Received 14 November 2023, accepted 10 December 2023, date of publication 18 December 2023, date of current version 22 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3343843

RESEARCH ARTICLE

Topological and Sequential Neural Network Model for Detecting Fake News

DONGIN JUNG, EUNGYEOP KIM, AND YOON-SIK CHO
Department of Artificial Intelligence, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Yoon-Sik Cho (yoonsik@cau.ac.kr)

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2021R1F1A1063389) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-01341, AI Graduate School Program Chung-Ang University). The work of Eungyeop Kim was supported by the Chung-Ang University Graduate Research Scholarship in 2021.

ABSTRACT Fake news can be easily propagated through social media and cause negative societal effects. Since fake news is disinformation with malicious intent, manual fact-checking requires great effort. In order to cope with these challenges, many automatic fake news detection models have been introduced. Recent studies have shown that social network information along with news content can be used effectively for detecting fake news. In this paper, we propose a Topological and Sequential Neural Network model (TSNN) for detecting fake news by capturing the diffusion patterns between source news and users in social networks. We employ the supernode approach instead of simple graph pooling methods to extract representative features in graph topological structure. To better learn the representations in the supernode, we design two-staged graph neural networks reflecting the heterogeneity between news and Twitter users. Our model additionally captures sequential information on news diffusion path by using a transformer. We evaluate our model with two fake news benchmark datasets annotated by fact-checking websites: *PolitiFact* and *GossipCop*. TSNN achieves 92.15% accuracy and 92.11% F1-score on *PolitiFact*, and 97.91% accuracy and 97.88% F1-score on *GossipCop*. These results demonstrate that our model significantly outperforms other baselines, establishing it as a state-of-the-art solution for fake news detection. To verify the effectiveness following model configuration, we perform ablation studies to demonstrate how each component among our two-stage graph neural networks, and sequential information modules contribute to the performance improvements.

INDEX TERMS Fake news detection, graph neural network, graph classification.

I. INTRODUCTION

The technological advancement from traditional newspapers to online news platforms has revolutionized access to information. This transition has drastically enabled rapid acquisition and sharing of news on an unparalleled scale, granting readers the flexibility to choose news according to their personal preferences. However, these benefits have not come without serious drawbacks. The digital age has also witnessed an alarming increase in the production of fake news content. These deceptive articles are purposefully crafted to mislead readers, with the hidden aim of boosting click-through rates for the creators. The effects of these misleading news pieces are far-reaching, inducing personal, public, and even global damages. *Economically and emotionally,* individuals and communities have suffered due to the misinformation spread by fake news. A critical example of its detrimental effects is seen in the case of the COVID-19 pandemic, where misinformation had a substantial impact on public health guidelines and overall well-being of communities worldwide [1]. Given this scenario, the detection of fake news has transformed from being a mere concern to a crucial task, demanding concerted efforts from both societal and research domains.

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

While the phrase "fake news" is widely recognized, it is ironically undefined and lacks a standardized definition [2]. Previous studies [2], [3] have tried to tackle this issue by dividing the societal problems caused by fake news into multiple categories. These categories range from rumor, misinformation, clickbait, propaganda, to disinformation. Each type is characterized based on factors like the authenticity of the information, the intent behind spreading it, and the inherent characteristics of the information itself. In this particular research, our primary focus lies on the disinformation category of fake news, which implies news pieces that are knowingly false and are designed to deceive readers.

