Performance Evaluation of Word Embedding Algorithms

Shilpi Kulshretha
Department of CSE
Jagannath University, Jaipur, India

Lokesh Lodha
Department of Electronics. & Comm.
Jaipur National University, Jaipur, India

Abstract:- This study intends to explore the field of word embedding and thoroughly examine and contrast various word embedding algorithms. Words retain their semantic relationships and meaning when they are transformed into vectors using word embedding models. Numerous methods have been put forth, each with unique benefits and drawbacks. Making wise choices when using word embedding for NLP tasks requires an understanding of these methods and their relative efficacy. The study presents methodologies, potential uses of each technique and discussed advantages, disadvantages. The fundamental ideas and workings of well-known word embedding methods, such as Word2Vec, GloVe, FastText, contextual embedding ELMo, and BERT, are evaluated in this paper. The performance of these algorithms are evaluated for three datasets on the basis of words similarity and word analogy and finally results are compared.

Keywords:- *Embedding, Word2Vec, Global Vectors for Word Representation (GloVe), Embedding from Language Models (ELMo), BERT.*

I. INTRODUCTION

Word embedding is technique uses Vector Space Model concept to transform words into vectors in natural language processing. This technique offers a dense representation of words in a continuous vector space and now a days it has become a key technique in machine translation and NLP tasks. These techniques help machines comprehend and process human language more effectively by capturing contextual information and semantic relationships from words and sentences. The effectiveness of these algorithms depends on the selection of embedding technique for the application. Several NLP tasks, including named entity recognition, machine translation, sentiment analysis, and more, can be strongly impacted by the words.

Humans have always attempted to complete complicated tasks at the speed of light, thanks to the development of computers and computational capacity which made this possible. Word embedding provides a continuous and distributed representation that captures semantic and contextual information from sentences, in contrast to traditional approaches that represented words as discrete symbols or sparse representations. This is a supervised learning task that discards the classification accuracy by using the categories and the data vector as input. If the accuracy is sufficient, it can be applied to additional classification tasks; if not, more complexity is required to achieve optimal outcomes. This comparative study may help researchers and practitioners select the best word embedding strategy for their particular applications.

The research study is structured in five sections where first section introduces the NLP tasks and word embedding what about. Second section gives an overview of the basic ideas and precepts guiding word embedding. Section 3 discusses the popular word embedding techniques in detail, outlining their strengths and weaknesses. Section 4 discusses the evaluation framework and criteria used for comparing these techniques. Section 5 presents the experimental results and comparative analysis. Finally these methods are compared according to a range of criteria, including transferability, quality of embedding, computational efficiency, and adaptability to different languages or domains.

II. OVERVIEW ON WORD EMBEDDING

A. Vector Space Model (VSM)

The Vector Space Model forms the foundational concept for word Embedding. It represents words as vectors in a multi-dimensional space, where each dimension corresponds to a specific aspect or feature of the word. This representation allows mathematical operations and computations on words, enabling algorithms to understand relationships and similarities. Here we try to densely pack the information of the text into a vector which formally takes some hundred or thousand dimensions [1]. It had the first use case in the SMArt Information Retrieval System. The VSM has many use cases some of which are:

- Relevancy Ranking
- Information Retrieval
- Information Gathering

In word embedding a fundamental principle dictates that words appearing in comparable contexts tend to manifest proximity within the vector space representation. This signifies that their corresponding vectors exhibit similarity, emphasizing the preservation of contextual meaning and semantic relationships during the embedding process. For example, If the words "cat" and "dog" are frequently observed in the dataset within the context of "owner," the resulting word Embedding for "cat" and "dog" will demonstrate closeness in the vector representation. This proximity reflects their shared contextual relationship with "owner" and highlights the ability of word embedding to capture semantic associations based on the co-occurrence of words.

