ABSTRACT The citation intent extraction and classification has long been studied as it is a good measure of relevancy. Different approaches have classified the citations into different classes; including weak and strong, positive and negative, important and unimportant. Others have gone further from binary classification to multi-classes, including extension, use, background, or comparison. Researchers have utilized various elements of the information, including both meta and contents of the paper. The actual context of any referred article lies within the citation context where a paper is referred. Various attempts have been made to study the citation context to capture the citation intent, but very few have encoded the words to their contextual representations. For automated classification, we need to train deep learning models, which take the citation context as input and provides the reason for citing a paper. Deep neural models work on numeric data, and therefore, we must convert the text information to its numeric representation. Natural languages are much complex than computer languages. Computer languages have a pre-defined fixed syntax where each word has a unique meaning. In contrast, every word in natural language may have a different meaning and may well be understood by understanding the position, previous discussion, and neighboring words. The extra information provides the context of a word within a sentence. We have, therefore, used contextual word representation, which is trained through deep neural networks. Deep models require massive data for generalizing the model, however, the existing state-of-the-art datasets don’t provide much information for the training models to get generalized. Therefore, we have developed our own scholarly dataset, Citation Context Dataset with Intent (C2D-I), an extension of the C2D dataset. We used a transformers based model for capturing the contextual representation of words. Our proposed method outperformed the existing benchmark methods with F1 score of 89%.

INDEX TERMS Citation intent classification, citation reason, citation context, transformers model, research paper similarity.
to work under discussion [2]. In order to fully understand the nature of a citation we must study the full contents of a publication. Of particular interest, out of the full content of an article, is the text where in-text citation has been made called the citation context.

Different authors have categorized the citation reason, also called citation intent or citation function, into various categories. [3] have categorized the citations into three classes with a positive, weak, or neutral relationship with the citing paper. Jurgens et al. [4] have claimed that the citation may be for six different reasons, and the strength of relevancy of these categories are different from each other. The earlier citation intent classification methods were based on non-textual information including bibliographic information [5], metadata and profiling of the authors [6]. With the advancement in natural language processing techniques full contents exploration has become more robust and effective in summarization, and text classification. Therefore, recent approaches are based on full text information utilizing the actual contents of a paper for exploring why a paper is cited in a citing paper [7]–[12].

In this paper, we present a similar technique which is based on the full content a research article in order to investigate the citation reasons. We can thus investigate the maps of science by reading the edge labels of a citation network and see how a portion of a citation network is connected to the rest of the network. We have used advanced context based language models for converting the text data and classifying the text to a particular class of intent based on the context in which a paper is being cited. We have used the BERT transformer for the contextual representation using attention model. The model is trained on a large number of citation context achieving F1 score of 89 percent as compared to previously claimed benchmark 79, and 84 percent by Jurgens et al. [4] and Cohan et al. [13], respectively. This study will have its applications in citation recommender systems, domain-specific expert searching, digital libraries, and academic search engines, building academic-social networks, and other research-oriented information systems. The citation information system, strengthened with the citation reasons, will help authors avoid inappropriate references in a paper and suggest the relevant one to the authors.

The rest of the paper is organized as follows: in Section II, we introduce existing citation intent classification methods. Section III discusses our proposed study framework. The details of each of the steps are further discussed in the subsections of section III. Section IV evaluates the classification models and compares the results. Finally, we conclude our study in section V.

II. RELATED WORK

Protracted literature exists for studying the science of relating articles to recommend, summarize, categorize, and extract common interests of authors. In this section we first discuss the general types in which we can categorize the methods of finding paper relevance and intent classification. Afterwards, we highlight the recent work and explore their strengths and limitations.

A. CATEGORIES OF METHODS

We have categorized the existing techniques into various categories based on four types of information used by these techniques: bibliographic, metadata, collaborative, and contents of the authors and research articles. Each group follows a general methodology and typically has some common steps of procedures that they go through. We discuss the general methods of each of these categories and their benefits and limitations in detail.

1) BIBLIOGRAPHIC INFORMATION BASED METHODS

Bibiliographic information is based on the references within an article. Citation is a systematic method of referring to a scientific article to support, negate, or associate a claim, knowledge, or methodology. A network of citations is formed as a result of mutually referring publications for various reasons. This network of citations has been studied and investigated by researchers to induce a relationship among the documents. It plays a significant role in calculating the impact of journals, institutions, and authors, including Impact Factor [14] and H-Index [15] to name a few. Many state-of-the-art methods have been proposed based on bibliographic information to predict the relationship among research articles. They normally share the assumption that if an author1 cites author2 or a paper1 cites paper2, they are relevant, irrespective of the citation purpose, importance, sentiment, topic, or reason. Raja et al. [5] have used bibliographic information for finding relevant articles of a queried paper. F. Xia et al [16] uses bibliographic information to extract network structure of citations. Haifeng Liu et al. [6] uses the associative mining technique to obtain a paper representation of the citing paper from the citation context.

The bibliographic-based approaches mostly use one of the two concepts: bibliographic coupling and co-citation coupling. M.M. Kessler introduced bibliographic coupling in 1963 [17]. Bibliographic coupling occurs when two papers refer to a common paper, then the two citing papers are said to be bibliographically coupled with each other and considered a strong relationship between these two papers. The strength of such coupled papers becomes stronger with an increased number of references they have in common. The concept is explained in Figure 1 where the article A and B are bibliographically coupled as they have referenced three papers; C, D, and E together.

S. Henry introduced the concept of co-citation coupling in 1973 [18] and claimed it as a better similarity indicator of research documents. Co-citation coupling appears when two papers list together in the list of referenced articles, although they do not cite each other directly. The strength of the relationship becomes more robust when their co-citation increases. Thus the frequency of co-cited is defined by the frequency with which they appear together in the reference list of some other paper. In Figure 2 article A and B are
co-cited, having frequency 3, as they both are cited in papers C, D, and E.

Although bibliographic information-based approaches are most widely used these techniques have some limitations along with their strengths. Below are some of the strengths and limitations of bibliographic information-based techniques:

- **Strengths:**
  - Reference is obligatory to scientific articles; therefore, it works for all the research documents, uniformly
  - Bibliography information is openly available in case of both open and close journals; therefore, the availability of this information is guaranteed
  - Metrics like impact factor, H-Index are already using it and are considered very stable and reliable

- **Limitations:**
  - Documents other than scientific articles may not have this information as they may not cite documents within them.
  - The reason for citation and thus nature of relativity of documents may not be measured by using the only bibliographic information

- If papers from a different domain are cited in a research article, they will be considered relative to each other, irrespective of the reason for citation

2) **METADATA BASED METHODS**

It is not all in the data; some information lies in the extra details gathered alongside the information being contributed, as depicted in Figure 3. Metadata is defined as data about data. In the context of research publications, metadata will mean the data about published articles, including title, list of authors, keywords, publication venue, publication date, journal category, the topic of a research article, volume, issue, and page numbers as shown in Figure 3. All such information which is not essentially the part of publication contents, form metadata. Metadata is also freely available, irrespective of the scope of the venue. Apart from research articles, Web 2.0 has made metadata of more interest, motivating authors of web documents to entertain extra information about the contents on the web. The exposure of meta contents has made the system communicate intelligently and seamlessly to extract data semantics. Scientific communities have also built systems on top of metadata, although relatively less focused than web documents. Almost all the indexing services use metadata information for some necessary information mining. Afzal et al. [19] used topic information, authors, and publication date for searching the important, relevant research articles. They developed a recommender system that recommends articles published by the same group of authors having topic information. The system fails to provide recommendations in case of a lack of topic information. Sajid et al [20] have proposed finding the topic information based on the references within a paper.

We have summarised the advantages and weaknesses of metadata-based techniques below:

- **Strengths:**
  - A relatively more straightforward approach to implement
  - The information can simply be used as a feature list, as most of the metadata are common in all datasets
  - Suitable for statistical-based approaches. The information usually is readily available, and fewer efforts are required to extract such information

- **Limitations:**
  - The pieces of information are not comprehensive and do not exhibit many details about the article contents
  - The similarity of documents is judged based on keywords, venue, topic, and other similar information.

3) **COLLABORATIVE FILTERING (CF) BASED APPROACHES**

The user profiling provides comprehensive information about the interests, domain, likes, and dislikes of that user. Collaborative filtering process information from the collaboration of community members based on their history and interests, as described in Figure 4. It is performed by monitoring
the users’ profile and their activities. The user profile and activities exhibit the user interest group and their behavior. User profiles are managed by analysing their ratings and click streams, co-downloads, and other types of similar information. Researchers used collaborative filtering-based approaches to advocate the relationship among authors, journals, topics, and research articles. Khalid et al. [21] have proposed a collaborative filtering-based approach. These approaches generally have data sparsity, reduced coverage, synonymy, privacy, shilling, grey and black sheep issues, as explained by Su et al. [22].

- **Strengths:**
  - The nature of this technique is not specific to scientific articles but remains valid in any electronic form of information. Therefore, this method may be used for any kind of application.
  - Devastating user profiling and data sharing techniques have made its mode progressive and feasible. Therefore, we see newly emerged recommender systems built on top of collaborative filtering approaches [23], [24].

- **Limitations:**
  - In the case of research articles, authors do not share their interests as openly as in the case of other information; for example, users thoughts on a tweet or a post in a social networking application.
  - The system may not have enough information in case of new authors, a cold start problem. Therefore, without having enough information, the system based solely on collaborative filtering approaches will fail.
These approaches are firmly based on user input which may be misleading and may not be exhibiting the interests of an author.

- The recommendation of an article can be made by this method, but it is not possible to determine the article relationship using this techniques.

