IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 1, 2020, accepted November 1, 2020, date of publication November 6, 2020, date of current version November 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3035886

Additive Angular Margin Loss in Deep Graph Neural Network Classifier for Learning Graph Edit Distance

NADEEM IQBAL KAJLA[1], MALIK MUHAMMAD SAAD MISSEN[1], MUHAMMAD MUZZAMIL LUQMAN[2], MICKAEL COUSTATY[2], (Member, IEEE), ARIF MEHMOOD[1], AND GYU SANG CHOI[3]

[1]DCS & IT, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[2]Laboratoire Informatique, Image et Interaction (L3i), La Rochelle University, 17031 La Rochelle, France
[3]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 712-749, South Korea

Corresponding authors: Malik Muhammad Saad Missen (saad.missen@iub.edu.pk) and Gyu Sang Choi (castchoi@ynu.ac.kr)

This work was supported in part by the LabCom IDEAS under Grant ANR-18-LCV3-0008; in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1A2C1006159) and MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2016-0-00313) supervised by the IITP (Institute for Information & communications Technology Promotion).

ABSTRACT The recent success of graph neural networks (GNNs) in the area of pattern recognition (PR) has increased the interest of researchers to use these frameworks in non-euclidean structures. This non-euclidean structure includes graphs or manifolds that are called geometric deep learning (GDL). It has opened a new direction for researchers to deal with graphs using deep learning in document processing, outperforming conventional methods. We propose a Deep Graph Neural Network (DGNN) classifier-based on additive angular margin loss for the classification task in document analysis. Another contribution of this work is to investigate the performance of a DGNN as a classifier using different loss functions, which helps to minimize the loss for the document analysis problem. We compare additive angular margin loss, Cosine angular margin loss, and multiplicative angular margin loss. Furthermore, we give a comparison between the mentioned loss functions and the Softmax loss function. We also present the comparisons of results using different graph edit distance (GED) methods. Our quantitative results suggest, that by applying the additive angular marginal loss function makes more compact intra-class ability and increases the inter-class discrepancy which enhances the discriminating power of the DGNN. Enhancing the decision boundaries between the classes increase the intra-class compactness and inter-class discrimination power of the model.

INDEX TERMS Graph neural networks (GNN), graph learning, graph edit distance (GED), geometric deep learning (GDL), loss margin, loss function, neural networks.

I. INTRODUCTION

Graphs are used in wide application domains like protein structures [1]–[3], social networks [4], [5], document analysis [6], [7] etc. The handwritten character recognition problem in the document analysis domain is solved using various methods and techniques. Some researchers solve this problem using Graph Edit Distance (GED) [8]–[11] some solve this problem using Graph Embedding (GE) [7], [12]–[14] and now a recent work shows that researchers are using [15], to solve this problem. In Convolutional Neural Networks, an input vector is given to the network to learn a feature, and then predictions are given to the loss function. Figure 1 shows a typical framework of a Convolutional Neural Network (CNN).

Minimizing the loss function is the ultimate goal of any Neural Network (NN) architecture. It is always a challenging task to design a loss function that can efficiently be used in NN architecture to solve a given task. The loss function is used to calculate the loss by matching the labels predicted by the Neural Network with the actual labels. Triplet loss [16] based functions and the Softmax loss [17] based functions

The associate editor coordinating the review of this manuscript and approving it for publication was Tu Ngoc Nguyen.

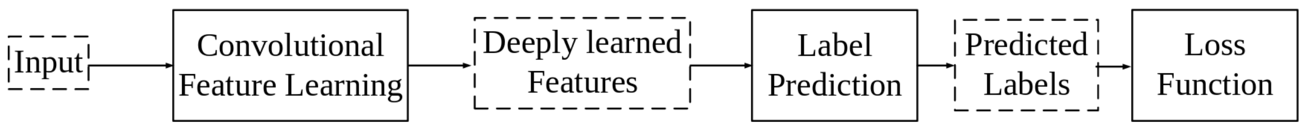

FIGURE 1. Framework of a typical convolutional Neural Network.

both have great performance and provide very good discriminating ability but recently proposed loss functions like Additive Angular Marginal Loss (ArcFace) [18] which is proposed to the face recognition problem also aimed to perform well in symbol recognition and the Handwritten Recognition. A large number of proposed work has been published as a variation of Softmax [17]. Wen *et al.* [19] proposed a center loss function that learns a center for each class and defines a distance for the feature to the corresponding class and the deep feature. Chen et. al. [20], Wan *et al.* [21], Qi *et al.* [22] and many others present the variations of Softmax [17] loss functions to improve the separability and discriminative power of a model.

In document analysis like a handwritten character recognition or symbol recognition, the learned feature not only needs to be separable but should be discriminative as well. So, the deep feature learned from a Densely Connected Neural Network (DCNN) is required to be discriminative and generalized for the unseen classes without label predictions. The Discriminative power of any loss function helps it to characterize in a separable inter-class difference and the compact intra-class variation as well [18]. It is non-trivial to construct an efficient loss function for a discriminative feature in a CNN because CNN is normally based on the mini-batch training by the Stochastic Gradient Descent optimizer which performs well in reflecting the global distribution of deep features [19]. As the training examples are normally huge in size, it is impractical to pass all the training examples as input in every iteration. To overcome this, the triplet loss [16] and the Softmax loss [23], [24] calculate the loss by constructing the loss functions for triplets or image pairs [18]. However, it dramatically grows the triplets or number of training pairs by comparing the image samples, and hence it results in instability of the model and slow convergence [21]. This can be handled by carefully selecting the triplets or image pairs but it also results in an increase in the computational complexity which can result in inconvenient training procedures [18].