B. Semantic Meaning and Context

A vector space which contains the semantic and contextual information of word. These words that are semantically similar or contextually related are positioned closer to each other in this vector space. This characteristic simplifies the NLP task of capturing nuances and semantics essential for accurate interpretation. Semantic representation is an essential component of NLP, and it enables both humans and computers to better understand the meaning of language. For example division by zero in mathematics is erroneous. The mathematical rules of grammar do not allow us to divide anything by zero. This error is in the context of mathematics, and it is an example of semantic errors [2].

C. Word Sense Disambiguation

Word sense disambiguation is the task of identifying the correct meaning of a word in each context in the sentence. Sometimes, we are unable to distinguish what the person wants to say and we do not get the meaning. For example we have a sentence "Let's eat Grandma". It is evident to us that this sentence has two meanings. Grandmother is being eaten in the first meaning and we know that it is not something that anyone would want. In second case Grandma join you at the table to eat something. Because many words have multiple meanings and because a word's meaning can change depending on the context in which it is used, word sense discrimination (WSD) is a difficult task [3] because it depends on previous sentences. WSD can be used in many different ways and applications, such as rule-based, statistical, and machine learning approaches. Although WSD is a difficult and demanding task because it is being utilized in many NLP applications.

D. Named Entity Recognition

Named entity recognition is a subtask of NLP that extract and identify essential information from the text ant it a key function in NLP. It recognizes and categorize named entities, such as individuals, groups, places, and goods, and is the initial stage in deciphering the meaning of text. We can attain higher accuracy in other NLP tasks, like machine translation, text summarization, and question answering, if we can execute NER accurately and efficiently. To keep things simple, we take an example: "The man with the telescope is who I saw." Here, it's unclear that if I was the one with the telescope or if someone else was holding the telescope.

E. Part of Speech Tagging

The part of speech tagging is the process of giving each word in a text corpus and it is known as "POS Tagging." In corpus linguistics, POS tagging is the process of marking a word in a text as corresponding to a particular part of speech on the basis of definition and context. Noun, Pronoun, Adjective, Verb, Adverb, Preposition, Conjunction, and Interjection are the parts of speech. A text corpus contains assortment of textual data used for NLP model.

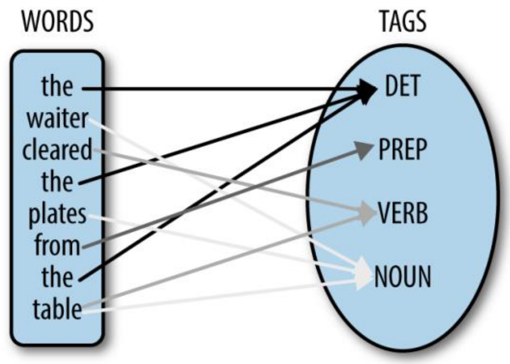

Fig. 1. POS Tagging for Words

F. Co-Reference Resolution:

Co-reference resolution is a difficult task because it necessitates a thorough comprehension of the text's meaning, including the connections between various entities. In the sentence "Mary went to the store and bought some apples," for instance. "She" refers to the subject "Mary" in the sentence "She ate them on the way home." To identify this coreference, the system must understand that "Mary" is the subject of the first sentence and the antecedent of the pronoun "she" in the second sentence.CR systems are typically trained on large corpora of text that have been manually annotated with coreference information. A CR system can be used to resolve coreferences in fresh text once it has been trained. In the sentence "Mary went to the store and bought some apples," for instance. "She" refers to the subject "Mary" in the sentence "She ate them on the way home". A machine would resolve this coreference by identifying all potential coreferences ("Mary" and "she") and using a variety of factors, such as number agreement, gender agreement, semantic similarity, and recency, to determine whether the two expressions actually refer to the same entity. Coreference resolution is a challenging task, but it is essential for many NLP tasks [4].

III. LITERATURE REVIEW

Saqqa and Samar et al [5] suggested approach Bengali language NLP researchers can quickly construct the necessary word embedding vectors for word representation in NLP. Jeffrey et al [6] performed a study on embedding and in this study they find that word2vec is efficient embedding technique which gives highest accuracy.