4) CONTENT-BASED APPROACHES
The real meaning and nature of a publication may not be clearly explained by just a piece of information like metadata, author interests, or how they have cited each other. For a clear idea of the relationship among the documents, an information system must have the semantics of a paper’s full contents. Content-based approaches utilise the text of a document, which is the real information belonging to an article. The similarity of publications is measured by the similarity between the contents of those two papers [2]. These techniques require the contents only and do not rely on any other measures, for instance, the references or metadata availability; therefore, these approaches can be generally applied to any kind of documents, including research articles, web pages, or news articles. However, understanding the full-text of a paper may be straightforward for humans, but computers are still combating to grasp the semantics of a text. Efforts have been made to make computers understand paper contents. We can mainly categories these techniques into two major categories. The statistical methods and contextualised approaches, discussed in detail below:

5) NON-CONTEXTUALIZED/STATISTICAL METHODS
These approaches have tried to predict the similarity of documents based on the statistical grounds of the text. Text filtration is applied to the content to extract key-terms, calculate term frequency, and find root words to understand the similarity between two documents. The text selected for this purpose may not necessarily be the full text of the paper as extensive text processing may become difficult. Therefore, different approaches have selected a different content types, including the abstract, title, a summary of a paper, citing paragraph, or the citation context. The text filtration process typically goes through the following steps:

a: TEXT PreProcessing
Pre-processing is one of the critical steps in text classification tasks. Uysal et al. [25] discussed four common text classification steps; stop word removal, tokenisation, case conversion, and stemming/lemmatisation.

1) Tokenization: In the tokenization stage, a sentence is broken into recognizable groups of characters that are validated from a dictionary, for example, WordPiece [26]. The text is typically converted to a lower case before tokenizing them. Certain characters are removed, including punctuation marks and special characters. A token can minimum contain a single character in case it is not a recognized word in the dictionary. The complete process is explained in section II-A5.

2) Filtering: In filtration step all those words which do not contribute to the meaning or context of words in a sentence are removed. The most common approach is the stop words removal. Stop words are the words that frequently occur in text irrelevant to the topic. Stop words frequently occurs in text without having much information. These include prepositions, conjunctions, articles. Stop word removal is a language-specific task requiring knowledge of a particular language, which is English in our case. We have used Natural Language Toolkit (NLTK) [27] for removing stop words in Python. It has a massive list of stop words already defined in 16 different languages. We have also extended a list of stop words provided by NLTK by adding some numbers and special characters that do not change the meaning of sentences.

3) Stemming: It is also language-dependent task which tries to obtain the root of derived words. It reduces the words by removing last few characters from them. The new word may not be a meaningful correct word. Stemming has remarkable effects on word embedding, as studied by Kantrowitz et al. [28].

4) Lemmatization: Stemming and lemmatization both extracts the root words by considering morphological analysis of the words. The only difference between the two is that stemmed words are not necessarily proper dictionary words whereas, the lemmatization maps words to their valid truncated form. It is also a language specific task, and is performed using various methods discussed by Peter et al. [29].

b: MEASURING THE TEXT SIMILARITY
After the applying the preprocessing steps on text, tokens are extracted as a result. The tokens similarity is computed by using various types of distance measuring techniques. The statistical methods of similarity makes a study to relate a number of data units. The measures further acts as building blocks for various clustering and classification algorithms. Some of the text distance measuring techniques are listed below:

1) Manhattan or City Block Distance Measure: It is the simplest method of calculating the distance between two points. It is called City Block distance as it resembles the blocks in streets, and the distance is measured by traveling in those blocks of the street. It calculates the distance between two points not by going straight but first horizontal and then vertical, as depicted in Figure 5.

For example, having two points; \( p1(x_1, y_1) \) and \( p2(x_2, y_2) \) the distance by looking at the difference between the \( x \) and \( y \) coordinates of the points. The distance is calculated by using the following expression:

\[
distance = |x_2 - x_1| + |y_2 - y_1|
\]
2) **Euclidean Measure:** Euclidean distance takes the advantage of measuring the direct difference between the points. The length of line between two points is given by the hypotenous of a right angle triangle, given in Figure 6. The Pythagorean theorem is used to calculate this length, by using the following:

\[
distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}
\]

3) **Cosine Similarity Measure:** Cosine similarity is most widely used for computing the distance between two vector representation of words. This measure is very different than the previous measures. It is more concerned with the orientation of two points in space rather than the direct distance between them, as shown in Figure 7. It calculates the distance by using the following equation, considering our previous example of \( p_1 \) and \( p_2 \):

\[
cos(\theta) = \frac{(x_1x_2 + y_1y_2)}{\sqrt{x_1^2 + y_1^2} \sqrt{x_2^2 + y_2^2}}
\]

The value of cosine ranges from 0 to 1. When the values is 1 it means the vectors are similar to each other. In case of 0 the vectors are not similar at all. The values may lie in between 0 and 1, which means the vectors are somehow similar.

4) **Jaccard Similarity:** It is also called Jaccard similarity coefficient. It compares the elements of a set and extracts the matching and not matching members. It computes the similarity of two sets of data. The value ranges from 0 to 100%. The similarity of the documents is directly proportional to the value of Jaccard similarity percentage, calculated as:

\[
J(A, B) = \frac{|A \cap B|}{|A \cup B|}
\]

6) **SEMANTIC BASED APPROACHES**

In the previous section, we discussed the general method of techniques based on measuring the word level similarity among research articles. In this section, we provide an overview of the techniques based on the semantics of the contents of documents. The semantics is differently defined by authors subject to their proposed technique. Some have used structural scaffolding, which considers the position of the words within a document or measure the word/sentence sentiments [13]. Similarly, WordNet ontologies, an extensive lexical database for the English language, have defined semantics for retrieving the sense of words [30].

For understanding the internal meanings of the text of a research article word embedding have been largely used. Word embedding gives a numeric representation to words. The words having similar meaning will have similar representations. Word embedding provides a distributed
representation instead of ‘localist representation’ or ‘one hot encoding’ [31]. Distributed representation has the ability to represent a concept in a low-dimensional vector space. The vectors are dense, rather than sparse, as in case of ‘one hot encoding’. One of the key developments for inspiring operations of deep neural networking methods for natural language processing tasks is the distributed representation of text. Word embedding are known for providing this distributed presentation of a text, and therefore, making text feasible in training deep learning models.

B. RECENT LITERATURE FOR IDENTIFICATION OF RELEVANT ARTICLES AND CITATION INTENT

In this section we study recent research that is carried out for the identification of citation intent in order to find relevant documents. The summary of this literature is provided in Table 1 along with each of these approaches’ strengths and weaknesses. The methods fall in one of the categories that have been discussed above. Some methods use the benefits of more than one, such as utilising contents information and the metadata details [32]–[34].

The important, relevant information has always been the need for researchers to analyse the existing approaches and study in a particular research area. Keeping in view the importance of the need, understanding the semantics of relationship among research articles have been studied for more than a decade utilising different techniques. Recent research in this area is conducted by Sakib et al. [33], who proposed a collaborative approach using citation context and provided a 2-level citation relationship. They proposed a recommender system that takes a paper of interest (POI) and recommends a number of candidate papers that might be relevant to the POI. In the first stage the proposed framework extracts all the candidate papers that might be similar to the POI. The candidate list is based on two lists; the citation and reference lists. The citation paper list includes the paper citing POI while the reference papers are the ones cited by POI. A binary occurrence matrix is created for citation and reference of candidate papers with POI.

The similarity, using these matrices, are measured using the Jaccard similarity coefficient, named $J_{co-occurred}$ and $J_{co-occurring}$. The final score of a candidate paper is measured by normalising the values of $J_{co-occurred}$ and $J_{co-occurring}$. Thus by using a 2-level relationship of co-occurrence a list of Top-N most similar paper $C_1$, $C_2$, $C_3$, ..., $C_n$ are presented in recommendation list. They have used a very small publicly available dataset provided by Sugiyama et al. [35]. For comparison they have used McNee et al. [36], Liu et al. [6], Haruna et al. [21] as baseline and evaluated the proposed techniques using Precision, Recall, Mean Average Precision (MAP), and Mean Reciprocal Rank (MRR) measures. This method is simply based on the occurrence matrices and does not utilise the paper contents, or metadata for recommending similar paper. It is a statistical approach and have used a very small dataset experiment.

S. Hassan et al. [34] proposed using deep learning to classify the significance of a citation from a collection of cited publications. They claimed that not all of the references are equally relevant. To discriminate between significant and irrelevant citations, they utilised a deep-learning model based on the Long Short-Term Memory (LSTM) [37]. They also provided a machine learning-based classification strategy for selecting the highest performing features using the Random Forest (RF) [38] classifier. The writers have compiled a list of 14 characteristics of a citation context.

Predefined hand-engineered characteristics such as linguistic patterns derived from paper content were criticised by Cohan et al. [13]. They thought that by directly extracting improved representations from the data, they might get better results. They presented a multitasking framework for incorporating paper structural knowledge. As structural scaffolding, their framework has two tasks: 1. predicting section title and 2. predicting if a citation is required. Their scaffolding also predicts the citation intent of a citation as background, method, or result class. They also used crowdsourcing to build the SciCite dataset, which consists of 6,627 publications with a total of 11,020.

Cohan et al. [13] compared their model to [4]’s state-of-the-art technique for citation intent classification. In terms of Precision, Recall, and F1 metrics, [13] outperformed the previous results. Jurgens et al. [4] employed pattern-based characteristics such as phase sequence, parts of speech, lexical categories representing positive or negative sentiments, and particular categories such as words, which they expanded from the previous state-of-the-art technique. They took [39] pattern list and supplemented it with newly discovered patterns and classifications. They further investigated topic-based features by claiming that a topic thematic framing can extract the citation function. A citation context detailing technique, for example, is more likely to be associated with the ‘USES’ function, but a citation context giving any definition is more likely to be associated with the ‘BACKGROUND’ class.