Most of the existing fake news detection utilizes deep learning mechanisms categorized as two types of methods: content-based and propagation-based. Content-based methods focus on analyzing the content of news articles. Using natural language text [4], [5], [6], [7] recognizes linguistic patterns, and assesses the consistency of information, or using image-text multimodality [8], [9] allows the broader perspective by considering the interplay between textual and visual components in evaluating the credibility of news content. Propagation-based methods identify the patterns of spreading information across networks and consider the dissemination pathways on social networks and online communities. To comprehend the propagation structure, some studies exploit graph neural networks (GNNs) [10], [11], knowledge-graphs [12], or multiple features [13], [14], [15] containing relationships between news, social media, and users. Especially, several graph neural network-based methodologies have been utilized to model the news propagation network in the form of a graph structure. The data structure of these models is highly effective in capturing the relationships among nodes and also in representing the propagation patterns. Within this graph neural network, nodes are designed to exchange messages and update information from neighboring nodes, simulating the interactive essence of social media networks. News content is widely disseminated via social media platforms, with Twitter being a major contributor. Twitter users share and post news on their accounts, contributing to the news propagation network. In an insightful study by Zhao et al. [16], it was found that the propagation patterns of fake news and real news exhibited big differences. It was observed that real news was largely shared in the first layer of the network, where the writer's original content was posted. However, fake news was shared more frequently beyond the first layer, meaning it was propagated via retweets more than original posts.

We propose a model called the Topological and Sequential Neural Network model (TSNN) that is specifically drawing from this significant observation. We present sophisticated experimental results on fake news detection datasets, demonstrating the potential of TSNN to outperform traditional methods in identifying fake news. Our experiments are structured to directly compare with foundational baselines, ensuring a straightforward evaluation of our model. The ablation study sheds light on the effectiveness of various components, revealing how each part contributes to the model's performance in fake news detection, while also pointing out areas for improvement.

Our contributions can be summarized as follows:

- We introduce a new concept of 'supernode', which stands as a comprehensive representation of the topological information within the graph. This approach is a notable shift from conventional methodologies, such as simple average pooling or max pooling, which may not fully capture the breadth of network data. The use of a supernode allows for a more encompassing view of the network, highlighting its distinctive features and capturing the complete topological landscape.
- In order to augment the effectiveness of the supernode and to gain a richer understanding of the graph data, we have constructed two-staged graph neural networks. This network deploys a leafGAT (leaf Graph Attention Network) at the leaf nodes and integrates a time-decay Graph Convolutional Network (GCN) throughout the full graph. This unique combination presents a multi-dimensional view of the graph, encapsulating diverse data points and providing an intricate representation of the graph structure.
- In acknowledgment of the temporal characteristics of news propagation, we have included sequential feature information along the path of news dissemination. To achieve this, we have harnessed the capabilities of a transformer architecture, a proven effective model for handling sequential data. The integration of temporal features offers a more in-depth analysis of how news propagates over time, making our model more effective in understanding and predicting fake news propagation patterns.

The remainder of this paper is organized as follows: Section II provides backgrounds for understanding our method and related work in the fields of fake news detection. Section III describes datasets about fake news detection, used in our experiments. Section IV outlines the methodology, describing the architecture of the TSNN model. Section V presents the experimental setup, including hyperparameter details, environment, evaluation metrics, and comparison with baseline methods. The ablation study with dynamic analysis is discussed in Section VI. Finally, Section VII concludes the paper by summarizing the contributions and outlining potential avenues for future research.

The implementation of our model can be found at GitHub repository.[1]

II. BACKGROUND

Our TSNN is based on graph neural networks and transformer architecture for fake news detection. GNNs are an essential method for understanding topological patterns

[1] https://github.com/dongin1009/TSNN-DFN

in our topological information layer. Previous fake news detection approaches based on GNNs were proposed with their own advantages and outstanding performance. This section describes the background knowledge of graph neural networks and previous fake news detection approaches to explain our method.

A. GRAPH NEURAL NETWORKS

GNNs are influential methods for representation learning in graph-structured data, which are commonly found in various fields such as social networks [17], chemical molecular structure [18], and recommendation systems [19]. These systems consist of nodes and edges where nodes represent entities and edges represent the connections or relationships between these entities. GNNs are useful for capturing intricate relationships and dependencies between nodes in complex graph-structured data, enabling enhanced understanding and analysis of the underlying connections. By aggregating information from neighboring nodes and propagating it through layers, GNNs can distill contextual information, making them valuable for tasks such as node classification, link prediction, and graph classification. The unique feature of GNNs is their ability to propagate node messages to neighboring nodes connected via these edges and dynamically update node features.