In this paper, we present a Deep Graph Neural Network (DGNN) Classifier for learning graph distances and present the results with various famous loss functions and their effect in the document analysis domain. We also present a comparison of graph matching performance of GED. We apply this DGNN approach on the topic of graphical elements recognition in documents. While many approaches tends to recognize them using some statistical approaches, we assume that representing them with a graph will help in retrieving similar structures. We would also like to say that using structural approaches helps in understand how the system has proceed and then reduce the semantic gap between the machine and the end-user. The rest of the paper is organized as follows. In section II we give an overview of the related work. We compare traditional Graph Edit Distance methods with Graph Neural Networks. We discuss the importance of loss functions in training neural networks and give an overview of the loss functions used in the experiments. We conclude section II with a problem statement and short details of our contribution. In section III we propose a framework to robustly train Deep Graph Neural Network for the document analysis problem. In section IV we give information about the experimental setup and the dataset used in our experiments. We provide an extensive analysis of the benefits of the proposed framework on the letters dataset in section V. We also discuss the results and provide quantitative insights. We conclude our work with a conclusion where we suggest future directions for our work.

II. RELATED WORK

In this section, we present the state of the art work about Graph Edit Distance (GED), Graph Neural Networks (GNNs), and loss functions. It follows the research gap in the literature review, the problem statement, and our contribution.

A. GRAPH EDIT DISTANCE (GED)

Graphs have a wide variety of application domains. Due to very extensive use in various applications large efforts have been done to improve the graph-based methods and techniques and to make the use of graphs more effective and efficient in every application domain such as graph learning, graph matching, graph mining, graph similarity, etc. Computation graph similarity is a major and core field of research. Graph similarity is computed with GED [25]. GED calculates the minimum cost of an edit path between two graphs [25]. An edit path can be defined as a sequence of edit operations like deleting, inserting, and relabelling of edges. An example of possible edit paths is shown in figure 2 to transform one graph into another. GED computation problem is an NP-hard problem, especially it is hard when graphs have a large number of vertices and/or edges [26].

The computational complexity of the nodes increases exponentially deep in the graph involved [15], which also leads to higher FLOPs for the model. Many approximations to GED have been suggested because of the statistical complexity. The computation of GED is normally solved with tree-based search algorithms [27]. In this method, all

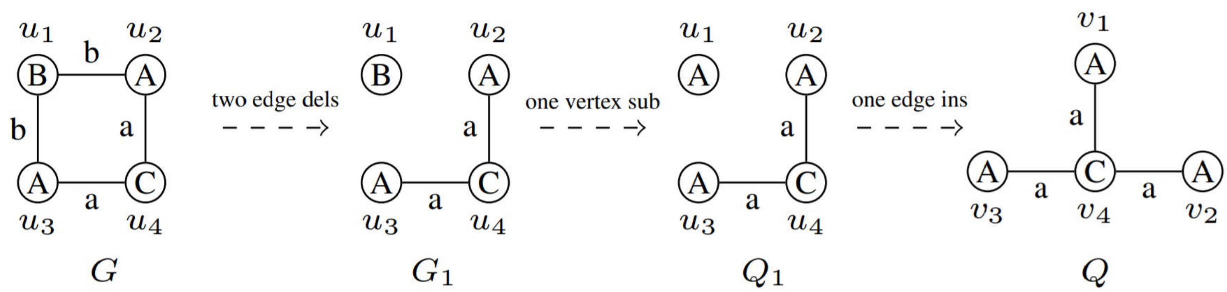

FIGURE 2. An edit path P between Graph G and Q Chen *et al.* [25].

possible mappings of edges and vertices of compared graphs are explored [9]. Based on how the method is generating the node's successors we can divide these graph edit distance methods into two categories: *the edge based* methods and *the vertex based* methods [25]. A* graph edit distance method proposed by Riesen and Bunke [28] and DF graph edit distance proposed by Abu-Aisheh et.al. [27] are the examples of vertex-based graph edit distance methods. Method proposed by Gouda and Hassaan [29] is an edge-based graph edit distance method. It is based on common substructure isomorphism which performs well for sparse graphs. Due to the high computational complexity of GED, several approximate methods also proposed by researchers. Riesen et.al. [9] proposed a Bipartite graph matching algorithm to solve the assignment problem using edit operations like insertion, deletion, etc. Fischer et.al. [11] proposed another approximate Graph Edit Distance method. It provides lower bound of the original GED. Another method proposed in [30] provides variant solutions for the efficient and accurate approximation of GED, but obtaining better information about the edges and nodes within the graph is still an open issue [15].

In this work, we propose a method to learn a graph distance between graphs based on Graph Neural Network and local descriptions of a graph. The proposed method is based on calculating Euclidean distance by collecting the information of nodes and edges of graphs and can directly discriminate between two classes rather than computing an exact graph edit distance.

B. GRAPH NEURAL NETWORKS (GNN)

Graphs have less mathematical structure and high computational complexity in their domain comparing with feature vectors [31]. Due to high computational complexity in many cases, the graph-based techniques are less famous in different applications of pattern recognition [32]. Issues of recognition can be formulated as an assigned graph matching issue. The extraction of signatures in [15] node is combined with an optimal assignment method for matching the assigned graphs. In particular, they show how local descriptions used to define the node cost of an assignment issue using the Hungarian method; Also, they have proposed a distance formula to calculate the distance between assigned graphs. However, with the new enhancements made in parallel and deep learning fronts, the graph-based techniques are more likely to be applicable in many fields of pattern recognition [33]. The deep learning-based methods when applied to graphs are known as graph neural networks. [34]. The graph neural networks are widely used in different domains due to its convincing performance and interpretability [35]. Convolutional neural networks (CNNs) can be considered as the first motivation for the GNNs, as CNNs have the ability to extract multi-scale localized spatial features and compose them to construct highly expressive representation [34]. CNN has been extended to graphs via Graph Convolutional Networks (GCN), but these models will most likely underperform on highly regular graphs, which is often the case when graphs represent real-life content (like maps, textual documents, natural scene images) as graphs are the results of pre-processing systems which tend to denoise the original content. Solving problems with GNNs in different domains are interesting because of the key points:

- Graphs are locally connected structures.
- In GNNs shared weights are used to reduce the computational costs as compared to the traditional spectral graphs [36].
- Various feature sizes can be captured with multi-layer structures to deal with the hierarchical patterns.