They also studied a list of arguments that reflect a class of citations, as well as archetypal argument characteristics. They discovered commonly recurring arguments in syntactic locations for archetypal argument featuring. For the ‘EXTEND’ class of citations, for example, the words ‘follow,’ ‘unfold,’ and ‘extend’ are commonly used. The occurrence of an argument is represented by a vector. The resemblance of a citation to a citation class is determined by the average of those occurrences. This work has shown to be state-of-the-art research in this field, since it employed natural language processing characteristics in depth to measure the citation reason and importance. When referencing a research article, writers are attentive to the discourse structure and publication venue, according to this study.

Suppawong et al. [32] utilized both meta and contents for extracting the usage intent of algorithms within research articles. They investigated that AlgorithmSee [40], a prototype algorithm search engine within web documents, fail
to capture the properties including algorithm class, complexity, performance, type of problem and data-structure, and instruction-wise information. Furthermore, they highlighted the importance of understanding the impact and reason of using an algorithm. An algorithm extensively used to solve a problem can be categorized as generalizable and therefore can be recommended as solution to solve a particular problem in that domain. Whereas, an algorithm used as building block is highlighted as influential and can be recommended as benchmark to newly proposed algorithms. The automatic classification of why an algorithm is used in a literature is studied by using the citation context where the algorithm is referred. Although this study focuses on algorithm reason identification and may not be relevant to our study, but the core concept of automatic extraction of reason of citing an algorithm is directly relevant to our study.

Chen et al. [41] proposed Citation Authority Diffusion (CAD) technique for the identification of important survey papers. Their proposed system may be accessed via web service call names “Survey Importance Measurement (SIM)”. SIM automatically provides an authoritative paper list in a research area. Their approach consists of three modules including information collection, organization and presentation. Information collector consists of key concept extractor and survey material finder. Key concept extractor discovers the potential concept of the target paper. Material finder collects the relevant survey material simulating human behaviour for collecting information of interest. The information organizer discovers the potential survey papers and their relationship. The final module, information presentation, discovers the survey novelty score towards the target research. The novelty of papers is calculated on the bases of number of citations. Academic papers published before 2008 are extracted from CiteSeer in order to perform experiments. The testing dataset consists of 1,612 papers having quality references. They also performed experiments on a graph of papers extracted from Google Scholar. CAD recommends papers related to referenced article. It also suggests other related important papers that are not included in the bibliography of a paper. They can focus on the important papers among the recent relevant articles. The authors have claimed to achieve a considerable accuracy in terms of Recall and Co-cited probability measure.

Valenzuela et al. [42] proposed a supervised classification approach, using an SVM classifier, to indicate that the cited work is used or extended in the new work. They have modeled the task in two levels. At the first level, they have classified work into important and non-important. At the second level, they have classified references into four classes: related work, comparison, use, extension. They annotated a dataset of 450 citations, which is publicly available. Their approach is based on three key observations: 1) A paper is more important if it has a larger number of citations, 2) The importance of a citation can be judged from its appearance in a research article. For example, articles cited in the Method section are more important than ones cited in the Related Work section, 3) The citation form as direct or indirect may reflect its importance. The authors identified the direct citations using rules that follow the citation format using a regular expression. The indirect citations were identified by focusing on author names and description of the algorithm cited in a paper. Their approach outperformed the baseline achieving a precision of 0.65 for a recall of 0.9, compared to 0.2 by the baseline. Valenzuela et al. [42] have studied the importance of research articles but their proposed classifier is trained on a very small dataset of only 450 records, which cannot generalize the behavior of citations. Their study is based on multiple information from the full text of a paper, considering section information, citation counts, and the way a citation is made but that statistical information cannot contribute enough while studying them alone without looking at the internal context of a citation context.

Faiza et al. [43] proposed an important paper identification classification method based on meta information from the research articles. Their main objective was to analyze the extent to which metadata can behave similarly to content-based approaches. The proposed a binary supervised learning classification model using features like titles, authors, keywords and references, etc. They employed SVM, Kernel Logistic Regression (KLR), and RF classifiers to classify the documents as important or non-important. The calculated similarity and dissimilarity are based on metadata and content information. The metadata similarity is calculated based on title, authors, keywords, and bibliographic coupling. The content-based similarity is measured based on abstract and cue terms. Experiments were performed by analyzing different combinations of features while increasing the number of combinations in each set. They used two datasets having 465, and 488 paper citation pairs, respectively. The proposed is a good choice where the full content of a paper is not available, as it is based on metadata and has provided results very close to in-text citation-based methods. This approach is good in relating documents as important or non-important for each other but may fail to provide the reason for their relativity. This is because the classification is only binary in nature and the proposed method finds the global relatively of the documents. The datasets they have used are also very small for a large study and for training a machine learning-based model.

Nazir et al. [45] proposed a binary classification method and explored the significance of applying a threshold value of frequency count. The frequency count has significance in understanding the important citations from a manuscript. But the limits of the cut-off value of frequency count were never studied before and therefore this study attempted to set a threshold value. For evaluation purposes, they annotated 465 pairs by domain experts and compared their results with state-of-the-art precision values. They achieved a precision of 0.75 in comparison to the previous 0.72 state-of-the-art precision value.

1http://citeseerx.ist.psu.edu
| Paper | Inputs & Outputs | Strengths | Weaknesses |
|-------|-----------------|-----------|------------|
| **Paper:** Chen et al. [41]  
**Dataset:** CiteSeerX  
**Method:** Bibliographic information  
**Citation Intent:** No Measures  
**Recall, Co-citation Probability** | **Input:** List of papers cited  
**Output:** Relevant survey articles, topic discovery | - CAD recommends papers related to referenced article  
- It also suggests other related important papers that are not included in the bibliography of a paper  
- They can focus on the important papers among the recent relevant articles | - The system is experimented on CiteSeerX dataset only, and therefore is not generalized. The claimed accuracy may not be achieved in case of other sources. The papers also belong to only Computer Science domain, in which case results may change for other domains  
- The novelty assessment would be incomplete without fetching information from other data sources |
| **Paper:** Zhu et al. [44]  
**Dataset:** 3,143 pairs annotated by domain expert  
**Method:** Full contents of a paper  
**Citation Intent:** Yes Measures  
**Precision, Recall, F-measure** | **Input:** Count-based, similarity-based, context-based, and position-based features  
**Output:** Classified citation as Influential or Non-Influential | - Classified the citations as influential and non-influential with the claim that most of the citation are not influential and find the influential ones is a crucial   
- Out of five features the author concluded that citation frequency is the best feature for finding the influence of a reference | - The dataset used is very small  
- Domain expertise of the annotators is not explained which may be questioned |
| **Paper:** Valenzuela et al. [42]  
**Dataset:** Developed their own  
**Method:** Contents  
**Citation Intent:** Yes Measures  
**Precision-Recall Curve** | **Input:** Section in which citation made, number of citations, citation form(direct or indirect)  
**Output:** Important, Not Important | - Their study is based on multiple information from the full text of a paper, considering section information, citation counts, and the way a citation is made   
- Two level of citation classification is considered. Classifying citation into important and un-important and further as related, comparison, usage, or extend | - The dataset is too small for a machine learning to well generalized on given examples in each class  
- Although multiple features are considered for classifying the citations as important/unimportant yet the statistical information cannot contribute enough while studying them alone without looking at the internal context of a citation context |
| **Paper:** Jurgens et al. [4]  
**Dataset:** ACL-ARC  
**Method:** Contents  
**Citation Intent:** Yes Measures  
**Macro F1** | **Input:** Citation context  
**Output:** Classify citation context intent as one of six classes. | - Pattern-based features (lexical categories-sentiment, specific categories-we extend, ‘we build upon’)  
- Topic-based features(describing topic-background, describing methodology-USBS)  
- Prototypical Argument features(vector of arguments) and Why a paper is cited  
- Authors are sensitive to discourse structure and publication venue when citing a research paper | - The dataset is too small for machine learning to generalize on it, particularly in case of some classes like ‘future’  
- Pattern-based features may not be good for understanding the internal contextual meanings of a citation context  
- Although used natural language processing extensively but the feature cannot be fully automated |
| Paper | Inputs & Outputs | Strengths | Weaknesses |
|-------|-----------------|-----------|------------|
| Paper: Hassan et al. [34]  
Dataset: Developed their own [34]  
Method: Deep Learning, Machine Learning  
Citation Intent: Yes  
Measures: Precision and Recall | Input: Full-text of a paper. The citation context is provided as input to the algorithm  
Output: Identifies the citation as ‘important’ or ‘not-important’ | - Can help improve citation-based full-text summarization techniques  
- Identified 14 features of citation importance  
- The importance of citation is explored using more sophisticated method instead of scoring on the bases of number of citations | - The features are somehow identical, and most of them can be combined together  
- LSTM based model normally fails to capture the real context of words in a sentence, as the model does not read the tokens in both directions (left to right and right to left) |
| Paper: Faiza et al. [43]  
Dataset: –  
Method: Metadata  
Citation Intent: Yes  
Measures: F-measure | Input: Titles, Authors name, Keywords, Categories, and References  
Output: Important, Non-Important | - The study has achieved results almost similar to content-based approaches using meta information about the articles  
- This approach can be best utilized in cases where the full contents of a paper are not accessible | - The approach finds the global relevancy of a document and does not look into the citation context local intent  
- The classification is a binary one, classifying a paper as important or not important only |
| Paper: Cohan et al. [13]  
Dataset: ACL-ARC [4], Sci-Cite [13]  
Method: Contents  
Citation Intent: Yes  
Measures: Precision, Recall, F1 | Input: Citation Context, Section information  
Output: Citation Intent as ‘background’, ‘method’, or ‘results’ | - Bi-directional LSTM reads the context of words in a sentence from both sides  
- The three classes distribution seems more reasonable  
- They have also proposed a new state-of-the-art datasets  
- The dataset is not specific to a single domain | - Two bi-directional LSTM are trained seperately, which does not really provide the actual context of words in citation context  
- The dataset is crowdsourced and involves human/expert annotation which may be biased |
| Paper: Sakib et al. [33]  
Dataset: Sugiyama et al. [35]  
Method: Collaborative, Co-occurrence  
Citation Intent: No  
Measures: Precision, F1, MAP, MRR and Jaccard Coefficient | Input:  
- Paper of Interest  
- Cited papers  
Output:  
- Relevant papers  
- Useful papers  
- Recommends papers for Interest | - Does not depend on priori user profiles—a major issue in collaborative filing based approaches  
- Utilized 2-level paper-citation relations to find hidden associations between papers  
- Recommends papers of interest | - This method is simply based on the occurrence matrices and does not utilize the paper contents, or metadata for recommending similar paper  
- Used a very small dataset experiments  
- Does not extract the citation intent |
| Paper: Nazir et al. [45]  
Dataset: 465 pairs annotated by domain expert  
Method: Citation Frequency  
Citation Intent: Yes  
Measures: Precision, Recall, Accuracy, F-measure | Input:  
- Extracted data from ACL Anthology site  
- Converted to pdf  
- Extracted citation count from full text of the papers  
Output: Classified citation as Important or Non-Important | - Threshold of citation count is explored for the first time  
- Achieved improvement for binary classification in terms of precision as compared to state-of-the-art values  
- Comparatively simple technique as it used only one feature while the state-of-the-art used 8 features | - The dataset used is very small  
- Domain expertise of the annotators is not explained which may be questioned  
- Authors have different behaviors while citing the same research paper multiple times in an article. Some authors do not cite a paper frequently once cited already |