Several GNN convolution architectures have been developed in recent years, each designed for different types of graph-structured data. In this paper, we focus on three significant architectures: the graph convolutional network (GCN) [20], graph attention network (GAT) [21], and GraphSAGE [22]. The GCN [20] operates by propagating a node's message to its neighboring nodes and updating the node features based on the received messages. Even with its simplicity, it is a powerful method widely applied in various fields. In contrast, the GAT [21], an evolved version of the GCN, incorporates attention mechanisms into GNNs, enabling it to assign different importance levels to different nodes. This attention-driven approach enhances the GAT's ability to capture nuanced relationships within the graph, enabling it to excel in tasks that require modeling complex and varying node interactions. GraphSAGE [22] adopts a novel approach of sampling local neighbor nodes and propagating messages within these sampled nodes, which effectively reduces the computational complexity and enhances scalability. By efficiently aggregating information from a diverse set of local neighbors, GraphSAGE manages to strike a balance between computational efficiency and capturing global graph characteristics, making it well-suited for large-scale graph-based learning tasks.

Significant strides have been made in the development of GNN structures. However, effectively encapsulating an entire graph remains a challenging task. This problem is particularly noticeable in graph classification tasks. The difficulty stems from the arbitrary structure and unpredictable connections within a graph. The ordinary average pooling or max pooling is often used strategy in graph classification tasks. These methods may lead to a loss of important structural information and fail to capture nuanced patterns within the graph, resulting in suboptimal classification performance. To address this, various graph pooling methods have been proposed, such as the differentiable pooling (DiffPool) [23] which uses soft cluster assignments, and the top-K pooling [24] which selects the top-K scored nodes for representation. Both methods aim to provide a more meaningful and representative snapshot of the graph data, enhancing the graph classification performance.

B. FAKE NEWS DETECTION

The GCNFN (Graph Convolutional Network Fake News detection) model, introduced by Monti et al. [25], signified a paradigm shift in the field of fake news detection. This pioneering approach marked the first utilization of the GCN [20] within this context, thereby presenting the concept of a news propagation graph specifically designed for the task of detecting fake news. In essence, GCNFN capitalizes on the abilities of GCN to decode intricate patterns within news propagation across social networks, making substantial strides toward enhancing the identification of fake news.

After the introduction of GCNFN, the Bi-GCN (Bidirectional Graph Convolutional Network) model was proposed by Bian et al. [26] which captures graph structure bidirectionally for a rumor detection task. Taking a unique stance, Bi-GCN encompassed a bidirectional GCN operation, integrating both top-down and bottom-up approaches. By consolidating the outputs from these two unidirectional GCNs, Bi-GCN offered a comprehensive depiction of the network's information propagation, thereby improving the efficiency of fake news detection.

Further advancing the field, Dou et al. [27] have presented the UPFD (User Preference-aware Fake news Detection) model with news embeddings and user encodings as graph-structured data. This model added a novel layer of complexity by incorporating user preference, specifically targeting users who are the initial disseminators of news on Twitter. To accurately represent these user preferences, UPFD utilized an average of embeddings from a user's 200 most recent tweets. UPFD's empirical evidence suggested that combining news context and user preferences proved more effective than relying solely on news features. In addition, UPFD adopted various graph neural network architectures, including GCN [20], GAT [21], and GraphSAGE [22], to measure their effectiveness.

Addressing the issue of uncertainty within news propagation, Wei et al. [28] proposed the UPSR (Uncertainty-aware Propagation Structure Reconstruction) model for fake news detection. Utilizing a novel Gaussian Propagation Estimation, UPSR aims to reconstruct the original propagation structure and learn the latent interactions between network nodes. The UPSR reconstructs long-range and potential interactions in the uncertain propagation to explore diverse structural patterns. This novel approach offers a sophisticated way of dealing with the inherent uncertainties in news propagation, consequently strengthening the accuracy of fake news detection.