Although GNNs have great success in various applications, GNN fails to provide good performance in the following conditions [43].

- Unlike CNN and RNNs, due to the shallow structure of GNN it has an issue of over-smoothing. However, some study proposes to design deep GNN to tackle this problem which is a challenging issue,
- GNN suffers from poor additivity in dynamic graphs, but dynamic GNN is a possible solution which is under active research,
- In GNN, still there is no optimal methods to generate graphs from raw data,
- GNN has limited scalability in web-scale applications like social networks and recommendation systems, and it computational expensive during its scaling up.

GNNs are able to model the relationship between the nodes in a graph and to produce a numeric representation of it. GNNs are getting more and more significant as so many real-world data that can be represented as a graph (social networks, chemical compounds, maps, natural scene images, and the textual documents). [15] presents a Message Passing Neural Network (MPNN) to learn enriched graph representations. The work provides significant results in hand written character recognition and keyword spotting application of document analysis.

Our major focus, in this paper, is to present a classifier based on GED to measure the similarity between two graphs. It can capture the structural information of nodes and edges of different graphs and can learn a metric among two graphs to discriminate two classes.

C. LOSS FUNCTIONS

Loss function plays a vital role in any Neural Network. The Loss function is used to minimize the loss in a Neural Network. The loss function is used to calculate the loss by matching the labels predicted by the Neural Network with the actual labels [16]. Triplet Loss proposed by Schroff *et al.* [16] and the Softmax Loss proposed by Bridle [17] are widely used loss functions in Neural Network frameworks. Many loss functions are proposed by the researchers in recent years based on the Triplet loss and Softmax loss to increase the discriminating power of a model.

SphereFace loss [37] used the idea of angular margin for improving the decision margin between the classes. CosFace Loss [38], [39] use cosine margin to the target logit to get the good performance as compared to the SphereFace loss. Recently proposed ArcFace loss by Deng *et al.* [18] used Additive Angular Margin Loss to improve the performance in terms of discrimination between the classes. These loss functions are efficient and have more advantages of engaging, effectiveness, and easiness. Comparing to the Triplet loss and Softmax loss, these loss functions add less computational complexity during the training phase. These loss functions are proposed specifically for the domain of Face recognition. Keeping in mind the advantages of these loss functions over the traditional loss functions, we implement these in the domain of document analysis. In this work, we applied them in hand written character recognition domain and compared the results of the proposed Deep Graph Neural Network. We provide comparisons of all these loss functions with Softmax loss in this article.

D. PROBLEM STATEMENT AND OUR CONTRIBUTION

In the work of Riba *et al.* [15], the authors proposed a Message Passing Neural Network (MPNN). More specifically, they proposed a simple but effective metric based on the Hausdorff distance [11] in which the edges are not taken into account because of the local structure used by HED has been embedded during the passing phase of the message. We proposed a DGNN that learns a feature vector by collecting the information from nodes as well as from edges. we give the feature vector, containing information about the edges and nodes, to the Neural Network. This information on the edges of a graph is useful as it helps to improve the discriminatory power of the feature vector. By proposing a mechanism to take this information into account (and demonstrating its effectiveness by experimental results), our proposed work contributes to the body of knowledge on GNNs.

As graphs are sensitive to noise, they can misclassify the samples which are very close to the decision boundary of the classifier. So, to enhance the discriminating power of the model we use the Additive Angular Margin loss in our model. We also use the multiplicative angular margin loss and cosine margin loss to see the difference among these three angular margin losses. Moreover, in the work of [15] calculating the graph distances edges are not taken into account. So, by taking the information of edges alongside the nodes we can train our model more robustly.

III. PROPOSED FRAMEWORK

We propose a Deep Graph Neural Network (DGNN) architecture based on Riba *et al.* [15] work where a Neural Network architecture was proposed to learn a graph distance based on Hausdorff Edit Distance (HED) [11]. The Hausdorff matching provides a lower bound of order O(n1 · n2) of the original graph edit distance [15]. The main drawback of their approach is that it relies on the fact that the Edges are not taken into account as the HED embeds local structure during the message passing phase of the Neural Network, where a neural network architecture was proposed to learn Hausdorff Edit Distance (HED) proposed by Fischer *et al.* [15]. We then propose to gather local descriptions of edges alongside nodes. To this end, we propose to use the Heterogeneous Euclidean Overlap Metric Distance (HEOMD) originally proposed by Jouili *et al.* [10]. The extracted node signatures provide the local structural description of the graph. The node signature can be a spectrum of different attribute types, including numeric and symbolic data, requiring more complex metrics. Therefore, they use the Heterogeneous Euclidean Overlap Metric (HEOM) to compute the distance between two node signatures. They proved that the proposed method performs for node-to-node correspondences between two graphs, and provides excellent results to retrieve a different kind of images represented by attributed graphs when compared with the Umeyama method for incorrect graph matching and the Zass probabilistic method [10]. We gather the symbolic and numeric attributes of edges and nodes of the graphs to calculate the distance between two graphs with formula in equation 1:

$$Dist(g_i, g_j) = \frac{\hat{M}}{|M|} + \left| |g_i| - |g_j| \right| \tag{1}$$

where $|M|$ is the number of matching operations, and $\hat{M}$ is the matching cost. This matching cost is calculated by taking the sum of all matching operations costs. So, while calculating the distance, we compute the information about numeric and symbolic attributes of local nodes and edges. The distance represents the matching cost which is normalized by the matching size and is increased by the different sizes of graphs.