2https://github.com/slab-itu/imp_citations
TABLE 1. (Continued.) Recent literature for identification of relevant articles and citation intent.

| Paper: Suppawong et al. [32] | Input: Structure, Lexical, Morphological, Grammatical, Cue Words, Sentiment, Venue |
|-------------------------------|---------------------------------|
| Dataset: AlgoCite             | Output: Classify Algorithm citation function as Use, Extend, Mention, NotAlgo |
| Method: Contents              |                                 |
| Citation Intent: Yes          |                                 |
| Measures: Precision, Recall, F1, Receiver Operating Characteristic (ROC) |                                 |
| Weaknesses                    |                                 |
| Zhu et a. [44] used in-text, similarity, context, and position-based features for classifying citations into two classes: influential and non-influential. They argued that not all citations in an article are influential – are closely related. The authors further claimed that most of the citations are merely to provide background information on the proposed work. To evaluate their proposed citation classification method, they developed a dataset of 3,143 pairs from around 100 research publications. The crawled Association for Computational Linguistics Anthology and annotated the paper-reference pairs by domain experts for construction of the dataset. The classification model is trained using a support vector machine which gained 0.35 precision. The recent literature listed in Table 1 for citation intent classification and relevant document identification is mainly based on meta-information, which does not provide the complete details that may be extracted from the entire contents of research articles. Furthermore, the methods which are based on full contents have used statistical methods like counting the number of cue phrases based on various key-term identification approaches. The final category includes those directly related techniques that have utilized the contextual approaches to extract the citation intent from the citation context. Those approaches have their own limitations in capturing the real meaning of the context from the citation context. The bidirectional methods can capture the actual context in which a referenced article has been cited.

III. PROPOSED WORK

We discussed the existing state-of-the-art techniques for important, relevant article identification in the previous section. We observed that the citation context plays a vital role in predicting citation and identifying the relationship between cited and citing paper. This section discusses our proposed method for automated citation classification and measuring the similarity between two articles, as depicted in Figure 8. We first discuss the selection of features for our proposed classification model training, followed by the development of the model for intent classification.

A. FEATURE SELECTION FOR TRAINING THE MODEL

The input to our proposed system includes citation context and section in which in-text citation has been made. Literature has made several attempts [19], [20] to investigate the importance of various features to understand the reason of a citation and thus measure relativity among the research documents. Citation context has proved to be the most effective in finding the reason for citation [4], [13]. The previous study has made attempts to understand the citation context by using the statistical approaches and natural processing approaches. The natural processing methods have mainly utilised text feature extraction. Recently the importance of contextual analyses of text has been explored after new advances in natural language processing, including LSTM [37], two-way LSTM [12], and transformers [46] for contextual analyses of the text. Therefore, we have also selected the citation context as input to the system. Several methods have been used in the literature to extract the citation context. They have set different citation context window sizes, including the citing sentence, the sentence before and after it, and the complete paragraph in which a paper has been cited.

B. CITATION INTENT CLASSIFICATION MODEL DEVELOPMENT

When the system is provided with the input features, it predicts the citation reason of a referenced articles. We have developed and trained a classification model for citation intent extraction. Training a huge number of text data, starting with language modeling and then domain specific training, requires tremendous amount of computation and time. Computers do not understand words rather they work on numbers and matrices, called word embeddings. The idea is to map each word to a point in space where similar words in meaning are physically closer to each other in the embedding space, depicted in Figure 9.
The text data is a sequential data where the input has some defined order. If we use Recurrent Neural Networks (RNN), for calculating the word embeddings, it will take a lot of time for our sequence-to-vector neural model. The network will also have vanishing/exploding gradient effect for long sentences. A truncated back-propagation version will also be not applicable in our case due to longer sentences and hence longer size of networks. Long Short Term Memory (LSTM) [37] solves problem by skipping a lot of processing by retaining the memory for longer sequences. This will solve the problem for longer sentence but for average text data LSTM have proved to be slower than RNN. The main problem with LSTM is that the sequential data is passed serially rather than in parallel. LSTM also fails to capture the true meaning of words as LSTM networks learn from left to right only. The bidirectional-LSTM [12] are shallow networks of right-to-left and left-to-right networks. They are learning left-to-right and right-to-left context separately and then concatenating them, therefore, the true context is lost.

We needed to parallelize the sequential data to make the process efficient. We also wanted to learn both right-to-left and left-to-right contexts simultaneously to truly capture the context of a sentence. Transformers network architecture solved this problem in 2017 [48] by introducing encoder-decoder architecture, much like RNN networks. In the case of RNN, we pass input sequences one after the other, managing the input’s timestep. Embeddings are generated one timestep at a time. The current word hidden states have dependencies with the previous word embeddings. In the case of transformers, there is no concept of timesteps for the input. We pass all the words of a sentence simultaneously and determine the word embedding at once. This increases the network performance tremendously by processing all the words in parallel and captures the actual context of a sentence by looking at both left-to-right and right-to-left contexts.

Transformers include both the encoder and decoder having a stack of six encoder and decoders [48]. The encoding component produces the embeddings of the input text having context information. The decoding component takes
the embeddings produced by the encoding component and map the input context to the output context. For example, in the case of sequence-to-sequence models like language translation, the encoding component captures the context of the input sentence. It passes the input context to the decoder component, which produces the output language sentence. Both components can work separately, have some underlying understanding of the language, and therefore can build a new architecture based on transformers. One such example is the Bi-Directional Encoder Representation from Transformers (BERT) [49]. The transformers alone could be used for language translation, but the BERT model can be used for various tasks, including neural machine translation, question answering, sentiment analysis, and text summarization.

The reason behind this fact is that BERT extracts the internal contextual representation of a text, and that representation can be used for a variety of tasks. BERT has two models: BERT base and BERT large. The BERT base has 12 transformer blocks consisting of 12 attention heads trained on 110 million parameters [49]. BERT large has 24 layers with 16 attention heads trained on 340 million parameters, depicted in Figure 8 and 11.

BERT is trained in two steps. The first step, pre-training, is general for all the tasks while the second step, fine-tuning, is for a specific downstream tasks. The steps are explained in detail in next section.

1) PRE-TRAINING STEP
Pre-training is based on the concept of Transfer Learning (TL) [50]. TL is a machine learning concept that aims to transfer knowledge gained during the process of solving one related problem to a different but related one [50]. The practical implementation of TL in the case of word embedding is by training a neural network on a relatively more extensive set of text data and transferring that knowledge to a domain-specific problem on text data. This concept has long been used in computer vision for a couple of years now, where a model is pre-trained on the huge ImageNet corpus [51]. The model, thus, learns on general image features rather than randomly initialized. The model is then fine-tuned for any vision-specific task. In the case of text data, the pre-training procedure mostly follows the existing literature on language model pre-training. BERT uses the BooksCorpus having 800M words and English Wikipedia having 2,500M words, for the pre-training step. The focus was mainly on long sentences rather than lists, tables, and heading while pre-training the model [46].

BERT has been trained for two tasks simultaneously; Masked Language Model and Next Sentence Prediction.

1) Masked Language Model (MLM): For training a language model, a prediction goal is set; typically, the next word prediction (“Amir was hit by a _____ ”). The next word predictions adopt directional approaches, as discussed earlier, usually left-to-right. The directional approaches fail to capture the context of a sentence. BERT adopts the MLM approach to overcome the problem, in which 15 percent of the words are masked – kept hidden with a [MASKED] token. The prediction goal is thus set to predict the masked words only, without adjusting the model parameters for the rest of the tokens. The rest of the terms provides the context to the masked words. The loss function ignores the prediction of the non-masked words, consequently increasing the context-awareness.

2) Next Sentence Prediction (NSP): The next sentence prediction task predicts the subsequent sentence of an input sentence. During training, 50% of the inputs are the correct pairs in which the second sentences are the ones from the original document. The rest of the 50% pairs have the second sentence randomly selected from the corpus. The prediction goal is to find the pair having randomly selected the second sentence. The training process requires tokenizing the sentences and adding sentence and positional embedding to each of the tokens. A softmax is added to calculate the probability of isNextSentence.