Several well-known word embeddings, including word2vec, Glove, and FastText, were investigated by Cagliero et al. [7] in a number of downstream tasks, such as sentiment analysis and text inference. According to their findings, starting the embedding layer at random can be trained to produce results that are comparable to starting it with pre-trained classic embeddings. Using cross-language datasets (English and Arabic), M. Fawzy et al. [8] examined latent semantic analysis on word2vec and GloVe in the topic segmentation (TS) task and they conducted a thorough analysis of the word2vec model and investigate its influence on TS using various training strategies and they concluded that when training algorithms are carefully selected based on the features of a language-specific dataset, word2vec performs well. P. Shah et al [9] performed a study and used datasets from multiple domains to investigate the effects of pre-trained word embedding. Their findings show that using pre-trained embedding as feature representations has a substantial effect on RC's performance and make the system easier.

IV. METHODOLOGY

This study performs a comparative analysis on popular word embedding techniques. This study includes the popular word embedding techniques like Word2Vec, GloVe, fastText, ELMo and BERT. The examination process and brief discussion of each algorithms is given below.

- *Word2Vec:* Word2vec is a group of related models that are used to produce word embedding and these model used shallow neural networks that are trained to reconstruct linguistic contexts of words. This approach uses continuously sliding Skip-gram or continuously sliding Bag-of-Words (CBOW) which are two well-known techniques for creating datasets. Integration of this Word2Vec are used NLP and deep learning libraries is proof of its widespread adoption. The Word2Vec converts words into dense vectors, facilitating the capture of complex semantic relationships, and has been crucial in the advancement of numerous NLP applications.

- *C-BOW Model* – The CBOW model works on the basis of surrounding words concepts. It takes word as input and trying to predict the target word in the center of the window. Predicting the central word in each corpus is the task of the CBOW. In the CBOW model, the distributed representations of word are combined to predict the word in the middle and to be more precise, given the words that come before and after the target word, the goal is to predict and identify the word that is in the middle of this context and Skip gram model predicts the context.

- Skip-gram Model- This method has many uses in NLP and has proven effective in capturing complex semantic relationships between words. This focuses on context word prediction given a target word. By increasing the likelihood of adjacent words, it seeks to obtain a thorough grasp of the context around the given word. For example given the word is "jumped" in the sentence, we ought to be able to guess the other words, such as "the," "cat," "over," and "puddle" in the sentence. During the training phase of the neural network, this is the essential for the construction of vectors. Collection of texts is the first step to create a list of distinct words and each of which is given a unique index, in order to construct a dataset for training. In this scenario, each word is unique, corresponding to indices such as [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

The sliding window is used to make training simpler. The words inside the window serve as the input for a given window size, and the target word is located in the center of the sentence. For example, the dataset could look like this with a window size of 4: [[1, 2, 4, 5], 3]. In this case center word text is 3 and the final calculation is compared to the center word index, and the neural network's input is [1, 2, 3, 4]. For mapping, a loss function is used to aid in training and direct the neural network towards the acquisition of meaningful representations and a logarithmic function is a good choice.

- *GloVe:* This word embedding technique uses a novel method to capture word semantics and creates word vectors by utilizing global word co-occurrence statistics, highlighting word analogy and similarity connections. GloVe is well-known for its efficiency and scalability and it has become more and more popular in NLP and provides rich, context-aware word representations. GloVe support a wide range of applications, including sentiment analysis and machine translation and many others. This is unsupervised approach in the field of word embedding and NLP.

- *Word Commonality and Semantic Relations:* GloVe word embedding provides a nuanced understanding that is beneficial for tasks like NER and word similarity measurements. When it examines semantic relationships and word commonality, it performs exceptionally very well.
- *Global Corpus Statistics:* GloVe and Word2Vec integrates global corpus statistics into its model architecture. This integration enhances accuracy and model performance, particularly in limited size of word corpus when training is performed. The ability of this method is to leverage these global statistics contributes to a more robust word embedding model.

