing/.

[12] CNXSOFT, (2016), Anet A8 3D printer specifications, [Online]. Available: https://www.cnx-software.com/2016/11/03/anet-a8-diy-3d-printer-could-be-aworthwhile-first-3d-printer-for-156.

Next Release Problem: A Systematic Literature Review

Umer Iqbal[1] Khubaib Amjad Alam[2]

Abstract

Agile software development is a widely used software development methodology which welcomes highly changing customers' requirements. In Agile software development process, the whole software is delivered into a series of small releases. Each release incorporates a subset of whole software requirements. The selection of requirements to be incorporated in the next release is a complex activity. It was first termed as "Next Release Problem" by Bagnal. Many techniques were proposed later on to solve NRP The main objectives of this research are: (1) to classify NRP papers according to four criteria: techniques used, datasets used, objectives (either single or multiple), publication channels and trends; and (2) to analyze these studies from three perspectives: study objectives, optimization techniques to solve NRP and limitations of study. We performed a systematic literature review on NRP studies published in the period 2010-2018 and reviewed them on an automated four electronic databases. We identified a total of 27 studies published between 2010 and 2018 and classified them on predefined classification criteria. Based on the findings of this research it is concluded that multi-objective optimization techniques are the most widely used techniques. Among multi-objective optimization techniques applied in the context of NRP, NSGA-II provides the best solution both in term of convergence speed and solution quality while for the single-objective optimization problem, Simulated Annealing provides promising results. It is also observed that customer's satisfaction is widely used objectives to be maximized in either single objective or multi-objective optimization techniques. Furthermore, 10 real-world datasets were identified during this research. It is observed that the latest optimization techniques are given less attention to solving NRP which have shown promising results in many cases as compared to techniques applied to cater NRP.

Index Terms: Next Release Problem, Next Release Planning, Release Planning, Systematic Literature Review

1 Introduction

The software development process is a process of breaking down software development work into distinct phases to carry out the development process more competently. Many different software process models are proposed until now but the agile software development process gained the most reputation among all. A recent survey showed that almost 70 % of companies use agile sometimes, often and always [1]. In Agile, whole software is delivered in the form of small releases and each release incorporates a small number of whole system requirements. The selection of requirements which will be chosen in the next release is a major challenge.

[1] *National University of Computer and Emerging Sciences, Islamabad | umerpervaiz12@gmail.com*
[2] *National University of Computer and Emerging Sciences Islamabad | khubaib.ajmad@nu.edu.pk*

The selection involves taking care of many criteria's such as customer satisfaction, overall development cost, risk, development time on the basis of which the requirements are chosen for the next release. The Bagnall [2] termed this problem as "Next Release Problem" and proposed a solution based on single objective optimization. He also concluded that NRP is NP-Hard problem and thus the best way to solve the problem is to use heuristic methods. Over time many different techniques and methods are applied on different datasets to solve NRP.

Few studies provide a detailed overview of the techniques proposed for the resolution of NRP. Another problem in the previous studies is that each study focuses on some specific method or technique and does not cover the entire domain. The main goal and purpose of this research is to introduce a broader and precise overview of almost all of the commonly used latest techniques in NRP in the form of a Systematic Literature Review. We followed the guidelines of Kitchenham [3].

The organization of the paper is as follows: The detailed steps and the structured strategy of SLR is described in section 2. Section 3 contains the presentation and discussion of results. Finally, the conclusion and future directions are given in section 4. Section 5 contains the references.

2 Systematic Literature Review

A systematic literature review was conducted by following the guidelines of Kitchenham [3] and the collected data is analyzed in an unbiased and structured fashion. The first and the basic step to start the process of SLR was the formulation of protocol that was designed and structured by Umer Iqbal and reviewed by Dr. Khubaib Amjad Alam. Now the steps performed in SLR are described in the next sections.

A Research Questions

The research questions are given in Table I.

Table 1: Research Questions

RQ #	Research Question	Motivation
RQ 1	What are the existing methods, techniques, and algorithms proposed to cater to the next release problem?	The aim is to identify and compare existing methods and algorithms that are proposed to solve the next release problem.
RQ 3	What datasets are used in the context of NRP?	The aim is to identify different datasets used in the context of NRP and their sources
RQ 4	What is the overall research productivity of next release problem?	The aim is to identify the overall research productivity and to identify different research groups working on next release problems.

B Electronic Databases

Table 2 shows the database and the online link to that database.