2) FINE-TUNING STEP
The model learned the language and context using masking and next sentence prediction tasks in the pre-training step. Now the model has a general idea of how the language is framed. To use the language for a specific task, we must fine-tune the earlier trained model on more task-specific natural language processing tasks. We need to replace the fully connected output layers with a new set of output layers, as shown in Figure 10.

We can perform a supervised training of the initially trained model with a task-specific training set, which in our case is the labelled citation sentences. The underlying task in our case is the citation classification, and therefore we input the labelled dataset containing citation context annotated with citation intent classes.

The model is fine-tuned on the basis of input labels. The training does not take long as only the output parameters are learned from scratch, whereas the rest of the model is slightly fine-tuned. As a result, training time is much faster while capturing both the right-to-left and left-to-right context of each of the words in a citation context, providing better contextual analyses.

Figure 11 explains our proposed model for the citation intent classification describing the encoding of citation context and predicting the citation intent. Assuming Figure’ citation’, let R denotes a reference/citation and C denotes the sentence in which the article is referred. The tokens of C are given by \( T_c = t_1, t_2, \ldots, t_n \). Each of the tokens is fed into the contextual embedding. The tokens pass through 24 stacked encoders. The encoders contain a self-attention and feed-forward network. The encoders are stacked such that the output of one encoder becomes the input to the next encoder, and finally, we get the contextualized word representation of the input citation context. The contextualized
word representation is then fed into a classifier, classifying the citation context as one of the citation intents.

Once we have the citation intent of all the citations, we can measure the relevancy of a cited paper with a citing paper based on the assigned citation intent class and its section information in which the article has been cited. We explain the citation scoring calculation in the next section.

IV. RESULTS AND DISCUSSION
In this section, we demonstrate the practical implementation of our proposed methodology using various datasets. We set an experimental environment for executing our method. We then simulate our work using the Kaggle platform using Python language. We explain the functions of various Python libraries used during simulation. We also provide a comparison of the results of our proposed method with the existing state-of-the-art methods. For comparison, we used various measures, including Precision, Recall and F1 score.

A. EXPERIMENTAL SETUPS
We performed experiments using the Kaggle platform. We used three datasets, given in Table 2. The ACL-ARC dataset has six intent classes, whereas SciCite and C2D-I datasets have only three intent classes, as shown in Table 2. ACL-ARC and SciCite are manually curated, whereas C2D-I is a synthetic dataset. C2D-I is relatively much larger than the other two datasets and, therefore, may potentially provide better results when used by deep neural network approaches as they generalize well on large datasets. For comparison on each of these datasets, the intent classes must be the same, and therefore we removed the extra classes from the ACL-ARC dataset having less than a hundred instances, including Motivation, Extension, and Future Work. It is difficult to generalize the model based on such a small number of instances, as argued by [7]. The Uses class in ACL-ARC is the same as the Method class in SciCite and C2D-I. Similarly, the Comparison class is the same as the Results class in SciCite and C2D-I. After this assumption, we have the same number and label of intent classes in all of our selected datasets. We also balanced the C2D-I dataset, as recommended by [52], by taking the same number of instances from each of the citation intent classes. Having the datasets, we are ready to perform the experiments using Python language in the Kaggle platform. The python code along with with the additional files can be found at github reposity https://github.com/romankht84/BERTcic.

The datasets contain labelled citation contexts, which are fed into the model as input parameters. For contextual embedding of the citation intent, we have proposed our model in the previous section. We have used BERT embedding for the contextual representation of citation sentences. It takes one or two sentences as input. Each of the sentences is separated
by [SEP] token. In our case, we input only one sentence as in the text classification task, unlike the QuestionAnswer task, which takes two sentences. The input text starts with a special token [CLS]. BERT requires both of the tokens, [SEP] and [CLS], even if we are supplying only one sentence, as in our case.

In case of two sentences (an example from ACL-ARC dataset):

[CLS] It was used to obtain value to define \n the utility → model

[SEP] This was done by MERT optimisation towards post ← -edits under the TER target metric

In the case of one sentence (an example from ACL-ARC dataset):

[CLS] It was used to obtain value to define the utility model [SEP]

1) TOKENISATION

The first step in converting the words to numerical representation is tokenising the input text. We used official tokenisation script created by the Google team for BERT which is based on WordPiece [26] tokenisation. Before tokenisation, the text is converted to lower case and then tokens are created as given below:

```python
3!wget --quiet https://raw.githubusercontent.com/tensorflow/models/master/official/nlp/bert/tokenization.py
```

```python
text = tokenizer.tokenize(text)
text = text[:max_len-2]
input_sequence = ["[CLS]" + text + "[SEP]"
```

TABLE 2. Selected datasets for citation intent classification.

| # | Dataset | Dataset Type | Intent Classes and Instance Distributions |
|---|---------|--------------|------------------------------------------|
| 1 | ACL-ARC | Manual       | • Background = 987
• Uses = 363
• CompareOrContrast = 347
• Motivation = 86
• Extension = 72
• Future Work = 69 |
| 2 | SciCite  | Manual       | • Background = 6,375
• Method = 3,154
• Results = 1,491 |
| 3 | C2D-I   | Synthetic    | • Background = 10,000
• Method = 10,000
• Results = 10,000 |
The tokens created from the input citation context are:
\['[CLS]', 'it', 'was', 'used', 'to', 'obtain', 'value', 'to', 'define', 'the', 'utility', 'model', '[SEP]']

In case a word is not available in BERT vocabulary it is split into subwords, for example:

**Sentence:** 'This sentence is used for Embedding'

Will have the following tokens:
\['[CLS]', 'this', 'sentence', 'is', 'used', 'for', 'em', '##bed', '##ding', '[SEP]']

Notice that the word 'embedding' has been split into sub-words having first word followed by ## representation of the sub-words. An extreme level of sub-words can have single characters of a word. BERT vocabulary consists of approximately 30K tokens [49]. After tokenising the citation context, the sentence is converted to vocabulary indices, for example in Table 3.

tokens = tokenizer.convert_tokens_to_ids

→ (input_sequence)

2) SEGMENT ID

The initial training of BERT is performed on sentence pairs, and therefore it assumes input to be given in pairs. It uses 1s and 0s to distinguish between the two sentences. In our case, however, we have provided only one sentence; therefore, we have set the segment Id 1s for all the tokens of a citation context.

segments_ids = [1] * len(input_sequence) + [0] * pad_len

The pad_len is used for padding the sentences which are shorter than the max_length parameters which is set to 512 in our case. A sentence can be of maximum 512 size. The rest of the size, in case of smaller inputs, will be padded to make the vectors of same size.

Having the input ready, we set the model for fine-tuning. We used the tensorflow_hub package for accessing the TensorFlow Hub repository. A pre-trained model of BERT “bert_en_uncased_L-24_H-1024_A-16/1” is added from the repository. This model is trained on lower case words having 24 transformer blocks (hidden layers). The tensors have 1024 hidden states with an attention size of 16. The pre-trained model is then fine-tuned by supplying the datasets to the pre-trained model. We tried to be as closer to the basic model of BERT as possible. The shape of the input tensor is given by (batch_size, max_len, hidden_dim)

3) BATCH SIZE

Batch size is a hyper-parameter. It specifies the sample count to be used for adjusting the model parameters. The model parameters can be adjusted based on a single sample (in case of stochastic gradient descent (SGD) [53], all training samples (in case of batch gradient descent), or a group of training samples (in case of mini-batch gradient descent). We have set 32 as the batch_size value for our experiments. The number of steps during each epoch is, therefore, the training size divided by the batch_size.

4) SIZE OF INPUT SAMPLE

The maximum size of the input to the model is 512. The rest of the text is simply truncated by the BERT model. Some other studies take end part of the input text, the first part of the center text from the input instance. For longer sentences [54]–[56] have proposed a solution that can take longer inputs than 512 words.

5) EPOCHS

Epochs define the number of times each sample/batch is used while fitting the model. It is a hyper-parameter that is sent as an argument to learning algorithms. Due to the complexity of our model, dataset size, and the number of parameters, each epoch took 20 to 40 minutes. We also used TPU accelerators to reduce the training time. Traditionally, the epoch does not take much time, the epochs are usually in hundreds, but we used the early stopping strategy. The early stopping stops the training process once there is no further improvement in terms of accuracy or loss. Our training was completed in 34, 17, and 10 epochs for ACL-ARC, SciCite, and C2D-I datasets.

6) VALIDATION SPLIT

We used a stratified split of 80 - 20% from the training set of each of the datasets for training and validation set, meaning that training will not be performed on 20 percent of the training data. Only 80% of the records will be used to train the model during each epoch. The rest of 20% will be used to validate the results after each epoch. We then calculate accuracy and loss for both training and validation.

7) TRAINING OPTIMIZER

We selected the Adam optimizer [57] as it is the de-facto optimizer in many cases. Optimizers define how neural networks learn. They adjust the model parameters such that the loss function is at its lowest. Some of the famous optimizers include gradient descent, SGD, SGD with momentum, Adam, and Adam [58]. Each of the optimizers has its benefits in a particular type of problem. The choice of selecting the optimizer is more empirical than mathematical. Adam works well in most cases and therefore is the de-facto optimizer for most of the neural networks.

Furthermore, we used ‘softmax’ as the activation function, ‘categorical_crossentropy’ as loss, and ‘accuracy’ as the metrics.