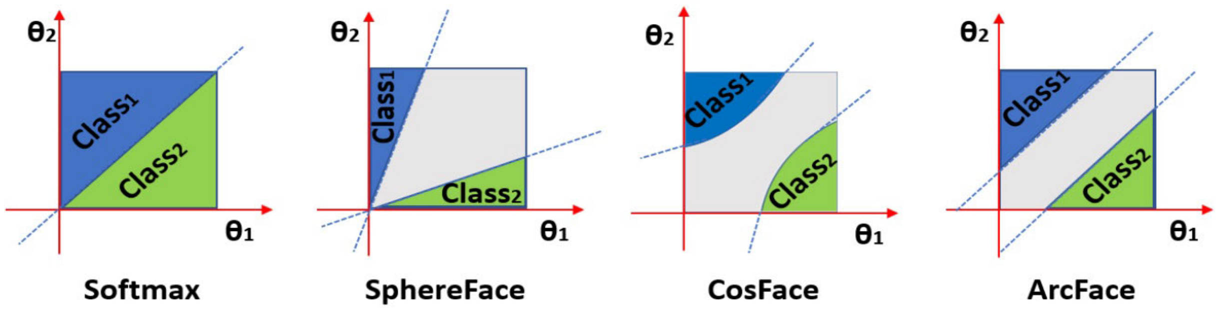

FIGURE 3. Decision boundaries for binary classification of loss functions using Multiplicative margin, Cosine margin and additive margin [18].

We also propose to use the Additive Angular Margin (ArcFace) [18] loss function in the model. This ArcFace loss uses the cosine distance to calculate the angle among the target weight and the current feature. An additive angular margin is added to the target angle and then cosine function is applied again to get the target logit back. All logits are re-scaled again by a fixed feature norm. We also use standard Softmax loss Function shown in Equation 2 and compare the results with the ArcFace loss [18].

$$L_{SM} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{W_{y_i}^T} x_i + b_{yi}}{\sum_{j=1}^{n} e^{W_j^T} x_i + b_j} \tag{2}$$

Our proposed approach adds the Additive Angular Margin m between x_i and W_{y_i} to make a more compact intraclass ability and increase the inter-class discrepancy. So, we applied this Additive Angular Margin loss by the Equation 3 as proposed by Deng *et al.* [18]: which uses Additive Angular Margin loss to directly optimize the geometric distance for the correspondence among the arc and the angle in a hypersphere.

$$L_{AAM} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(cos(\theta_{y_i}+m))+\sum_{j=1,j\neq y_i}^{n} e^{s(cos\theta_j)}}} \tag{3}$$

We applied the additive cosine margin loss function of our Deep Graph Neural Network model as proposed by Wang *et al.* [38]. The cosine angular margin loss function has more intrinsic consistency with a standard softmax loss function. With the formulation of cosine angular margin, it matches the similarity measurement which is frequently used by recognition of patterns. It is very effective in improving the inter-class cosine angular margin discriminative information. The equation 4 is used to calculate the cosine margin loss function.

$$L_{CM} = \frac{1}{N}\sum_{i=1}^{N} -log \frac{e^{s(cos(\theta_y,i)-m)}}{e^{s(cos(\theta_y,i)-m)+\sum_{j=1,j\neq y_i}^{n} e^{s(cos\theta_j,i)}}} \tag{4}$$

We also implement the multiplicative angular margin loss function as proposed by Liu *et al.* [37] and compare the results with the standard Softmax loss [17]. Sphere face loss is calculated by equation 5. This loss function uses a multiplicative angular margin loss.

$$L_{sphere} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{||x_i||\ \psi(\theta_{y_i},i)}}{e^{||x_i||\ \psi(\theta_{y_i},i)}+\sum_{j\neq y_i} e^{||x_i||\ \cos(\theta j,i)}} \tag{5}$$

Numerically, the marginal loss functions when added to the angle or added to the cosine space enforce the inter-class discrepancy and it also ensures more intra-class compactness.

Loss function which uses the Additive Angular Margin enforce more gap between the classes. It has more discriminating power than the multiplicative and cosine margin loss functions. Figure 3 shows the geometric distance between the loss functions consists of Additive angular margin, Multiplicative angular margin, and cosine margin. The additive angular margin has a better geometric attribute as the angular margin corresponds precisely to the geodesic distance [18]. The proposed ArcFace has a constant linear, angular margin throughout the entire interval. On the other hand, SphereFace and CosFace have a nonlinear angular margin only. The slight difference in margin designs can have a "butterfly effect" on model training [18].

IV. EXPERIMENTAL DETAILS

In this section, we will describe the dataset used in our implementations following the experimental setup to train the model.

A. DATASET

We used the Letters Dataset [8] from IAm Graph repository. This dataset is proposed by Riesen *et al.* [8] in 2008,

This becomes a defacto standard for almost every Graph Based technique since then, particularly the graph based methods for hand written character recognition. The dataset consists of three categories as *LOW, MED*, and *HIGH* based on the distortion. The LOW category has less distortion in the graphs of letters as compared to the MED and HIGH. Similarly, the HIGH category has more distorted graphs such that some characters are even not readable by humans, as shown in Figure 4. The dataset is further divided into train, test, and validation sets. Each set is consists of 750 graphs. So, 2250 graphs are in each category and in total 6750 graphs. As shown in Table 1 it consists of 15 classes A, E, F, H, I, K,

TABLE 1. Letters Dataset Details for each category LOW, MED, HIGH [8].

Dataset	Sub category	No. of Graphs	No of classes	Average Vertex	Average Edge
Letters	LOW	2250 (750,750,750)	15	4.676	3.132
	MED	2250 (750,750,750)	15	4.675	3.206
	HIGH	2250 (750,750,750)	15	4.670	4.500

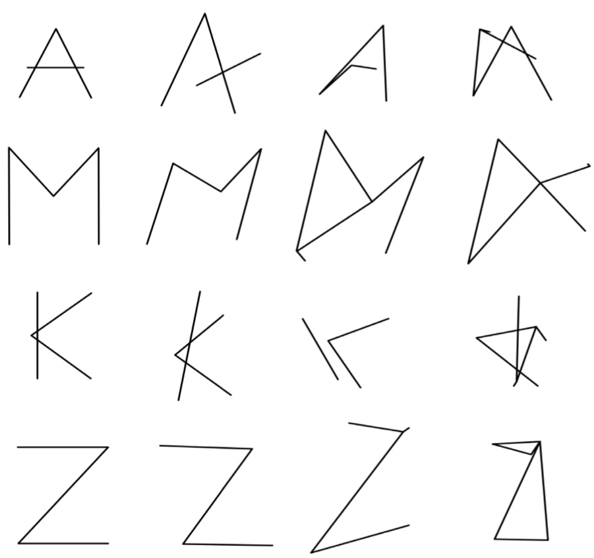

FIGURE 4. Some examples of letters with different level of distortions(low/medium/high from left to right) from Letters Dataset [8].