III. DATASETS

The FakeNewsNet [29] dataset[2] collected news with its fake or real label which was evaluated by renowned fact-checking websites, *PolitiFact* and *GossipCop*. The *PolitiFact* focuses on politically oriented news and the *GossipCop* is centered around celebrity gossip. These websites play a critical role in distinguishing real news from fake news. However, the original complete news and tweet texts were not disclosed by the news publisher copyrights and Twitter privacy policies. In certain cases, the web pages of original news are deleted and are no longer available.

In this study, we utilized datasets made publicly accessible[3] for the identification of fake news which have been derived from existing research [27]. Dou et al. [27] provided these datasets in the form of pre-encoded news vectors, Twitter users vectors, and their interconnection from the original news retweet graphs of FakeNewsNet [29]. The datasets incorporate news embedding and user preference embedding vectors. These user preference embedding vectors were formulated by averaging text embeddings from the most recent 200 tweets for individual users. They suggested that this process gives an overall representation of a user's preference in a condensed form.

As indicated in Table 1, the *PolitiFact* dataset comprises 157 real and fake news pieces, respectively. The real and fake news pieces maintain a balanced ratio. Also, *GossipCop* dataset has much more amount including 5464 source news pieces which have 2732 of real news and 2732 of fake news. In both datasets, the fake news cases have a higher average graph depth of news diffusion network compared to the real news cases. This statistic indicates that fake news is shared more extensively at a deeper level than real news. In other words, real news is more frequently shared as a 'tweet' (at the first depth level) than a 'retweet' (at the second depth level or above).

TABLE 1. Statistics of fake news detection benchmark datasets and graphs.

Dataset	*PolitiFact*		*GossipCop*	
Label	Real (0)	Fake (1)	Real (0)	Fake (1)
#News	157	157	2,732	2,732
Sum. #Tweets	15,310	20,280	205,347	103,451
Min. #Tweets	2	2	2	2
Max. #Tweets	496	486	198	197
#Avg. Depth per graph	3.62	3.89	2.48	2.54

For our experiments, we used pre-encoded embeddings by Word2Vec [30] from the spaCy python library. In this setup, we preserved 25% of the dataset randomly for generating a reproducible testing set, in line with the methodology presented in [28] and [31]. The remaining 75% of the dataset is subjected to a 5-fold cross-validation strategy, enhancing the reliability of our experimental outcomes. The training, validation, and testing datasets have the same label proportion as original dataset.

[2]https://github.com/KaiDMML/FakeNewsNet

[3]https://github.com/safe-graph/GNN-FakeNews#datasets

IV. METHODS

We propose our Topological and Sequential Neural Network model (TSNN) for detecting fake news. As shown in Fig. 1, TSNN is composed of 4 modules which are news diffusion modeling, topological information layer, sequential information layer, and classification layer. The news diffusion modeling is the first step, creating a detailed map of the journey news articles take on social media platforms, representing each post as nodes in a network. The topological information layer uses an element known as a 'supernode' which is a condensed expression of the entire network structure. This layer provides valuable insights into the overall topology of the news diffusion pattern. Simultaneously, the sequential information layer contributes to this process by understanding the timeline of news propagation. It uses a transformer model, a mechanism adept at tracking the sequence and timing of the news spread, effectively capturing the temporal aspect of the information flow. Finally, the classification layer employs all the insights from the previous layers and leverages this comprehensive understanding to categorize the news as either real or fake. In the following, we provide details of each module.

A. NEWS DIFFUSION MODELING

The foundation of our model is centered on the concept of news diffusion modeling. This process starts with leveraging the *FakeNewsNet* dataset [29], which is a comprehensive collection of source news articles and their corresponding shared tweets, complete with the timestamps τ_i when they were posted. These news articles, once released, found their way through the intricate web of Twitter, leading to the formation of a distinct propagation network unique to each piece of news. This structure is made as a tree structure in which a piece of source news and Twitter users, consists of a root node and leaf nodes. A graph of tree structure represents the propagation network for news sharing flow.