Table 2: Electronic Databases

Identifier	Database	URL
ED1	IEEE Xplore	http://ieeexplore.ieee.org/
ED2	ACM	http://dl.acm.org/
ED3	Science Direct	http://sciencedirect.com/
ED4	Springer Link	http://link.springer.com/

The studies that were the part of this research activity were from a time span of 2010 to 2018. The digital libraries that were considered are IEEE, ACM, Science Direct, Springer and Google Scholar databases based on title, abstract, and keywords.

C Search Strategy

The first part of this step is to identify the major key terms and the synonyms and alternatives of these terms. The idea behind the formulation of these terms is to construct a query string that will help to continue the remaining search method. A process consisting of three steps was followed to find the relevant studies to answer the research questions [30]. In the first step, the search string was formed. In the second step, we applied this search string on the selected digital libraries to get the required papers. In the third step, it was made sure that no relevant paper was missed.

Table 3: Search String

Database Name	Search String
IEEE Xplore Digital Library	(Next Release Problem) OR (Next Release Planning)
ACM Digital Library	Content.ftsc: ("Next Release Problem " OR " Next Release Planning ")
Science Direct	("Next Release Problem " OR " Next Release Planning ")
Springer	("Next Release Problem " OR " Next Release Planning")

Table 3 contains the search strings against all the four electronic databases that were used to find the studies for this research. As each database is different from each other so search strings are also different for each database.

D Search Process

In order to make sure that we were not leaving any related study, a two-stage search process [30] was adopted:

- **Initial search stage**

Here, we used the proposed search terms to search for primary candidate studies in the four electronic databases. The retrieved papers were grouped together to form a set of candidate papers.

- **Secondary search stage**

In this step, we reviewed all the studies retrieved after title based search where we read the abstracts of the remaining studies and based on the abstract the studies which were not relevant were excluded and the studies that passed this search qualified for the full-text reading.

E Study Selection Process

This step was designed to get the most relevant studies which were retrieved from five electronic databases in order to answer the research questions. The selection procedure which is given in Figure 1, consists of the following basic steps:

- Initial records
- Title based records
- Abstract based records
- Full article based records

F Inclusion and Exclusion criteria

Based on the above criteria if the study meets the inclusion criteria and none of the exclusion criteria is met then such a study is further moved to the next stage that is quality assessment criteria.

Table 4: Inclusion & Exclusion Criteria

	Inclusion and exclusion criteria
	Inclusion criteria
IC1	Studies related to next release problem
IC2	Articles from peer-reviewed publication venues
IC3	The inclusion of studies from 2010 to 2018
IC4	The inclusion of the most recent article in case of multiple studies on the same theme
	Exclusion criteria
EC1	Articles that are not in the English language
EC2	Editorial, short papers, posters, and extended abstracts will not be included

G Quality Assessment criteria

The quality criteria that is designed for this SLR is given in this section.

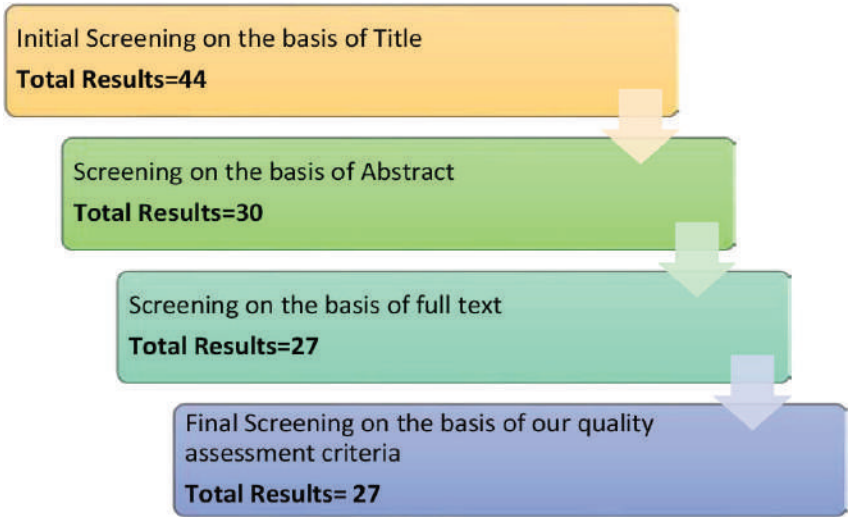

Figure 1: Study Screening Process

Table 5: Quality Assessment Criteria

QC #	Question	Score
QC1	Is the study has clearly defined goals and objectives?	Y|N|P
QC2	Is research in paper assist the aim of next release problem?	Y|N|P
QC3	Is the study propose valid or novelty technique/method?	Y|N|P
QC4	Are limitations of study explicitly stated?	Y|N|P

3 Results & Discussion

This section contains the results and discussion related to the research questions presented in Table. I

RQ1: What are the existing methods, techniques, and algorithms proposed to cater to the next release problem?