L, M, N, T, V, W, X, Y, Z in each subset of LOW, MED, and HIGH.

B. EXPERIMENTAL SETUP

We trained our Neural Network classifier based on *K*-NN. The training process ran for 1000 epochs. Stochastic Gradient Descent is used as an iterative method for optimizing the activation function. Weight *Decay* is used to update the Neural Network's weights after every epoch. This Decay optimizer is multiplied by a value slightly less than 1. This helps the weight to prevent growing too large. 0.9 value of momentum is used to give momentum in the convergence of the model. We used the Gamma hyper-parameter and multiplied it with the learning rate after every 150 epochs. The value of gamma is set to 0.1. The main focus of our experiments is to train a Deep Graph Neural Network classifier based on the Additive Angular Margin Loss. We train our classifier using four loss functions. We compare our results with six different states of the art methods i.e. HED (2015), SoftHd MPNN (2018), MPNN (2017), FMGE with SVM (2013), GRALGv1 (2014), and Learning cost Function for Graph Matching using HEOMD (2018) The fairness of our results is ensured by applying four different loss functions to the proposed methods and compare the results with three subsets of Letters dataset: LOW, MED, HIGH. Experiments were performed in the same conditions for all methods which guarantee the fairness of the results. None of the methods were optimized to have an explicit advantage over other methods. Algorithm 1 shows the complete pseudo code of our proposed model.

Algorithm 1 The Pseudo-Code of Proposed Method

PROGRAM trainModel
INPUT graphs g_i, g_j and $L(0, 1)$ is a label indicating positive or negative pairs of graphs
Initialization: G_w, A Deep graph neural network(DGNN)
1: Obtain the attributed graphs g' for the input graphs:
2: $g'_i = G_w(g_i)$
3: $g'_j = G_w(g_j)$
4: For the two attributed graphs, calculate the matching cost, $\hat{M}$ which is the sum of all matching operation costs
  FOR t in T time steps for each DGNN
    call: update
    call: message
  END FOR
5: Calculate Proposed distance between g_i, g_j'using symbolic and numeric attributes of edges and nodes $D = \frac{\hat{M}}{|M|} + ||g'_1| - |g'_2||$
6: Calculating objective function by $L_{AAM} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(cos(\theta_{y_i}+m))+\sum_{j=1,j\neq y_i}^{n} e^{s(cos\theta_j)}}}$
OUTPUT: Positive or negative pairs
END

V. RESULTS AND DISCUSSION

The main focus of our experiment is to train a Deep Graph Neural Network Classifier Based on Additive Angular Margin Loss. We train our classifier with the four margin loss functions.

Our approach contributes towards the document analysis problem; we used a deep neural network on the letters dataset, which is categorized into three categories (LOW, MED, and HIGH) and evaluated the performance of the model on the bases of four different loss functions. SphereFace Loss Function used the idea of angular margin in loss function. CosFace loss uses cosine margin in the Loss Function. ArcFace loss by Deng *et al.* [18] used Additive Angular Margin in Loss function to improve the performance in terms of discrimination between the classes. We provide comparisons of all these Loss Functions with SoftMax Loss in our proposed approach using DGNN. Results described in Table 2 show that if marginal loss function is applied then it will provide better results as compared to the standard Softmax loss function. Figure 5 shows the results of LOW subset from Letter dataset [8]. The proposed approach using the Additive

TABLE 2. Distance computation (percentage) of different methods using different Loss functions on Letters Dataset with standard deviation for 6 runs.

Distance Method	Loss Function	LOW	MED	HIGH
Hausdorff Edit Distance *(2015)* **[11]**	Softmax Loss	93.13,±0.498	84.02, ±0.707	68.12, ±0.604
	SphereFace Loss	93.66, ±0.286	84.79, ±0.842	66.76, ±0.867
	CosFace Loss	94.67, ±0.596	84.57, ±0.598	72.57, ±0.656
	ArcFace Loss	95.66, ±0.583	85.79, ±0.545	72.76, ±0.679
SoftHd *(2018)* **[15]**	Softmax Loss	95.56, ±0.615	85.35, ±0.528	71.75, ±0.724
	SphereFace Loss	95.73, ±0.725	85.86, ±0.873	71.29, ±0.942
	CosFace Loss	96.14, ±0.321	87.42, ±0.338	73.91, ±0.407
	ArcFace Loss	96.98, ±0.568	89.86, ±0.638	74.75, ±0.687
MPNN *(2017)* **[40]**	–	94.80, ±0.68	86.10, ±1.81	75.70, ±1.95
FMGE with SVM *(2013)* **[7]**	–	98.20	83.10	70.00
learning cost function using HEOMD *(2018)* **[41]**	–	44.80, ±5.94	34.13 ±9.78	29.07, ±4.36
GRALGv1 *(2014)* **[42]**	–	98.20	79.80	74.50
Proposed DGNN Using AAM Loss with HEOMD	Softmax Loss	93.12,±0.421	84.27, ±0.612	72.57, ±0.738
	SphereFace Loss	95.13, ±0.638	85.01, ±0.553	76.86, ±0.806
	CosFace Loss	96.73, ±0.276	88.74, ±0.651	78.85, ±0.492
	ArcFace Loss	**98.87**, ±0.159	**90.57**, ±0.429	**83.34**, ±0.352