In this representation, the root node denoted as N or U_0, stands for the source news article when it was first introduced on the platform. The root node is the embedding of news text, that averaged Word2Vec [30] word representation for the whole news text. As the news spreads and gets shared, we see the emergence of the leaf nodes, denoted as $U_{1:n}$. Each of these leaf nodes corresponds to individual Twitter users who engaged with the article, be it through sharing, commenting, or tweeting about it. This Twitter user node was produced by averaging 200 historical tweets represented as Word2Vec [30] word embedding. In the context of our study's graph structure, the nodes represent the news pieces or the Twitter users, the edges show the interactions or

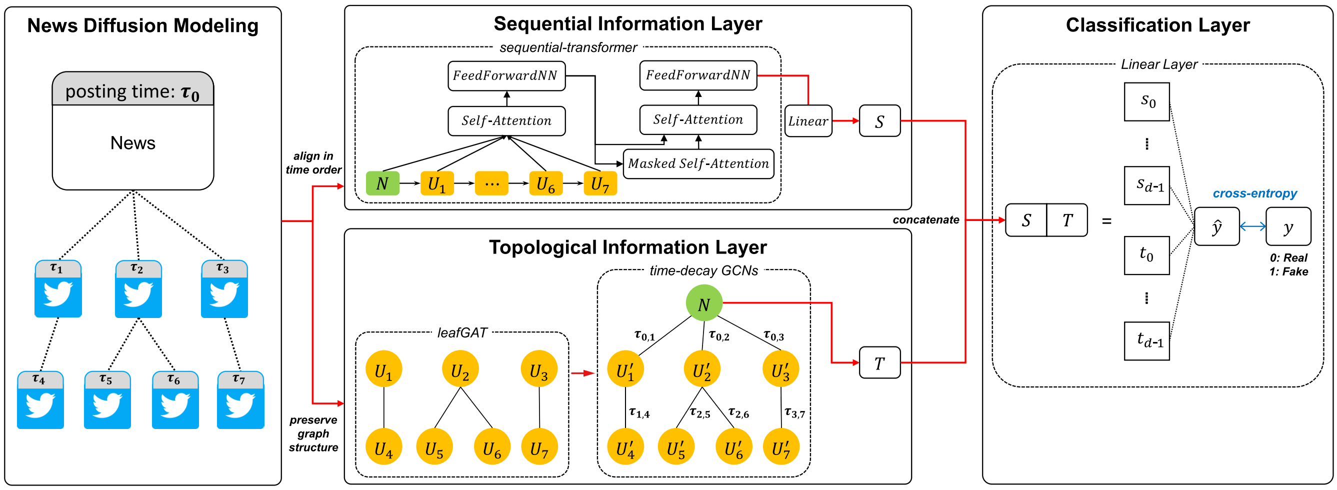

FIGURE 1. Our overall topological and sequential neural network (TSNN) model architecture. TSNN comprises four key layers that are news diffusion modeling, topological information layer, sequential information layer, and classification layer. Inputs are news N, Twitter users U_i, posting timestamp τ_i, and news-user connections. The final output $\hat{y}$ is utilized with real/fake label y for calculating loss. The model updates weights through gradient descent by cross-entropy loss function.

sharing activity between them. This interaction shows how fast a news story spreads, which can help tell if it is fake or not. Specifically, an edge might represent a user sharing a news piece or a user retweeting another user's post about a certain news article. This interaction is essential as it captures the propagation of news across the platform and helps in understanding how quickly and widely a particular news item spreads, which can be an indicator of its harmfulness.

By creating this tree structure, we aim to capture two critical aspects: the inherent graph structure representing connections and the sequence in which the news propagates. To convey forward, tree-structured news diffusion network processes appropriately to fit the next layer. First, according to posting time, news and tweets are aligned chronologically to make one sequence and this sequence feeds into the sequential-transformer of sequential information layer. Second, the tree structure preserving its nodes, edge connections, and timestamp, feeds into the two-stage GNN layers of topological information layer. These structured representations serve as the input for the subsequent processing layers within our model, ensuring a detailed and clear understanding of the news spread.