- *FastText:* This competitive algorithm was created by Facebook Research in 2016 and it uses the concept of sliding window for creating a training dataset. One of the excelling libraries for deep learning was also created by this group. This algorithm was developed mostly for text classification, but it has found its use in word embedding too. In the other methods, a word was being used as a fundamental quantity for text processing, but the FastText method uses characters as a base for its purpose. Since a bunch of characters can itself make a big dataset, therefore less training is needed for FastText to work.

- *Example Sentence: "The performance of the model was exceptional"*
- *Sub word Segmentation:* FastText segments each word into sub words, commonly using bi-grams and trigrams. For the given sentence, the sub word segmentation would include sub words such as <Th, he, e , p, er, fo, or, rf, , rm, , nc, ce, of, ma, th, he, m, de, el, l, wa, mo, od, as, s, ce, ep, pt, ti, ex, xc, io, na, al, on >.

- *Sub word Vector Representation:* Each sub word is associated with a pre-trained vector representation, obtained through an unsupervised learning process. These vectors capture the semantic and morphological information of the sub words.

ISSN No:-2456-2165

- *Word Vector Computation:* The word vector for a particular word is computed as the sum of its sub word vectors. For example, the vector representation of the word "model" would be the sum of its subword vectors, where each subword is associated with a pre-trained vector. For example, <mo> + <od> + <de> + <el>.

This indicates that the word vector for the word "model" is the total of all of its bigrams. Every term in the lexicon is depicted as a single-hot encoded vector. The size of the vocabulary is represented by this binary vector, where all other values are set to 0 and only the index corresponding to the current word is set to 1. Text can also be generated with FastText and it is an effective tool that can be applied to a range of NLP tasks. This is easy to use and delivers cutting-edge outcomes for a variety of tasks and it generates a word sequence as output in order to accomplish a word sequence. The output sequence of the model is generated by forecasting the subsequent word in the sequence using the words.

- *BERT:* This uses masked language model which is a pre-trained model to predict the masked words based on the surrounding words' context, and random words in the input sentences are masked. Now days it has seen a rise in interest in pre-trained language models (PLMs) as a way to improve natural language processing related tasks. PLMs learn to represent the meaning of words and phrases by considering their context from both sides. One of the most popular PLMs is called BERT (Bidirectional encoder representation from transformers), which has demonstrated state-of-the-art performance on a range of natural language processing tasks, such as text summarization, sentiment analysis, and question answering. By taking into account both left and right context, this bidirectional approach differs from previous models and enables this algorithms to capture a more thorough understanding of word relationships.

- *GPT:* Generative Pre-trained Transformer uses self-attention mechanisms while generating predictions, the attention mechanism enables the model to concentrate on distinct segments of the input sequence. GPT is a key component of contemporary NLP and it is a transformer-based architecture that works on the pre-training and fine-tuning principles. The most recent model in the GPT family uses three orders of magnitude more parameters. These types' transformers have proven to be an extremely effective tools for producing text or sentences like a human being.

A fundamental component of GPT is a transformer neural network which uses attention mechanisms. The architecture of the model consists of feed-forward neural networks, positional encodings, and several layers of attention. Positional encoding is added to the input embedding to provide information about the positions of tokens in the sequence, as transformers do not understand the sequential order of input tokens by default.

V. PERFORMANCE EVALUATION

There are numerous word embedding methods available and it is crucial to select a set of evaluation criteria that are pertinent to the task or tasks for which word embedding will be used in order to compare various word embedding techniques. Every algorithm or technique has its advantages and disadvantages. To select a benchmark dataset that adequately represent the tasks for which the term "Embedding" will be used, is also crucial. Two parameters semantic similarity and syntactic similarity are frequently used evaluation criterion that gauges how well a embedding captures the meaning or relatedness of words. Both requirements semantic and syntactic frequently call for figuring out similarities or distances within the embedding space.