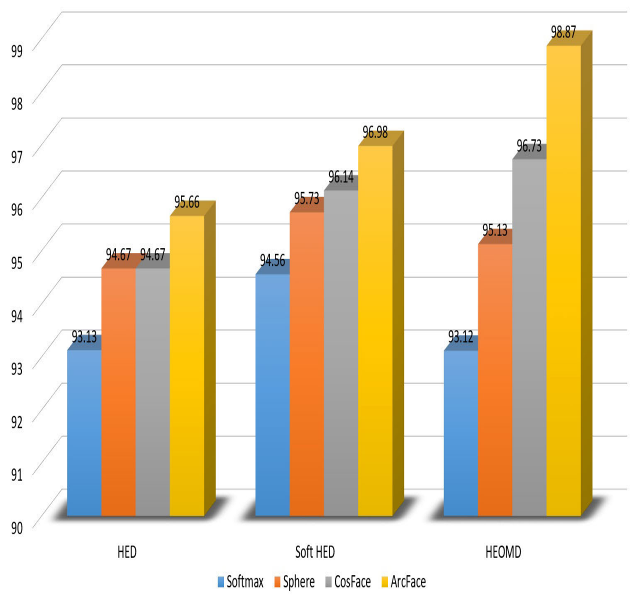

FIGURE 5. Graph Distance computation HD, Soft and HEOMD for Letters LOW Dataset using different loss Functions.

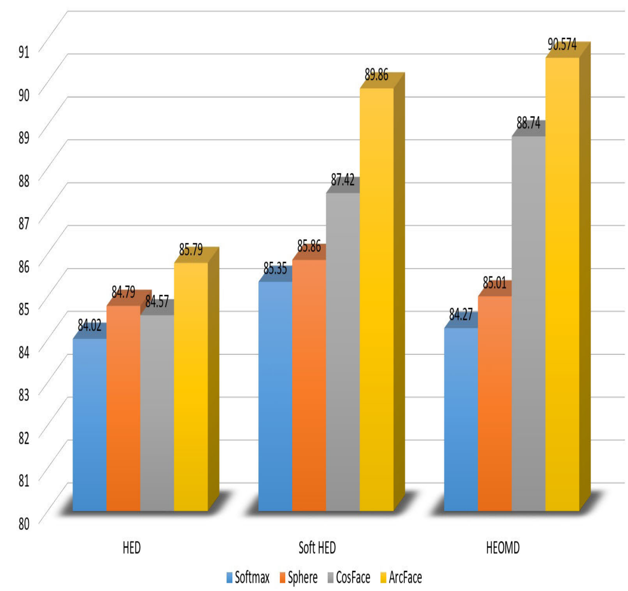

FIGURE 6. Graph Distance computation HD, Soft and HEOMD for Letters MED Dataset using different loss Functions.

Angular Margin loss function in Deep Graph Neural Network outperforms many recent states of the art methods that show the importance of symbolic and numeric information of edges along with the significance of the Additive angular margin loss function. Using the cosine margin loss function also performs well as compared to the multiplicative loss function and the standard Softmax loss function.

Figures 6 and 7 show the results comparisons of MED and a HIGH subset of the letter dataset respectively. This shows that how the discriminating power of a model can be increased by applying the additive angular marginal loss function. It helps to increase the decision boundary between the classes. Although there is numerical similarity. We test the proposed model and compare the results of the four-loss functions on the Letters dataset using different (GED) Graph edit methods. From the results of Table 2, the Hausdorff Edit Distance (HED) method of computing graph distance performs better when the ArcFace is used in the DGNN. The loss function is applied to all three categories (LOW, MED, HIGH) of the Letters dataset, and we can say that Standard SoftMax Loss Function performs least effectively on three categories of data using HED distance metrics. Using the HEOMD distance metric ArcFace Loss Function outperforms all three categories of data compared to all other loss Function. While comparing the results of these recent state of the art methods, we proposed that the HEOMD distance metric provides significant results using the ArcFace Loss Function. Compared to the Hausdorff Edit Distance (HED)

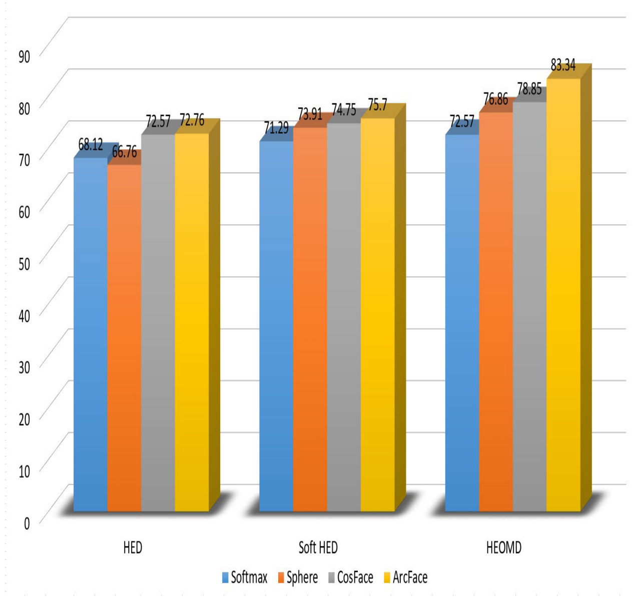

FIGURE 7. **Graph Distance computation HD, Soft and HEOMD for Letters HIGH Dataset using different loss Functions.**

method as the method suffers a drawback i.e., the Edges are not considered as the HED embeds local structure during the message passing phase of the Neural Network[10,15]. So, this drawback can be overcome by the proposed method. This method considers the local structure of graphs edges alongside nodes. Table 2 presents that calculating distance with HEOMD without using additive angular margin loss provides a very poor recognition rate [41]. Using the symbolic and numeric attributes of edges and nodes of the graphs along with the Additive angular margin loss function. Results show the significance of our proposed method. Among the three marginal loss functions, the Additive Angular marginal loss function has a better geometric attribute. Results show that the Additive angular margin loss optimizes the geodesic margin directly with the correspondence among the angle and arc in the hypersphere. This additive angular loss has little effect on the computational complexity during training. Results show that by considering the edges information along with nodes in a DGNN and then using the Additive Angular Marginal loss function, we can improve the decision margins in different classes. This improves the inter-class margin and also increases the intra-class compactness which helps the model in performing better and improves the recognition rate.