The detailed view of the techniques, contribution type that is used in the studies with references and number of studies for each technique are shown in Table 6.

Table 6: Techniques Used to Solve Nrp

Technique	No. of Studies	References of Studies
NSGA-II	10	[3],[5],[6],[16],[18],[23-29]
Ant Colony Optimization (ACO)	4	[4],[13],[17],[26]
Genetic Algorithm (GA)	6	[11],[13],[16],[19],[27],[29]
Simulated Annealing (SA)	8	[4],[11],[16],[18-19], [22],[27],[29]
GRASP	3	[4],[13],[26]
MOEA	4	[3],[5-6],[17]
Hill Climbing	2	[8],[24]
Others	20	[3-16],[18],[20-21],[28-29]

The abbreviations of the techniques used in Table. 6 are given in Table 7.

Table 7: Abbrivation of Techniques

Technique Name	Abbreviation
Non-Dominated Sorting Genetic Algorithm	NSGA-II
Ant Colony Optimization	ACO
Genetic Algorithm	GA
SA	Simulated Annealing
HC	Hill Climbing
GRASP	Greedy Randomized Adaptive Search Procedure
MoCell	Multi-Objective Cellular Genetic Algorithm
MOEA	Multi-objective Evolutionary Algorithm
Others	Others

The graphical representation of the techniques and their usage in percentage is shown in Figure 2.