Evaluation of Word Similarity: Calculate the degree of similarity between words embedding in word pairs and contrast the results with scores of human annotations. Pearson or Spearman correlation are two common measures of similarity. Extrinsic evaluation is just as important as intrinsic evaluation and it consists the term "Embedding" into tasks related to downstream natural language processing. Computational efficiency, scalability, and adaptability of word embedding across domains should be evaluated and how different embedding techniques handle out-of-vocabulary words, rare words, and multilingual contexts is important.

TABLE 1. PRE-TRAINED MODELS USED FOR OUR TESTS

Serial No.	Name	Training Corpus	Approximate size
1	Googlenews-Vectors-Negative300.bin	Google News Corpus	1.5GB
2	glove.42B.300d	Common Crawl(Websites)	4.5GB
3	wiki-news-300d-1M-subword.vec	Wikipedia	2GB

The insights gained from these evaluations, can guide researchers to select the most suitable embedding technique for their specific NLP tasks. Choosing the appropriate evaluation criteria and benchmark datasets plays an important role in assessing and comparing various word embedding techniques. Word Similarity test, Quality of classification, and word analogy test are chosen for analysis.

TABLE 2. DATASETS USED TO CHECK EFFICIENCY ON WORD SIMILARITY

Name	Entries	Year
WordSim-353	353	2002
WordSim-353-SIM	203	2009
WordSim-353-REL	253	2009
Miller-Charles (MC-30) Dataset	30	1991
Rubenstein & Goodenough	65	1965

(RG-65)		
SimVerb-999	999	2014
SimLex-999	999	2014

For performing word analogy test, the Google dataset and the MSR dataset are used. These dataset evaluates the ability of word embedding and to capture the semantic and syntactic relationships between words. The Google dataset contains 19,544 number of questions which can be divided into groups one is "morpho-syntactic" and other one is "semantic". There are total 8,000 analogy issues in the other MSR dataset.

VI. RESULTS AND DISCUSSION

Table 3, 4, 5, and 6 show the results of the study and these show that Word2Vec performed faster and more accurately than GloVe and FastText on every dataset. The reason is that Word2Vec could accurately capture the semantic relationships between words. It gives highest word similarity 81.3 for RG-65 dataset and gives highest word analogy 74.4.

Word2Vec, GloVe, and FastText are the three word embedding methods we employed in our experiment. Words are learned to be represented as vectors of real numbers using neural network-based models, which underpin all three word embedding techniques. Following that, these vectors can be applied to a range of natural language processing tasks, including machine translation, sentiment analysis, and text classification. These are all highly well-liked methods in NLP that are applied to some fascinating tasks such as: machine translation, similarity detection, analogy detection, named entity recognition

TABLE 3. COMPARISON OF WORD SIMILARITY

Name	Word2vec	GloVe	fastText
WS-353	64.3	59.7	64.3
WS-353-REL	53.4	55.9	56.4
WS-353-SIM	74	66.8	72.1
MC-30	74.7	74.2	76.3
RG-65	81.3	75.1	77.3
SimVerb-999	24.5	17.2	21.9
SimLex-999	37.2	32.4	35.2

TABLE 4. COMPARISON OF WORD ANALOGY

Name	Word2vec	GloVe	fastText
Google (Add)	70.7	68.4	40.5
Google (Mul)	70.8	68.7	45.1
Semantic (Add)	74.4	76.1	19.1
Semantic (Mul)	74.1	75.9	24.8
Syntactic (Add)	67.6	61.9	58.3
Syntactic (Mul)	68.1	62.7	61.9
MSR (Add)	56.2	50.3	48.6
MSR (Add)	56.8	51.6	52.2

Categorization of words into different clusters in a machine learning task because of semantic similarity. Three datasets were used for this purpose namely AP dataset and BLESS dataset.

TABLE 5. COMPARISON OF CONCEPT CATEGORIZATION

Name	Categories	Word2vec	GloVe	fastText
AP Dataset	21	65.7	61.4	59.0
BLESS Dataset	56	74.0	82.0	73.0
BM Dataset	27	45.1	43.6	41.9

The final evaluation criteria we chose was the Outlier Detection criteria. We adopted two datasets for outlier detection: WordSim-500 and 8-8-8 datasets.