GNNs handle complex structures and preserving global information more efficiently compared to other algorithms. Our method allows us to calculate the discriminative features of two classes without explicitly calculating graph edit distance which has high computational complexity. Therefore, GNN is a suitable choice for our task. Compared to traditional graph methods we can benefit from shared weights in GNNs to reduce the computational complexity. Representations learned by GNNs allow us to capture the structural invariance among different writer styles and the inherent stroke variability of handwriting which is computationally costly to achieve with traditional GED methods. Computation complexity is a matter of concern in deep learning. However, the computational complexity of our method during training is negligible. Current GPUs can easily support millions of identities for training and the model parallel strategy can easily support many more identities. Furthermore, here we have outlined our current work in improving the performance of classification and suggesting a more suitable loss function in the domain of hand written character recognition. During training, our total batch number is 1000 and the average time per batch is 0.122. so total execution/training time is $1000*0.122 = 122$. The test time of our model for calculating distance average execution time during the test using k-NN is 0.327.

In this work, We compare our results with six different states of the art methods, and the fairness of our results is ensured by applying four different loss functions to the proposed methods and compare the results with three subsets of the Letters dataset: LOW, MED, HIGH. Experiments were performed in the same conditions for all methods which guarantee the fairness of the results. None of the methods were optimized to have an explicit advantage over other methods.

VI. CONCLUSION

In this work, we proposed a deep graph neural network-based on Additive marginal loss which outperforms the recent state of the art methods in the document analysis problem. We showed that collecting information from nodes and edges, and then using that information in DGNN with the marginal loss function instead of standard Softmax loss results in better recognition rate. Our experimental results show that the cosine margin loss function also performs well as compared to the other loss functions. Empirical results suggest that enhancing the decision boundary between the classes increase the intra-class compactness and inter-class discrimination power of the model.

In this work, we focus on collecting local information of edges and nodes and use the additive angular loss functions on document analysis problem, and our results suggest that fine-tuning the loss function can dramatically increase the performance of the neural network architectures. Our further work is to test the proposed model and compare the results on the other domains using different (GED) graph edit methods and to validate our method with more application. Therefore, this finding could be useful to investigate the results obtained with Additive Margin loss to other domains such as chemical molecules, medical imaging, RNA sequence prediction tasks. In the future, we will apply our method in diffusion-weighted images (i.e. MR imaging for tumor characterization) to investigate the preservation of rotational-invariant features of the image.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

[1] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1263–1272.
[2] P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert, "Graph kernels for molecular structure–activity relationship analysis with support vector machines," *J. Chem. Inf. Model.*, vol. 45, no. 4, pp. 939–951, 2005.
[3] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, Jun. 2018, pp. 593–607.
[4] R. Iijima and Y. Kamada, "Social distance and network structures," *Theor. Econ.*, vol. 12, no. 2, pp. 655–689, May 2017.
[5] C. S. Q. Siew, D. U. Wulff, N. M. Beckage, and Y. N. Kenett, "Cognitive network science: A review of research on cognition through the lens of network representations, processes, and dynamics," *Complexity*, vol. 2019, pp. 1–24, Jun. 2019.
[6] D. Zhou, S. Zhu, K. Yu, X. Song, B. L. Tseng, H. Zha, and C. L. Giles, "Learning multiple graphs for document recommendations," in *Proc. 17th Int. Conf. World Wide Web - WWW*, 2008, pp. 141–150.
[7] M. M. Luqman, J.-Y. Ramel, J. Lladós, and T. Brouard, "Fuzzy multi-level graph embedding," *Pattern Recognit.*, vol. 46, no. 2, pp. 551–565, Feb. 2013.
[8] K. Riesen and H. Bunke, "IAM graph database repository for graph based pattern recognition and machine learning," in *Proc. Joint IAPR Int. Workshops Stat. Techn. Pattern Recognit. (SPR) Struct. Syntactic Pattern Recognit. (SSPR)*. Berlin, Germany: Springer, Dec. 2008, pp. 287–297.
[9] K. Riesen and H. Bunke, "Approximate graph edit distance computation by means of bipartite graph matching," *Image Vis. Comput.*, vol. 27, no. 7, pp. 950–959, Jun. 2009.
[10] S. Jouili, I. Mili, and S. Tabbone, "Attributed graph matching using local descriptions," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.* Berlin, Germany: Springer, Sep. 2009, pp. 89–99.
[11] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen, and H. Bunke, "Approximation of graph edit distance based on hausdorff matching," *Pattern Recognit.*, vol. 48, no. 2, pp. 331–343, Feb. 2015.
[12] K. Riesen and H. Bunke, "Reducing the dimensionality of dissimilarity space embedding graph kernels," *Eng. Appl. Artif. Intell.*, vol. 22, no. 1, pp. 48–56, Feb. 2009.
[13] H. Bunke and K. Riesen, "Recent advances in graph-based pattern recognition with applications in document analysis," *Pattern Recognit.*, vol. 44, no. 5, pp. 1057–1067, May 2011.
[14] A. Dutta, P. Riba, J. Llados, and A. Fornes, "Pyramidal stochastic graphlet embedding for document pattern classification," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 33–38.
[15] P. Riba, A. Fischer, J. Llados, and A. Fornes, "Learning graph distances with message passing neural networks," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 2239–2244.
[16] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[17] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*. Berlin, Germany: Springer, 1990, pp. 227–236.
[18] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.
[19] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 499–515.
[20] B. Chen, W. Deng, and J. Du, "Noisy softmax: Improving the generalization ability of DCNN via postponing the early softmax saturation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5372–5381.
[21] W. Wan, Y. Zhong, T. Li, and J. Chen, "Rethinking feature distribution for loss functions in image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9117–9126.
[22] X. Qi and L. Zhang, "Face recognition via centralized coordinate learning," 2018, *arXiv:1801.05678*. [Online]. Available: http://arxiv.org/abs/1801.05678
[23] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.
[24] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1735–1742.
[25] X. Chen, H. Huo, J. Huan, and J. S. Vitter, "An efficient algorithm for graph edit distance computation," *Knowl.-Based Syst.*, vol. 163, pp. 762–775, Jan. 2019.
[26] Z. Zeng, A. K. H. Tung, J. Wang, J. Feng, and L. Zhou, "Comparing stars: On approximating graph edit distance," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 25–36, Aug. 2009.
[27] Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, and P. Martineau, "An exact graph edit distance algorithm for solving pattern recognition problems," in *Proc. Int. Conf. Pattern Recognit. Appl. Methods*, Jan. 2015.
[28] K. Riesen, S. Emmenegger, and H. Bunke, "A novel software toolkit for graph edit distance computation," in *Proc. Int. Workshop Graph-Based Represent. Pattern Recognit.* Berlin, Germany: Springer, May 2013, pp. 142–151.
[29] K. Gouda and M. Hassaan, "CSI_GED: An efficient approach for graph edit similarity computation," in *Proc. IEEE 32nd Int. Conf. Data Eng. (ICDE)*, May 2016, pp. 265–276.
[30] Z. Abu-Aisheh, B. Gaüzère, S. Bougleux, J.-Y. Ramel, L. Brun, R. Raveaux, P. Héroux, and S. Adam, "Graph edit distance contest: Results and future challenges," *Pattern Recognit. Lett.*, vol. 100, pp. 96–103, Dec. 2017.
[31] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 18, no. 03, pp. 265–298, May 2004.
[32] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. London, U.K.: Addison-Wesley, 1974, p. 378.
[33] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.
[34] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[35] W. Zhang, Y. Zhang, L. Xu, J. Zhou, Y. Liu, M. Gu, X. Liu, and S. Yang, "Modeling IoT equipment with graph neural networks," *IEEE Access*, vol. 7, pp. 32754–32764, 2019.
[36] F. Chung, *Spectral Graph Theory*. Providence, RI, USA: AMS, 1997.
[37] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 212–220.
[38] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.
[39] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Jul. 2018.
[40] P. Riba, A. Dutta, J. Llados, and A. Fornes, "Graph-based deep learning for graphics classification," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 29–30.
[41] R. D. O. Werneck, R. Raveaux, S. Tabbone, and R. D. S. Torres, "Learning cost functions for graph matching," in *Proc. Joint IAPR Int. Workshops Stat. Techn. Pattern Recognit. (SPR) Struct. Syntactic Pattern Recognit. (SSPR)*. Cham, Switzerland: Springer, Aug. 2018, pp. 345–354.
[42] F. M. Bianchi, L. Livi, A. Rizzi, and A. Sadeghian, "A granular computing approach to the design of optimized graph classification systems," *Soft Comput.*, vol. 18, no. 2, pp. 393–412, Feb. 2014.
[43] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020.