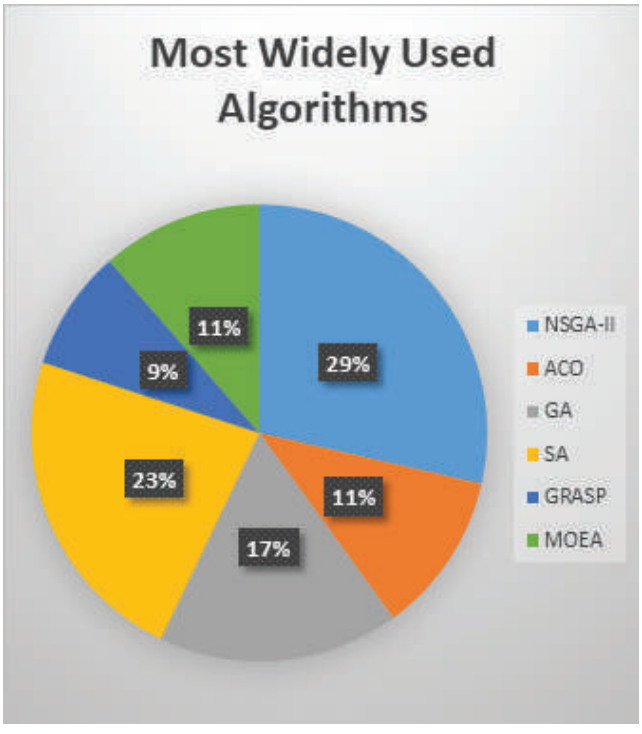

Figure 2: Usage Graph of Techniques

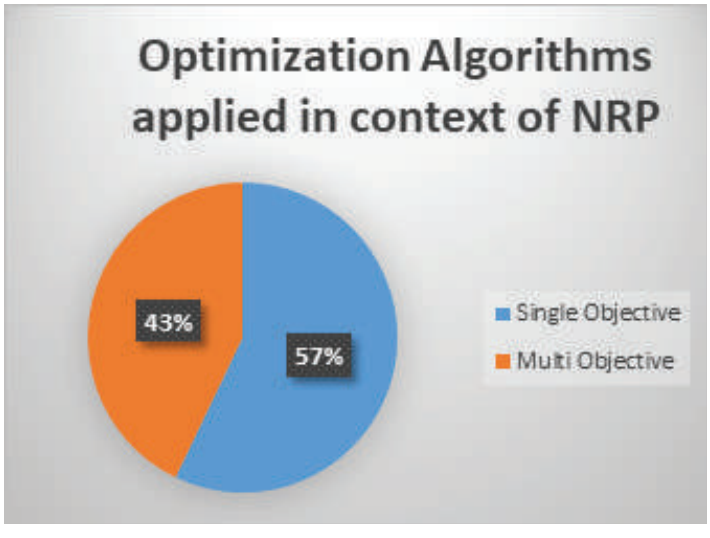

Figure 3: Frequency of Optimization Algorithms

It can be clearly seen in Fig 2 that NSGA-II is most widely used multi-objective optimization technique in the context of NRP while Simulated Annealing (SA) is most widely used in the context of Single Objective NRP. Other techniques like ACO, GA, and MOEA are also used by different authors for comparisons but the overall results are mostly outperformed by NSGA-II and Simulated Annealing. In most of the cases NSGA-II acts as baseline algorithm for comparison. Figure 3 shows that multi-objective optimization techniques have been given more attention as

compared to single objective optimization techniques.

RQ2: What datasets are used in the context of NRP?

All the information about the datasets used in different studies is given in Table 8 with datasets, a number of studies that contains dataset and reference of the studies.

Table 8: Datasets

Dataset	No. of Studies	References of Studies
Synthetic	17	[4-12],[14],[18],[19],[21], [24],[25-28]]
Motorola	6	[12],[14],[18],[27-29]
Mozilla	5	[3],[11],[18],[19],[29]
Eclipse	7	[3],[11],[16],[18],[19],[27],[29]
Gnome	5	[3],[11],[16],[18],[29]
MS-Word	3	[17], [23],[29]
Theme Based RP Dataset	2	[22],[29]
ReleasePlanner	2	[17],[29]
Ralic	2	[18],[29]
Baab	1	[29]

The scenario can be more easily visualized by the statistics presented in Fig 4 which shows the results of the datasets used in studies in the form of percentages. It can be seen that in most of the studies synthetic datasets have been used with the overall percentage of 34 %. Eclipse got the second spot with 14 %. The third position is for Motorola with 12 %. Fourth place is acquired by Gnome and Mozilla with 10% each. Other datasets didn't get much attention because those are not available easily.

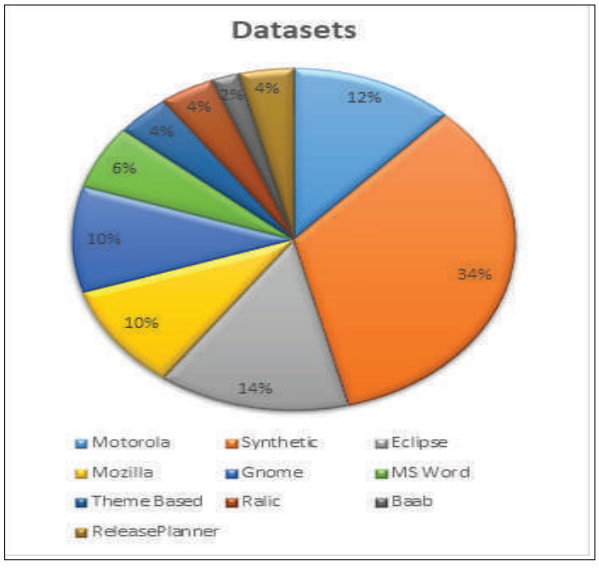

Figure 4: Usage graph of Datasets

RQ3: What is the overall research productivity of next release problem?

To answer this question, we have divided the number of studies into two phases according to years. In the first phase, we analyzed the trend during years 2010-2013 with respect to single and multi-objective techniques. The trend shows that work done on single objective optimization is significant during this time period as compared to multi-objective optimization which can be seen in Figure 5.

Figure 5 Research Trend During Years 2010-2018

The second phase consists of studies from years 2014-2018 and trend was analyzed again with respect to single and multi- objective optimization techniques proposed during this time period. Interestingly multi-objective optimizations techniques were more focused during this time span as compared to single objective optimization techniques which can be visualized through Figure 6

Figure 6 Research Trend During Years 2014-2018

To analyze the overall research productivity, we have considered the research papers that were published in the past years from 2010 to 2018, moreover, we have rated the studies that were able to qualify for the selected studies after quality assessment process and results will be presented. The number of studies according to publication year is shown in Fig 7.