B. TOPOLOGICAL INFORMATION LAYER

The next component is the topological information layer. The news diffusion network comprises a news node and multiple tweet user nodes. Each of these nodes carries specific embedding characteristics: news embeddings represent news content, whereas tweet user embeddings signify the average of tweets. Due to the heterogeneous nature of these embeddings, our topological information layer is formulated as two-stage GNN layers comprising leafGAT and time-decay GCNs to aggregate efficiently.

The first stage, leafGAT employs GAT [21] convolution as its foundation. We eliminate a root node (news) to perform message passing within leaf nodes (user). The leafGAT updates user nodes by considering diverse user preferences and avoids discrepancies from the nature of heterogeneous nodes. In the context of a deep graph, it can be challenging to propagate the information from the end leaf node to the root node. Therefore, leafGAT focuses on bringing up the message from the end of leaf node toward the root node. By enabling efficient message propagation across nodes in a hierarchical graph structure, leafGAT ensures that information flows seamlessly across the entire graph, fostering holistic dissemination of knowledge and contributing to a more thorough understanding of intricate relationships and patterns within the data. The leafGAT executes information propagation by considering the attention between only leaf nodes in the tree-structured graph. This attention performs to find importance from Twitter users. The leafGAT is formulated as follows:

$$\begin{aligned} U_i' &= \text{leafGAT}(U_i) \\ &= \alpha_{i,i}(\mathbf{W}\cdot U_i) + \mathop{\Big\|}_{\mathbf{h}=1}^{\mathbf{H}} \Big(\sum_{j\in\mathcal{N}_i, i,j\neq 0} \alpha_{i,j}^{\mathbf{h}}(\mathbf{W}^{\mathbf{h}}\cdot U_j)\Big), \end{aligned} \quad (1)$$

where $\|$ expresses the concatenation operation, $\mathbf{H}$ denotes the number of heads in the multi-head attention mechanism, and $\mathcal{N}_i$ is the neighborhood indices of node U_i in the graph.

The trainable weight matrix $\mathbf{W}$ is learned during the training process, and it plays a role in determining how information from different nodes is integrated. Through the optimization of $\mathbf{W}$, the model gains the ability to emphasize or attenuate certain aspects of node information to capture meaningful patterns and nuances present in the graph data. The attention coefficient $\alpha_{i,j}$ is calculated for each pair (i-th and j-th) of nodes in the graph, serving as a measure of how much attention node U_i should pay to node U_j when updating its own information. This attention mechanism allows the model to dynamically allocate importance to different relationships based on the content and context of the nodes, enhancing its ability to capture both local and global dependencies within the graph. Ultimately, attention coefficient α between nodes U_i and U_j is derived using the following formula:

$$\alpha_{i,j} = \frac{\exp\left(\phi\left(\mathbf{a}^\top[\mathbf{W}\cdot U_i \| \mathbf{W}\cdot U_j]\right)\right)}{\sum_{k\in\mathcal{G}} \exp\left(\phi\left(\mathbf{a}^\top[\mathbf{W}\cdot U_i \| \mathbf{W}\cdot U_k]\right)\right)}, \tag{2}$$

where ϕ denotes the LeakyReLU [32] activation function and $\mathbf{a}$ denotes attention mechanism structured as a single-layer feed-forward neural network and $\mathcal{G}$ means whole nodes in a single graph. LeakyReLU [32] is an activation function designed to address the 'dying ReLU' problem, allowing small negative values when the input is less than zero to maintain the activation and gradient flow. This trick makes the model avoid dying ReLU and gradient vanishing problems. After through leafGAT, the nodes are directed to the second stage, time-decay GCNs having edge weight as posting time difference. To make topological information feature T, root node N and updated leaf node U' are fed to time-decay GCNs which have two layered GCNs. In this layer, all nodes are updated by their neighborhood nodes and using time difference between node pairs. After two layered GCNs, we extract supernode to topological information feature T from root node U_0'' in the final layer as follows:

$$\begin{aligned} U_i^{(l)} &= \text{time-decay GCN}(U_i') \\ &= U_i^{(l-1)} + \sum_{j\in\mathcal{N}_i\cup\{i\}} \tau_{j,i}(\mathbf{W}^{(l-1)}\cdot U_j^{(l-1)}) \\ T &= U_0'', \end{aligned} \tag{3}$$

where $\tau_{j,i}$ represents the time-decay score between nodes U_i' and U_j'. This score is used to measure the posted time gap between two nodes, U_i' and U_j'. For each node, there is a unique self-loop score, $\tau_{i,i}$, which is always assigned a consistent value of 1. This ensures each node has a base reference.