8) MODEL OUTPUT

After setting the model and feeding data to it, we train the model. Understanding the model output is essential. The output can be taken in two ways:

1) **Sequence Output:** This form of output is the sequence of embeddings (hidden-states) at the last layer of the model. It includes embeddings for individual tokens, including the [CLS] token. Our input example will produce seven embeddings, having one embedding for
TABLE 3. Assignment of vocabulary indices to tokens.

| Tokens | [CLS] | it | was | used | to | obtain | value | define | ... | [SEP] |
|--------|-------|----|-----|------|----|--------|-------|--------|-----|-------|
| Indices | 101   | 11,232 | 343 | 324  | 532 | 349    | 9,201 | 5,843  | ... | 102   |

2) **Pooled Output**: The pooled output provides a single output embedding for a citation context, as depicted in Figure 13. It uses the [CLS] token embedding as the final embedding, which is further processed by a linear layer and Tanh activation function [59]. The original paper follows the pooled output for the text classification problem, and therefore we have also used the pooled output in our case.

The last layer of hidden states holds the output in the following shape (batch\_size, hidden\_dim), which is (32, 1024) in our case.

The BERT further has the classification layer, which classifies the final embedding to one of the citation intent classes. Unlike other word embedding techniques, BERT has the beauty to classify the vectors to a binary or a multi-class (as in our case). Our final classes include the background, method, and result.

**B. RESULTS AND EVALUATION**

In order to evaluate our proposed techniques, we performed experiments on our three selected datasets, C2D-I, SciCite, and ACL-ARC, listed in Table 4, having 30,000, 11,020, and 1,924 annotated citation contexts.

The datasets are split into training and test sets. The ACL-ARC dataset, developed by [4], has the least records in it. The training set has 88% instances, while the test set has 12% records. The records in the training and test set are annotated for the citation intent classes. The Background intent class has the most extensive set containing 58% of the training set size, while Method and Result classes have 21% instances, making the dataset imbalanced. The distribution is almost the same in the test set as well. This is because the Background class typically has the highest number of citations. In the SciCite dataset, the test set has the 0.25 split. The sets are already divided, and therefore we kept the split ratio unchanged while using the dataset for training and then testing. This dataset is also unbalanced having 55 - 60% records in the Background class. Finally, we selected 30,000 instances from the C2D-I dataset with the test set split of 0.2.

We created three models $M_1$, $M_2$, and $M_3$ based on ACL-ARC, SciCite, and C2D-I datasets, respectively. Each model is trained on the training set of the respective dataset. The models are then tested using the test part of the datasets. Precision, Recall, and F1 measures were then calculated for each model to validate our proposed technique’s effectiveness on different datasets and compare the results with the previous methods setting the same benchmark. For a classification task, as in our case, these measures have proved to be helpful evaluation metrics [60]. We discuss the results of each of these models in detail.

1) **ACL-ARC DATASET MODEL TRAINING AND RESULTS**

Figure 14 is the accuracy curve for training and validation of model $M_1$, based on ACL-ARC dataset. The training stopped after 34\textsuperscript{th} epoch. ACL-ARC dataset is small; therefore, the training time for epoch is lesser. The training curve is not smooth, varying the accuracy up and down. The accuracy goes down near the fifth epoch and fifteen after achieving local optima. It then goes smooth until there is no further improvement in training accuracy. The validation accuracy goes up and down, continuously trying to fit itself. Owing to the smaller dataset and complex nature of the model, it does not fit perfectly. We can say that our proposed model is not good for smaller datasets, and therefore, a larger dataset must be used while using transformer-based deep learning approaches. The loss curve portrays almost a similar behav-
TABLE 4. Distribution of dataset instances for training.

| Dataset   | Instances | Training Record Distribution | Test Record Distribution |
|-----------|-----------|------------------------------|--------------------------|
|           |           | Total | Background | Method | Result | Total | Background | Method | Result |
| ACL-ARC   | 1,924     | 1,489 | 867       | 317    | 305    | 208   | 120        | 46     | 42     |
|           |           | 88%   | 58%       | 21%    | 21%    | 12%   | 58%        | 22%    | 20%    |
| SciCite    | 11,020    | 8,243 | 4,840     | 2,294  | 1,109  | 2,777 | 1,535      | 860    | 382    |
|           |           | 75%   | 59%       | 28%    | 13%    | 25%   | 55%        | 31%    | 14%    |
| C2D-I     | 30,000    | 24,000 | 8,000   | 8,000  | 8,000  | 6,000 | 2,000      | 2,000  | 2,000  |
|           |           | 80%   | 34%       | 33%    | 33%    | 20%   | 34%        | 33%    | 33%    |

FIGURE 14. Training and validation accuracy curve.

ior in training and validation curves. Further, testing on the developed $M_1$ is performed on the test records as given in Table 4.

The test results are described using a confusion matrix in Figure 15. The confusion matrix depicts the ratio of predicted intent classes to the actual labels in the dataset. The matrix quantifies the true-positive, false-positive, true-negative, and false-negative for the multi-class problem. The total 208 records from the test set are predicted by the $M_1$. We see the actual class on the vertical side, whereas the predicted class of intent is shown horizontally. The color intensity describes the density of occurrences in that cell. The diagonal cells show the number of correctly classified instances, which is high compared to the rest of the values in each of the classes. The true-positive values for the Background class are higher as compared to the Method and Results class. This shows that the model predicts the Background class well while it can hardly predict the other classes. This behavior is due to the higher number of instances in the Background class while having very few instances labeled with other intent classes. The trained model could not get generalized well for those classes. This also affects the overall F1 score of the ACL-ARC model.

The confusion matrix alone does not provide sufficient information; therefore, we have calculated the other measures of accuracy, as used by [4], [13]. Table 5 lists Precision, Recall, and F1 scores for each class using our selected datasets. The difference in the F1 score of the individual classes is higher due to the imbalanced nature of the dataset. The overall Precision, Recall, and F1 score is given in the last column of the table. The F1 score of $M_1$ is 71.02%, which is better than the previously achieved state-of-the-art (SOTA) as explained in Section IV-C.

2) SCICITE DATASET MODEL TRAINING AND RESULTS

Figure 16 shows accuracy curve for training and validation of $M_2$, based on SciCite dataset. The training stopped after seventeen epochs. The validation accuracy of $M_2$ is better than the training accuracy, which is very rare. This may happen because we have added a dropout that sets a percentage of features to zero during training and does not count them. Dropout helps in generalizing the model instead of overfitting it. During validation, all the neurons are used, making the model more robust, and thus an increased accuracy is achieved during validation. The second reason may be that the validation set is easier, which may be true for a few sentences but not all the sentences in the validation set. Overall the train-
TABLE 5. Precision, Recall and F1 scores of citation intent classes using our selected datasets.

| Dataset | Background | Method | Result | Overall Score |
|---------|------------|--------|--------|---------------|
|         | Precision  | Recall | F1     | Precision     | Recall | F1     | Precision | Recall | F1     |
| C2D-I   | 0.89       | 0.89   | 0.89   | 0.90         | 0.89   | 0.89   | 0.89      | 0.90   | 0.90   | 89.3%  | 89.3% | 89.23% |
| SciCite | 0.89       | 0.87   | 0.88   | 0.81         | 0.85   | 0.83   | 0.82      | 0.81   | 0.81   | 85.45% | 84.17% | 84.07% |
| ACL-ARC | 0.79       | 0.81   | 0.8    | 0.68         | 0.74   | 0.71   | 0.69      | 0.57   | 0.62   | 71.81% | 70.62% | 71.02% |

Further testing of the model is performed using the test part of the dataset having 2,777 records. The results are described using a confusion matrix in Figure 17. The true-positive values in the diagonal of the matrix have much higher values than the wrongly predicted intent classes. The Precision, Recall and F1 scores are listed in Table 5.

3) C2D-I DATASET MODEL TRAINING AND RESULTS

Figure 18 is the accuracy curve for training and validation of model $M_3$, based on C2D-I dataset. The accuracy curve in Figure 18 for training is very closed yet better than the validation curve, showing that the model is well trained and regularized. The starting accuracy is very less with larger jumps in accuracy after training epochs. Once a significant accuracy is achieved, there is minimal change in accuracy which finally becomes negligible, where the model stops further training, and we use the trained model for testing using the test part of the C2D-I dataset.

The test results are shown in Figure 19 using a confusion matrix. The total 6,000 records from the test set are predicted by the $M_3$, which is trained on the C2D-I dataset. The true-positive values for the Method and Result class are higher than the Background class as the model was more accurate in predicting these two classes. This may be because the Background citation context is more generalized and does not have any specific high-frequency words.

The results of $M_3$ are better than the other two models. This is clearly because the $M_3$ is trained and tested on a balanced dataset. A balanced dataset can generalize a model for each of the classes instead of being biased for a particular class of intent, as argued by [52].

The confusion matrix alone does not provide sufficient information; therefore, we have calculated the other measures of accuracy, shown in Table 5. The table lists Precision, Recall, and F1 scores for each class using our selected datasets.

The difference in the F1 score of the individual classes, in Table 5, is almost negligible as the C2D-I is a balanced dataset.
TABLE 6. Comparison of the results using various dataset with previous SOTA.

|   | Dataset  | Original | State-of-the-art | BERT<sub>base</sub> | BERT<sub>CIC</sub> |
|---|----------|----------|------------------|---------------------|-------------------|
| 1 | ACL-ARC  | 54.6     | 67.9             | 66.32               | 71.02             |
| 2 | SciCite   | 84       | 84               | 79.56               | 84.07             |
| 3 | C2D-I    |          |                  | 84.65               | 89.23             |

FIGURE 19. Multi-class confusion matrix of our proposed methodology using C2D-I dataset.

dataset, and therefore the model is well generalized for each of the classes. The overall Precision, Recall, and F1 score is given in the last column of the table. M3 has achieved a new state-of-the-art (SOTA) F1 score of 89.23% which is a significant increase compared to the previously achieved best results. The gain can be considered much higher considering the title and citation worthiness scaffoldings in the previous SOTA, discussed in brief in Section IV-C.