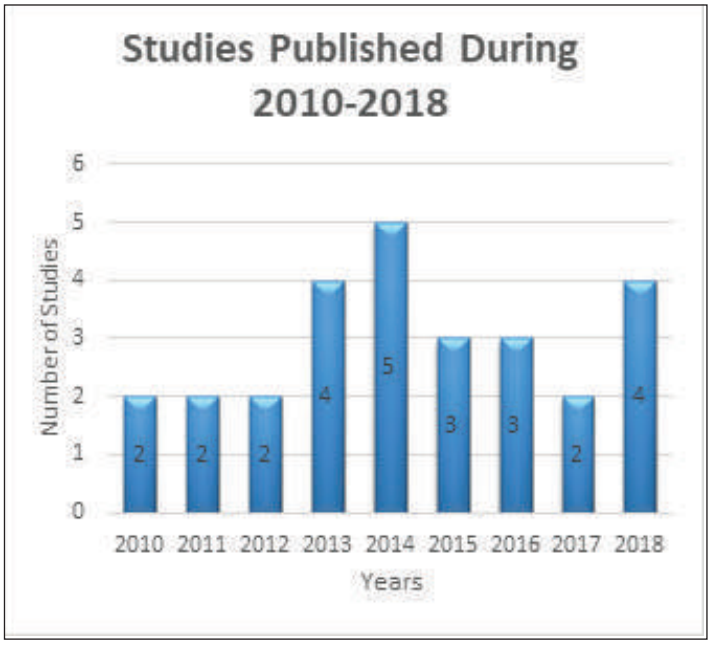

Figure 7: Studies Published During Years 2010-2018

Figure 6 shows that 2014 was the most active research year in which the highest number of studies were published. In 2017 the number was decreased to mere 2 studies but the graph rose again in 2018 with 4 studies published in a single year and more yet to come soon. These numbers are from the span of the last eight years and don't reflect the total number of studies published until yet in the field of NRP.

The quality levels of selected studies are shown in Table 10. The number of studies and the percentage that how many studies falls into a specific group are also shown.

Table 9: Quality Levels of Selected Studies

Quality Level	Number of Studies	Percentage (%)
Very High (score = 4)	12	44.44 %
High (score = 3.5)	8	29.62 %
Medium (score = 3)	7	25.92 %
Total	27	100.00 %

Table 9 reveals that more than 44 % of the studies lies in the highest quality span. While more than 29 % of studies managed to qualify for the high score and only 25.72 % are from medium quality according to our predefined quality assessment criteria.

4 Conclusion

This systematic literature review summarizes the existing literature published in the field of NRP. This paper primarily focused on reviewing the literature on NRP from the span of 2010 to 2018. This study classifies the literature according to different criteria like techniques, contributions, datasets, and quality. 44 studies were identified at the start from different databases and manual sources and after a series of different screening processes only 27 of those qualified for final full-text assessment. The quality of these 27 studies was then analyzed through predefined quality criteria.

The main findings of this research are as follows:

- Multi-Objective Optimization Techniques are the most widely used techniques.
- Among Multi-Objective Optimization Techniques, NSGA-II is widely popular because of its high convergence rate and good quality solutions. Similarly, for Single Objective Optimization, Simulated Annealing is used in most of the cases and showed promising results.
- Synthetic datasets are used in most of the previous studies because of the limited availability of real-world datasets.

NRP is still an emerging field and there is a need to apply the latest state of the art optimization techniques to solve it more efficiently. Furthermore, during the research, it was noted that most of the studies focused on certain common metrics like customer's satisfaction and development cost and other types of metrics like software maintainability, reliability, and traceability are given less attention. These areas should be focused on by researchers in the future.

References

[1] Pmi.org. (2017). 9th Global Project Management Survey. [online] Available at: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thoughtleadership/pulse/pulse-of-the-profession-2017.pdf [Accessed 10 Jan. 2019].

[2] Bagnall, A.J., Rayward-Smith, V.J. and Whittley, I.M., 2001. The next release problem. Information and software technology, 43(14), pp.883-890

[3] Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J. and Linkman, S., 2009. Systematic literature reviews in software engineering–a systematic literature review. Information and software technology, 51(1), pp.7-15.

[4] Jiang, H., Zhang, J., Xuan, J., Ren, Z. and Hu, Y., 2010, June. A hybrid ACO algorithm for the next release problem. In Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on (pp. 166-171). IEEE.

[5] Sureka, A., 2014, April. Requirements prioritization and next-release problem under Non-additive value conditions. In Software Engineering Conference (ASWEC), 2014 23rd Australian (pp. 120-123). IEEE..