NADEEM IQBAL KAJLA received the master's degree in computer science in 2016. He is currently pursuing the Ph.D. degree with The Islamia University of Bahawalpur. His areas of interest are pattern recognition, graph-based methods in pattern recognition, graph matching techniques, as well as graph learning and graph neural networks.

MALIK MUHAMMAD SAAD MISSEN received the master's and Ph.D. degrees from the University of Toulouse, France, in 2007 and 2011, respectively. He is currently working as an Assistant Professor with the Islamia University of Bahawalpur, Pakistan. He has also completed his postdoctoral position with the University of La Rochelle, France, in 2015. His main areas of research include information retrieval, human–computer interaction, and document image analysis.

MUHAMMAD MUZZAMIL LUQMAN received the master's (research) in informatics from the Computer Science Department, Ecole Polytechnique of François, Rabelais University of Tours, France (Univ Tours), in 2008, and the Ph.D. degree in computer science from Univ Tours, France, and the Autonoma University of Barcelona (UAB), Spain. He is currently a Research Scientist in document image analysis, pattern recognition, and computer vision. He has been working as a Research Engineer (Permanent) with the L3i Laboratory, Rochelle University [L3i][LRUniv], France, since November 2015. From September 2015 to November 2015, he worked as a Research Engineer (Contract) at the Bordeaux Bioinformatics Center–Centre de Bioinformatique de Bordeaux (CBiB), France. From September 2012 to August 2015, he was a Postdoctoral Researcher with Professor Jean-Marc Ogier, L3i Laboratory, University of La Rochelle (L3i La Rochelle), France. He has successfully defended his Ph.D. thesis—titled "Fuzzy Multilevel Graph Embedding for Recognition, Indexing and Retrieval of Graphic Document Images"—with distinction "très honorable *(magna cum laude)*" at the François Rabelais University of Tours, in March 2012. His Ph.D. thesis supervisors were Prof. Jean-Yves Ramel, Dr. Thierry Brouard, and Dr. Josep Llados. In 2007, he was awarded a scholarship by the Higher Education Commission (HEC), Government of Pakistan, for pursuing his master's studies and Ph.D. research in France. In 2004, he was awarded a gold medal and academic roll of honor by Government College University Lahore (GCU), Pakistan, for achieving distinction in bachelor's degree (Hons.) in computer science.

MICKAEL COUSTATY (Member, IEEE) is currently an Associate Professor with the Laboratoire Informatique, Image et Interaction, La Rochelle Université, France. His areas of interest are document analysis based on computer vision and natural language processing, digital trust in information exchange with the development of fraud detection techniques and document securing systems, segmentation and classification of large document flow, and pattern recognition.

ARIF MEHMOOD received the Ph.D. degree from the Department of Information and Communication Engineering, Yeungnam University, South Korea, in November 2017. Since November 2019, he has been an Assistant Professor with the Department Information Technology, The Islamia University of Bahawalpur, Pakistan. His current research interests include data mining, mainly working on AI and deep learning-based on text mining, and data science management technologies.

GYU SANG CHOI received the Ph.D. degree from the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, in 2005. He was a Research Staff Member with the Samsung Advanced Institute of Technology (SAIT) for Samsung Electronics, from 2006 to 2009. Since 2009, he has been a Faculty Member with the Department of Information and Communication, Yeungnam University, South Korea. His research areas include non-volatile memory and storage systems.