C. COMPARISON WITH THE BENCHMARK

In order to understand the gain in prediction accuracy of our proposed technique, we compared the results with previous state-of-the-art (SOTA) techniques, as shown in Table 6. The table provides the F1 score of classification prediction reported by the original paper of the dataset. The table then lists the SOTA F1 score that has been reported by any paper using a specific dataset. We have further used the BERT<sub>base</sub> and our proposed BERT<sub>CIC</sub> on these datasets to compare the accuracy measures with the previous benchmarks.

In the case of ACL-ARC, the originally reported results by Jurgens et al. [4] are improved by Cohan et al. [13]. Because Cohan et al. employed two scaffoldings, title and citation worthiness, in addition to Embeddings from Language Model (ELMO) [12], our BERT<sub>base</sub> model has a lower F1 score than the SOTA predictions. However, our proposed BERT<sub>CIC</sub> has performed better than the SOTA F1 score with a ∆3.12. The F1 score gain achieved, in comparison to the originally reported results, is 16.42. The ACL-ARC dataset is relatively tiny, so the models generally cannot perform well due to their short size, particularly for the intent classes like result and method where the number of instances is too tiny for a deep learning model to train.

When comparing the F1 score for the SciCite dataset to its fine-tuned version, there is not much of a difference, as shown in Table 6. Our proposed model produces results that are virtually identical to the fine-tuned version of SciCite, which also considers the section title in which the citation is made as well as the citation worthiness. We could not compare the results to SciCite without scaffolding since Cohan et al. didn’t disclose them. However, SciCite reports 77.8, 78.1, and 79.1 for section title scaffolding, citation worthiness, and both scaffolding, respectively. These results clearly dictate the minimal improvement of ∆6.27. The F1 score improvement in the case of SciCite is approximately two times that of ACL-ARC as this dataset is comparatively larger in size, and therefore the model is trained better.

The proposed model shows a significant improvement when trained on our own dataset, C2D-I. We have achieved a gain of 11.43 compared to the single scaffolding result of Cohan et al., which is considered the highest score achieved on the citation intent classification task in literature. Looking at Table 5 we observed consistency in the Precision, Recall, and F1 scores of the C2D-I dataset as the dataset is balanced, and the training is equally performed for each of the intent classes.

The results presented above indicate the improvements in the prediction of citation intent class using our proposed model. The addition of some prominent scaffolding like section title, frequency of citing an article, authorship, and citation network can make the classification better.

V. CONCLUSION

The number of research articles being published every day is increasing with newly emerged areas in science. For understanding a particular area, the current literature plays a vital role. Research papers refer to previous work in order to support their claim or information they are sharing. The citations, therefore, are essential to understand the relevant papers. A network of citations is formed, having the referenced article of every paper in it. Each citation in a paper may have its reason for citing a paper. Some citations are merely for the purpose of definitions or area exploration. Others may refer to a previous benchmark, using a methodology, comparing results, or extending someone’s work. Understanding the reason for a citation, called the citation intent, can help understand the type of relationship. The citation intent can also describe the strength of the relationship as some intent classes have a stronger association than others.
The type of relationship among research articles has long been studied using different information related to publications. The approaches are mainly categorized as metadata, bibliographic, collaborative, or content-based. Although the rest of the approaches seem realistic and have simple mathematical reasoning, one can understand that co-authorship frequency has a significant impact on understanding the relevance of the research article. However, we cannot understand the relation type unless we have access to the full content of a paper. The paper contents further have various sections of which the most relevant part is the citation context. Citation context has the actual text which describes the referenced article. It is simple for a human to understand the citation reason, but machines require extensive training to understand the semantics of a text.

Advances in natural language processing have made computers able to understand the sentiment and intent of a text. For machines to understand the text, it must be transformed into a machine-readable format. The machines can work on numeric data; therefore, the citation context must be converted to a numerical representation of word embedding. Researchers have long studied word embeddings. Natural languages do not have discrete meanings like numbers. The same words have different meanings depending on their position in a sentence. The same sentence may have different meanings depending on its position in the paragraph. Statistical non-contextualized word representations like one-hot vector, CBOW, n-gram, and word2vec have tried to capture the context of words in a sentence. However, new contextual representations, including ELMO, Infersent, and BERT, have been proven a paradigm shift in natural language processing. The contextual representation has solved many problems, including polysemy words. Polysemy means words with entirely different meanings depending on a word’s position; for example, ‘apple’ can be a fruit or a company name. Therefore, we have proposed a contextual word representation of citation context to capture the intent of a citation. We have utilized a transformer-based deep neural model for vector representation of words. The deep model training can be more generalized if we train it on a large dataset; therefore, we developed our own synthesized dataset. We then trained our model for following a chain of complex tasks, starting from text preprocessing. We further pooled the output to feed it to the final classification layer to classify the citation context into one of the three classes of intent.

We have performed detailed experiments to evaluate our proposed methodology. We used Precision, Recall, and F1 score to evaluate our classification model. In the first step, we checked the consistency of our proposed method on various datasets. We observed better results in the case of a huge dataset, which proved our first hypotheses. To further validate our proposed methodology, we compared our results with previous benchmark methods. We observed an accuracy of 89 percent. This proved our second hypothesis which is regarding the transformer-based contextual word representation of citation context classification.

This study has proved a significant contribution to finding important, relevant research articles. The study can also make the current indexing services more robust in giving the most pertinent information and categorizing the cited articles by their reason for citation.

In the future study, we can consider the other NLP-based meta information, other than the citation context, for understanding the citation relevancy and strength using deep contextualized word embedding. We can also combine the meta and NLP-based parameters as multi-folded features to a recommender system for recommending citation on the fly.

REFERENCES

[1] M. J. Moravcsik and P. Murugesan, “Some results on the function and quality of citations,” Social Stud. Sci., vol. 5, no. 1, pp. 86–92, 1975.
[2] A. M. Khan, A. Shahid, M. T. Afzal, F. Nazar, F. S. Alotaibi, and K. H. Alyoubi, “SWICS: Section-wise in-text citation score,” IEEE Access, vol. 7, pp. 137090–137102, 2019.
[3] S. Teufel, A. Siddharthan, and D. Tidhar, “Automatic classification of citation function,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2006, pp. 103–110.
[4] D. Jurgen, S. Kumar, R. Hoover, D. McFarland, and D. Jurafsky, “Measuring the evolution of a scientific field through citation frames,” Trans. Assoc. Comput. Linguistics, vol. 6, pp. 391–406, Dec. 2018.
[5] H. Raja and M. T. Afzal, “Paper recommendation using citation proximity in bibliographic coupling,” Turkish J. Elect. Eng. Comput. Sci., vol. 25, no. 4, pp. 2708–2718, 2017.
[6] H. Liu, X. Kong, X. Bai, W. Wang, T. M. Bekele, and F. Xia, “Context-based collaborative filtering for citation recommendation,” IEEE Access, vol. 9, pp. 9982–9995, 2021.
[7] M. Roman, A. Shahid, S. Khan, A. Koubaa, and L. Yu, “Citation intent classification using word embedding,” IEEE Access, vol. 9, pp. 1695–1703, 2015.
[8] A. Prasad, M. Kaur, and M. Y. Kan, “Neural ParsCit: A deep learning-based reference string parser,” Int. J. Digit. Libraries, vol. 19, no. 4, pp. 323–337, Nov. 2018.
[9] A. Shahid, M. T. Afzal, M. Abdar, M. E. Basiri, X. Zhou, N. Y. Yen, and J.-W. Chang, “Insights into relevant knowledge extraction techniques: A comprehensive review,” J. Supercomput., vol. 76, no. 3, pp. 1695–1733, Mar. 2020.
[10] A. Shahid, M. T. Afzal, M. Q. Saleem, M. S. Idrees, and M. K. Omer, “Extension of direct citation model using in-text citations,” Comput. Mater. Contin., vol. 6, no. A, pp. 3121–3138, 2021.
[11] D. R. Radev, P. Muthukrishnan, V. Qazvinian, and A. Abu-Jbara, “The ACL anthology network corpus,” Lang. Resour. Eval., vol. 47, no. 4, pp. 919–944, 2013.
[12] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., (Long Papers), vol. 1, 2018, pp. 1–30.
[13] A. Cohan, W. Ammar, M. van Zuylen, and F. Cady, “Structural scaffolds for citation intent classification in scientific publications,” 2019, arXiv:1904.01608.
[14] E. Garfield, “The impact factor,” Current Contents, vol. 25, no. 20, pp. 3–7, 1994.
[15] J. H. Hirsch, “An index to quantify an individual’s scientific research output,” Proc. Nat. Acad. Sci. USA, vol. 102, pp. 16569–16572, Nov. 2005.
[16] F. Xia, H. Liu, I. Lee, and L. Cao, “Scientific article recommendation: Exploiting common author relations and historical preferences,” IEEE Trans. Big Data, vol. 2, no. 2, pp. 101–112, Jun. 2016.
[17] M. Kessler, “Bibliographic coupling between scientific papers,” Amer. Document., vol. 14, no. 1, pp. 10–25, 1963.
[18] H. Small, “Co-citation in the scientific literature: A new measure of the relationship between two documents,” J. Amer. Soc. Inf. Sci., vol. 24, no. 4, pp. 265–269, 1973.
[19] M. T. Afzal, N. Kalathuramaiyer, and H. A. Maurer, “Creating links into the future,” J. Universal Comput. Sci., vol. 13, no. 9, pp. 1234–1245, 2007.
[20] N. A. Sajid, T. Ali, M. T. Afzal, M. Ahmad, and M. A. Qadir, “Exploiting reference section to classify paper’s topics,” in Proc. Int. Conf. Manage. Emergent Digit. Ecosyst. (MEDES), 2011, pp. 220–225.
[21] K. Haruna, M. A. Ismail, D. Damiasih, J. Sutopo, and T. Herawan, “A collaborative approach for research paper recommender system,” PLoS ONE, vol. 12, no. 10, Oct. 2017, Art. no. e0184516.