[6] Cai, X. and Wei, O., 2013, June. A hybrid of decomposition and domination based evolutionary algorithm for multi-objective software next release problem. In Control and Automation (ICCA), 2013 10th IEEE International Conference on (pp. 412-417). IEEE.

[7] Aydemir, F.B., Dalpiaz, F., Brinkkemper, S., Giorgini, P. and Mylopoulos, J., 2018, August. The Next Release Problem Revisited: A New Avenue for Goal Models. In 2018 IEEE 26th International Requirements Engineering Conference (RE) (pp. 5-16). IEEE.

[8] Mauša, G., Grbac, T.G., Bašić, B.D. and Pavˇević, M.O., 2013, July. Hill Climbing and simulated annealing in large scale next release problem. In EUROCON, 2013 IEEE (pp. 452-459). IEEE.

[9] Xuan, J., Jiang, H., Ren, Z. and Luo, Z., 2012. Solving the large scale next release problem with a backbone-based multilevel algorithm. IEEE Transactions on Software Engineering, 38(5), pp.1195-1212.

[10] Li, L., 2016, September. Exact analysis for next release problem. In Requirements Engineering Conference (RE), 2016 IEEE 24th International (pp. 438-443). IEEE.

[11] Paixão, M. and Souza, J., 2013, July. A scenario-based robust model for the next release problem. In Proceedings of the 15th annual conference on Genetic and evolutionary computation (pp. 1469-1476). ACM.

[12] Li, L., Harman, M., Letier, E. and Zhang, Y., 2014, July. Robust next release problem: handling uncertainty during optimization. In Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (pp. 1247-1254). ACM.

[13] del Sagrado, J., ÁAguila, I.M. and Orellana, F.J., 2011, July. Requirements interaction in the next release problem. In Proceedings of the 13th annual conference companion on Genetic and evolutionary computation (pp. 241-242). ACM.

[14] Harman, M., Krinke, J., Medina-Bulo, I., Palomo-Lozano, F., Ren, J. and Yoo, S., 2014. Exact scalable sensitivity analysis for the next release problem. ACM Transactions on Software Engineering and Methodology (TOSEM), 23(2), p.19.

[15] Jiang, H., Xuan, J. and Ren, Z., 2010, July. Approximate backbone based multilevel algorithm for next release problem. In Proceedings of the 12th annual conference on Genetic and evolutionary computation (pp. 1333-1340). ACM.

[16] Cheng, X., Huang, Y., Cai, X. and Wei, O., 2014, July. An adaptive memetic agorithm based on multiobjecitve optimization for software next release problem. In Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation (pp. 185-186). ACM.

[17] do Nascimento Ferreira, T., Araújo, A.A., Neto, A.D.B. and de Souza, J.T., 2016. Incorporating user preferences in ant colony optimization for the next release problem. Applied Soft Computing, 49, pp.1283-1296.

[18] Veerapen, N., Ochoa, G., Harman, M. and Burke, E.K., 2015. An integer linear programming approach to the single and bi-objective next release problem. Information and Software Technology, 65, pp.1-13.

[19] Paixao, M. and Souza, J., 2015. A robust optimization approach to the next release problem in the presence of uncertainties. Journal of Systems and Software, 103, pp.281-295.

[20] Chaves-González, J.M. and Pérez-Toledano, M.A., 2015. Differential evolution with Pareto tournament for the multi-objective next release problem. Applied Mathematics and Computation, 252, pp.1-13.

[21] Almeida, J.C., Pereira, F.D.C., Reis, M.V. and Piva, B., 2018, April. The Next Release Problem: Complexity, Exact Algorithms and Computations. In International Symposium on Combinatorial Optimization (pp. 26-38). Springer, Cham.

[22] Araújo, A.A., Paixao, M., Yeltsin, I., Dantas, A. and Souza, J., 2017. An architecture based on interactive optimization and machine learning applied to the next release problem. Automated Software Engineering, 24(3), pp.623-671.

[23] Pitangueira, A.M., Tonella, P., Susi, A., Maciel, R.S.P. and Barros, M., 2017. Minimizing the stakeholder dissatisfaction risk in requirement selection for next release planning. Information and Software Technology, 87, pp.104-118.

[24] Fuchshuber, R. and de Oliveira Barros, M., 2014, August. Improving heuristics for the next release problem through landscape visualization. In International Symposium on Search Based Software Engineering (pp. 222-227). Springer, Cham.