Each of the 500 clusters in the WordSim-500 is represented by a set of eight words with five to seven outliers. Eight clusters, each consisting of a set of eight words with eight outliers, make up the 8-8-8 dataset. We computed the Outlier Position Percentage (OPP) in addition to accuracy. Between the two datasets, the results, displayed in Table V, were inconsistent. On the WordSim-500 dataset, for instance, GloVe performed the best, but on the 8-8-8 dataset, it had the lowest accuracy.

TABLE 6. COMPARISON OF OUTLIER DETECTION

Name	Word2vec	GloVe	fastText
WS-500 (Accuracy)	14.02	15.09	10.68
WS-500 (OPP)	85.33	85.74	82.16
8 – 8 – 8 (Accuracy)	56.25	50.0	57.81
8 – 8 – 8 (OPP	84.38	84.77	84.38

This study does not cover advanced topics like machine translation because those require further training on our part. In the machine translation, to understand the meaning of a text in one language and converting it into another language is difficult task. It is necessary to take textual meaning of both languages same and machine translation can perform this up to some limitation. A robust machine translation systems can be created that are more accurate and effective than ever before by utilizing the most recent developments.

VII. CONCLUSION

In this study, the performance of various word embedding techniques was evaluated and compared. The findings of this study gives insightful information about how well various word embedding methods perform in tasks involving natural language processing. According to the result Word2Vec word embedding technique performs well compared to the other techniques. This technique could be used to create machine translation or text classification systems that are more precise and effective. This study also discovered that using a larger dataset to train the models enhanced the performance of all three word embedding techniques. Pre-trained model glove42B has 4GB size. This study used three word embedding strategies with some limitations. This study is limited to the two NLP tasks and it would be intriguing to compare how well these methods perform with other word embedding strategies. Overall this study presents the good insights and can guide practitioners to select a good word embedding technique for their applications.

REFERENCES

[1]. D. Suleiman and A. Awajan, "Comparative Study of Word Embeddings Models and Their Usage in Arabic Language Applications," *2018 International Arab Conference on Information Technology (ACIT)*, Werdanye, Lebanon, 2018, pp. 1-7, doi: 10.1109/ACIT.2018.8672674.

[2]. Wadud, Md. Anwar Hussen et al. "Word Embedding Methods for Word Representation in Deep Learning for Natural Language Processing." *Iraqi Journal of Science* (2022): n. pag.

[3]. H. Sanjurjo-González, "Improving accuracy of an existing semantic word labelling tool using word embeddings," *2021 16th Iberian Conference on Information Systems and Technologies (CISTI)*, Chaves, Portugal, 2021, pp. 1-5, doi: 10.23919/CISTI52073.2021.9476323.

[4]. M. N. Helaskar and S. S. Sonawane, "Text Classification Using Word Embeddings," *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, Pune, India, 2019, pp. 1-4, doi: 10.1109/ICCUBEA47591.2019.9129565.

[5]. Al-Saqqa, Samar and Arafat A. Awajan. "The Use of Word2vec Model in Sentiment Analysis: A Survey." *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control* (2019): n. pag.

[6]. Pennington, Jeffrey et al. "GloVe: Global Vectors for Word Representation." *Conference on Empirical Methods in Natural Language Processing* (2014).

[7]. L. Cagliero and M. L. Quatra, "Inferring Multilingual Domain-Specific Word Embeddings From Large Document Corpora," in *IEEE Access*, vol. 9, pp. 137309-137321, 2021, doi: 10.1109/ACCESS.2021.3118093.

[8]. M. Fawzy, M. W. Fakhr and M. A. Rizka, "Word Embeddings and Neural Network Architectures for Arabic Sentiment Analysis," *2020 16th International Computer Engineering Conference (ICENCO)*, Cairo, Egypt, 2020, pp. 92-96, doi: 10.1109/ICENCO49778.2020.9357377.