[22] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” Adv. Artif. Intell., vol. 2009, pp. 1–19, Oct. 2009.

[23] M. Nilashi, O. Ibrahi, and K. Bagherifard, “A recommender system based on collaborative filtering using Ontology and dimensionality reduction techniques,” Expert Syst. Appl., vol. 92, pp. 507–520, Feb. 2018.

[24] H. Zarzour, Z. Al-Sharif, M. Al-Ayyoub, and Y. Jararweh, “A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques,” in Proc. 9th Int. Conf. Inf. Commun. Syst. (ICICS), Apr. 2018, pp. 102–106.

[25] A. K. Uysal and S. Gunal, “The impact of preprocessing on text classification,” Inf. Process. Manage., vol. 50, no. 1, pp. 104–112, Jan. 2014.

[26] Y. Wu et al., “Google’s neural machine translation system: Bridging the gap between human and machine translation,” 2016, arXiv:1609.01844.

[27] E. Loper and S. Bird, “NLTK: The natural language toolkit,” in Proc. ACL Workshop Effective Tools Methodolog. Teach. Natural Lang. Process. Comput. Linguistics, Philadelphia, PA, USA: Association for Computational Linguistics, 2002, pp. 1–8.

[28] P. Willett, “The Porter stemming algorithm: Then and now,” in Proc. ACM Workshop Effective Tools Methodolog. Teach. Natural Lang. Process. Comput. Linguistics, Philadelphia, PA, USA: Association for Computational Linguistics, 2002, pp. 1–8.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.

[30] M. Kantrowitz, B. Mohit, and V. Mittal, “Stemming and its effects on TFIDF ranking (poster session),” in Proc. 23rd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), 2000, pp. 357–359, doi: 10.1145/345508.345650.

[31] P. Willett, “The Porter stemming algorithm: Then and now,” Program, vol. 40, no. 3, pp. 219–223, 2006.

[32] C. Fellbaum, “Wordnet,” in Theory and Applications of Ontology: Computer Applications, Cham, Switzerland: Springer, 2010, pp. 231–243.

[33] E. M. Gale, N. Martin, and J. S. Bowers, “When and where do feed-forward neural networks learn localist representations?” 2018, arXiv:1806.03934.

[34] S. Tuarob, S. W. Kang, P. Wettayakorn, C. Poramprasit, T. Sachati, S.-U. Hassan, and P. Haddaw, “Automatic classification of algorithm citation functions in scientific literature,” IEEE Trans. Knowl. Data Eng., vol. 32, no. 10, pp. 1881–1896, Oct. 2020.

[35] N. Sakib, R. B. Allegre, and K. Haruna, “A collaborative approach toward scientific paper recommendation using citation context,” IEEE Access, vol. 8, pp. 51246–51255, 2020.

[36] S.-U. Hassan, M. Imran, S. Iqbal, N. R. Aljohani, and R. Nawaz, “Deep context of citations using machine-learning models in scholarly full-text articles,” Scientometrics, vol. 117, no. 3, pp. 1645–1662, Dec. 2018.

[37] K. Sugiyama and M.-Y. Kan, “A comprehensive evaluation of scholarly paper recommendation using potential citation papers,” Int. J. Digit. Libraries, vol. 16, pp. 91–109, Jan. 2015.

[38] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl, “On the recommending of citations for research papers,” in Proc. ACM Conf. Comput. Supported Cooperat. Work, 2002, pp. 116–125.

[39] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[40] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[41] S. Teufel, “Argumentative zoning: Information extraction from scientific text,” Ph.D. dissertation, Univ. Edinburgh, Edinburgh, U.K., 1999.

[42] S. Tuarob, S. Bhatia, P. Mitra, and C. L. Giles, “AlgorithmSeer: A system for extracting and searching for algorithms in scholarly big data,” IEEE Trans. Big Data, vol. 2, no. 1, pp. 5–17, Mar. 2016.

[43] C.-H. Chen, S. D. Mayanglambam, F.-Y. Hsu, C.-Y. Lu, H.-M. Lee, and J.-M. Ho, “Novelty paper recommendation using citation authority diffusion,” in Proc. Int. Conf. Technol. Appl. Artif. Intell., Nov. 2011, pp. 126–131.

[44] M. Valenzuela, V. Ha, and O. Etzioni, “Identifying meaningful citations,” in Proc. Workshops 29th AAAI Conf. Artif. Intell., 2015, pp. 1–6.

[45] F. Quyuum and M. T. Afzal, “Identification of important citations by exploiting research articles’ metadata and cue-terms from content,” Scientometrics, vol. 118, no. 1, pp. 21–43, Jan. 2019.

[46] X. Zhu, P. Turney, D. Lemire, and A. Vellino, “Measuring academic influence: Not all citations are equal,” J. Assoc. Inf. Sci. Technol., vol. 66, no. 2, pp. 408–427, 2015.

[47] S. Nazir, M. Asif, S. Ahmad, F. Bukhari, M. T. Afzal, and H. Aljuidae, “Important citation identification by exploiting content and section-wise in-text citation count,” PLoS ONE, vol. 15, no. 3, Mar. 2020, Art. no. e0225885.

[48] C. Jeong, S. Jang, E. Park, and S. Choi, “A context-aware citation recommendation model with bert and graph convolutional networks,” Scientometrics, vol. 124, no. 3, pp. 1907–1922, 2020.

[49] B. L. Nielsen, S. L. Skau, F. Meier, and B. Larsen, “Optimal citation context window sizes for biomedical retrieval,” in Proc. CEUR Workshop, vol. 2345, pp. 191–203, 2017.

[50] A. Vasmanni, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017, arXiv:1706.03762.

[51] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.

[52] F. Zhuang, Z. Qi, K. Duan, D. X. Yi, Z. Zhe, H. Zhe, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” Proc. IEEE, vol. 109, no. 1, pp. 43–76, Jul. 2020.

[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[54] A. Luque, A. Carrasco, A. Martin, and A. de las Heras, “The impact of class imbalance in classification performance metrics based on the binary confusion matrix,” Pattern Recognit., vol. 91, pp. 216–231, Oct. 2019.

[55] S. Ruder, “An overview of gradient descent optimization algorithms,” 2016, arXiv:1609.04747.

[56] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The long-document transformer,” 2020, arXiv:2004.05150.

[57] N. Kitao, L. Kaiser, and A. Levskaya, “Reformer: The efficient transformer,” 2020, arXiv:2001.04451.

[58] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-XL: Attentive language models beyond a fixed-length context,” 2019, arXiv:1901.02860.

[59] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[60] E. Yazan and M. F. Talu, “Comparison of the stochastic gradient descent based optimization techniques,” in Proc. Int. Artif. Intell. Data Process. Symp. (IDAP), Sep. 2017, pp. 1–5.

[61] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” 2018, arXiv:1811.03378.

[62] I. O. Leyer, M. Krzyszowski, and N. Altman, “Classification evaluation,” Nature Methods, vol. 13, no. 8, pp. 1548–7105, 2016.
SHAFIULLAH KHAN received the Ph.D. degree in wireless networks security from Middlesex University, U.K. He is currently with the Institute of Information Technology, Kohat University of Science and Technology, Pakistan, as an Associate Professor. His research interests include wireless broadband network architecture, security and privacy, security threats, and mitigating techniques. He is serving as an editor in many international well-reputed journals.

LISU YU (Member, IEEE) received the B.E. degree from the School of Information Science and Technology, Mao Yisheng Honors College, Southwest Jiaotong University, Chengdu, China, in 2014, and the Ph.D. degree from the Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University, in 2019. He was a Visiting Scholar at the University of Arkansas, Fayetteville, AR, USA, and the University of Houston, Houston, TX, USA, from 2017 to 2019. He is currently a Distinguished Associate Professor with the School of Information Engineering, Nanchang University, China. His main research interests include advanced wireless communications, coded modulation, non-orthogonal multiple access (NOMA), fiber wireless communications, ultra-dense network (UDN), unmanned aerial vehicle (UAV), and visible light communication (VLC). He has served as the Student Activities Chair for the IEEE Communication Society Chengdu Chapter and several international conferences technical program committee (TPC) member, a section chair, and a special track chair. He is now serving as an Academic Editor for the PLOS ONE and the IEEE Communications Society Technical Committee on Green Communications and Computing (TCGCC) and Signal Processing and Computing for Communications (SPCC) Members.

MUHAMMAD ASIF received the M.S. and Ph.D. degrees from AIT, in 2009 and 2012, respectively, on the HEC Foreign Scholarship. He was a Research Scholar with the Computer Science and Information Management Department, Asian Institute of Technology, Thailand. During the course of time, he was a Visiting Researcher with the National Institute of Information Tokyo, Japan. He worked on some projects, including the Air Traffic Control System of Pakistan Air force. He is a Permanent Member of the Punjab Public Service Commission (PPSC) and an Advisor and a Program Evaluator at the National Computing Education Accreditation Council (NCEAC), Islamabad. He has been serving as a reviewer of a number of reputed journals and also authored a number of research papers in reputed journals and conferences. He has been serving as an Associate Editor for IEEE Access and the prestigious journal of the IEEE.

YAZEED YASIN GHADI received the Ph.D. degree in electrical and computer engineering from Queensland University. He is currently an Assistant Professor of software engineering with Al Ain University. He was a Postdoctoral Researcher with Queensland University, before joining Al Ain University. He has published more than 25 peer-reviewed journals and conference papers and holds three pending patents. His current research interests include developing novel electro-acoustic-optic neural interfaces for large-scale high-resolution electrophysiology and distributed optogenetic stimulation. He was a recipient of several awards. His dissertation on developing novel hybrid plasmonic photonic on-chip biochemical sensors received the Sigma Xi Best Ph.D. Thesis Award.

* * *