[25] Brasil, M.M.A., da Silva, T.G.N., de Freitas, F.G., de Souza, J.T. and Cortés, M.I., 2011, June. A multiobjective optimization approach to the software release planning with undefined number of releases and interdependent requirements. In International Conference on Enterprise Information Systems (pp. 300-314). Springer, Berlin, Heidelberg.

[26] Del Águila, I.M. and Del Sagrado, J., 2016. Three steps multiobjective decision process for software release planning. Complexity, 21(S1), pp.250-262.

[27] Paixão, M.H.E. and de Souza, J.T., 2013, August. A recoverable robust approach for the next release problem. In International Symposium on Search Based Software Engineering (pp. 172-187). Springer, Berlin, Heidelberg.

[28] Durillo, J.J., Zhang, Y., Alba, E., Harman, M. and Nebro, A.J., 2011. A study of the bi-objective next release problem. Empirical Software Engineering, 16(1), pp.29-60.

[29] Zhang, Y., Harman, M., Ochoa, G., Ruhe, G. and Brinkkemper, S., 2018. An Empirical Study of Meta-and Hyper-Heuristic Search for Multi-Objective Release Planning. ACM Transactions on Software Engineering and Methodology (TOSEM), 27(1), p.3.

[30] Idri, A., azzahra Amazal, F. and Abran, A., 2015. Analogy-based software development effort estimation: A systematic mapping and review. Information and Software Technology, 58, pp.206-230.

[31] Geng, J., Ying, S., Jia, X., Zhang, T., Liu, X., Guo, L. and Xuan, J., 2018. Supporting Many-Objective Software Requirements Decision: An Exploratory Study on the Next Release Problem. IEEE Access, 6, pp.60547-60558..

Call for Papers/Authors Guideline

KIET Journal of Computing & Information Sciences (KJCIS) is biannual publication of College of Computing & Information Sciences, Karachi Institute of Economics and Technologies. It is published in January and July every year. We are lucky to have on board prominent and scholarly academicians as part of Advisory Committee and reviewers.

KJCIS is a multi-disciplinary journal covering viewpoints/ researches / opinions relevant to the non exhaustive list of the topics including data mining, big data, machine learning, artificial intelligence, mobile applications, computer networks, cryptography & information security, mobile and wireless communication, adhoc & body area networks, software engineering, speech & pattern recognition, evolutionary computation, semantic web & its application, data base technologies & its applications, Internet of Things (IoT), computer vision, distributed computing, grid and cloud computing.

The authors may submit manuscripts abiding to following rules:-

- Certify that the paper is original and is not under consideration for publication in any other journal. Please mention so, in case it has been submitted elsewhere.
- Adhere to normal rules of business or research writing. Font style be 12 points and the length of the paper can vary between 3000 to 5000 words.
- Illustrations/tables or figures should be numbered consecutively in Arabic numerals and should be inserted appropriately within the text.
- The title page of the manuscript should contain the Title, the Name(s), email address and institutional affiliation, an abstract of not more than 200 words should be included. A footnote on the same sheet should give a short profile of the author(s).
- Full reference and /or websites link, should be given in accordance with the APA citation style. These will be listed as separate section at the end of the paper in bibliographic style. References should not exceed 50.
- All manuscripts would be subjected to tests of plagiarism before being peer reviewed.
- All manuscripts go through double blind peer review process .
- Electronic submission would only be accepted at kjcis@pafkiet.edu.pk
- All successful authors will be remunerated adequately.
- The Journal does not have any article processing and publication charges.

Submission is voluntary and all contributors will find a respectable acknowledgment on their opinion and effort from our team of editors. Submission of a paper will be held to imply that it contains original unpublished work. In case the paper has been forwarded for publication

elsewhere, kindly apprise in time if the paper has been accepted elsewhere. Manuscripts may be submitted before September and May to get published in Jan & July issues respectively. We encourage you to submit your manuscripts at kjcis@pafkiet.du.pk

Editorial Board KJCIS
College of Computing & Information Sciences
Karachi Institute of Economics and Technology

Karachi Institute of Economics and Technology
Korangi Creek, Karachi-75190, Pakistan
Tel: (9221) 3509114-7, 34532182, 34543280 Fax: (92221) 35009118
Email: kjcis@pafkiet.du.pk
http://kjcis.pafkiet.edu.pk