[9]. P. Shah, S. Shah and S. Joshi, "A Study of Various Word Embeddings in Deep Learning," *2022 3rd International Conference for Emerging Technology (INCET)*, Belgaum, India, 2022, pp. 1-5, doi: 10.1109/INCET54531.2022.9824963.

[10]. Y. Singh, M. Saini and Savita, "Impact and Performance Analysis of Various Activation Functions for Classification Problems," 2023 IEEE International Conference on Contemporary Computing and Communications (InC4), Bangalore, India, 2023, pp. 1-7, doi: 10.1109/InC457730.2023.10263129.

[11]. Vispute, S., Saini, M.L. (2022). Automation in Agriculture: A Systematic Survey of Research Activities in Agriculture Decision Support Systems Using Machine Learning. In: Singh, P.K., Wierzchoń, S.T., Chhabra, J.K., Tanwar, S. (eds) Futuristic Trends in Networks and Computing Technologies. Lecture Notes in Electrical Engineering, vol 936. Springer, Singapore. https://doi.org/10.1007/978-981-19-5037-7_56

[12]. M. L. Saini, V. K. Sharma and A. Kumar, "An Efficient Single and Double Error Correcting Block Codes with Low Redundancy for Digital Communications," 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), Gharuan, India, 2023, pp. 827-831, doi: 10.1109/InCACCT57535.2023.10141727.

[13]. Vispute, S.; Saini, M.L. Performance Analysis of Soil Health Classifiers Using Data Analytics Tools and Techniques for Best Model and Tool Selection. Int. J. Online Biomed. Eng. 2022, 18, 169–189.

[14]. Shailaja Pede and Madan Lal Saini, "A Brief Bibliometric Analysis and Visualisation of Scopus and WoS databases on Blockchain Technology in Healthcare Domain", Library Philosophy and Practice (e-journal), vol. 5106, 2021.

[15]. M. Gehlot and M. L. Saini, "Analysis of Different CNN Architectures for Tomato Leaf Disease Classification," 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 2020, pp. 1-6, doi: 10.1109/ICRAIE51050.2020.9358279.

[16]. Varun Sapra, Madan Lal Saini, "Deep learning network for identification of Ischemia using clinical data", International Journal of Engineering and Advanced Technology, ISSN: 2249-8958, Volume-8 Issue-5, June 2019.

[17]. Varun Sapra, Madan Lal Saini, "Computational Intelligence for Detection of Coronary Artery Disease with Optimized Features", International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075 Volume 8, Issue-6C, Pages 144-148, April 2019.

[18]. S. Kulshrestha and M. L. Saini, "Study for the Prediction of E-Commerce Business Market Growth using Machine Learning Algorithm," *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, Jaipur, India, 2020, pp. 1-6, doi: 10.1109/ICRAIE51050.2020.9358275

[19]. Sapra Varun , Saini M.L and Verma Luxmi, Identification of Coronary Artery Disease using Artificial Neural Network and Case-Based Reasoning, Recent Advances in Computer Science and Communications 2021; 14(8) .

ISSN No:-2456-2165

https://dx.doi.org/10.2174/266625581399920061322540 4

[20]. M. L. Saini, J. Panduro-Ramirez, J. Padilla-Caballero, A. Saxena, M. Tiwari and K. Ravi, "A Study on the Potential Role of Blockchain in Future Wireless Mobile Networks," 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2023, pp. 547-550, doi: 10.1109/ICACITE57410.2023.10182375.

[21]. Kulshrestha, Shilpi and Lal Saini, Madan, The Analytical Study of an E-Business Model for Establishing the Concept of Customer Retention (A Case Study of Myntra.com) (August 29, 2019). Proceedings of International Conference on Advancements in Computing & Management (ICACM) 2019, http://dx.doi.org/10.2139/ssrn.3444797

[22]. Kavita Lal, Madan Lal Saini; A study on deep fake identification techniques using deep learning. AIP Conf. Proc. 15 June 2023; 2782 (1): 020155. https://doi.org/10.1063/5.0154828