The importance of the time-decay score, $\tau_{j,i}$, plays a vital role in the context of our model. Its primary function is to gauge how much weight or importance should be given to nodes depending on how far they are from the central or root node. Nodes that are nearer to the root are assigned more importance, while nodes that are farther away are given less. In essence, it helps in distinguishing the proximity of nodes to the root. Furthermore, we use this time-decay score $\tau_{j,i}$ as an edge weight. This weight is crucial in deciding how the convolution should be applied between the i-th and j-th nodes that are interconnected. For a clearer understanding of how we calculate the time-decay score, we provide the following equation:

$$\tau_{j,i} = \frac{1}{\log(1 + (|\tau_i - \tau_j|)/60)}, \tag{4}$$

where τ_i is used to represent the posted timestamp of the i-th node, which is measured in units of seconds. Meanwhile, the time-decay factor, $\tau_{j,i}$, is calculated as the time difference between two nodes, measured in minute units. Nodes that were posted much later after the original news have lower scores, whereas those posted closer to when the news was released have higher scores. This system of scoring ensures nodes closer to the news release are deemed more significant.

By applying this method, we finally create an updated root node, denoted as U_0''. This node, also referred to as the 'supernode', is assumed to hold the entire topological information feature, T, of the entire graph. With this feature of the supernode, we ensure that the model captures all the essential time-based data while analyzing the graph's structure.

C. SEQUENTIAL INFORMATION LAYER

In the complex landscape of news diffusion networks, users propagate specific news postings on social media platforms like Twitter, creating a time-bound sequence of interactions. To effectively capture and analyze this sequential information inherent in the news diffusion network, we employ the transformer [33] architecture.

Transformer [33] is an innovative and powerful model in deep learning, which has multiple encoder and decoder blocks with a self-attention mechanism. The encoder of transformer extracts context-rich representation features from the input sequence, by self-attention capturing the relationships and dependencies on each word. The decoder of transformer generates output sequences or features based on the encoder's output features by masked self-attention and encoder-decoder attention in an autoregressive manner. Transformer offers several advantages and is shown for its exceptional performances in natural language processing, computer vision, and speech processing tasks. Rooted in its unique design, transformer leverages a multi-head dot product attention mechanism along with sinusoidal positional encoding to handle sequential data efficiently and accurately. This approach allows the model to capture varying levels of dependencies and relationships among the data, which are especially pertinent in understanding the propagation pattern of news articles over time.

In our approach, we input the news and tweet users diffusion sequence which is a list of user interactions with the news post arranged in chronological order, into the transformer model. Our chosen configuration of the transformer model comprises two stacked encoder blocks and two stacked decoder blocks, faithfully following the original architecture described in [33]. Also, all layers for the transformer have two-head attention which weighs the importance of two different aspects of the input sequence. Each encoder block integrates a multi-head self-attention mechanism and feed-forward networks, designed to extract complex feature representations from the input data. On the other hand, each decoder block comprises a masked multi-head self-attention mechanism, another multi-head encoder-decoder attention module, and a set of feed-forward networks. The masked attention ensures that the prediction for a specific position is dependent only on the known outputs at positions before or same as the current one, hence preserving the temporal sequence of the diffusion data. Unlike the original transformer, our module does not employ a tokenizer that separates the sequence into several tokens and adds special tokens to have particular purposes. This is due to the fact that the diffusion sequence already consists of distinct embeddings for individual news and Twitter users, i.e., $N, U_1, \ldots, U_i$. For this reason, our sequence has no [CLS] token for text classification, [SEP] token for separate two sequences, and [UNK] token for unknown vocabulary. Whereas we use padding in the mini-batch units like [PAD] token, and masking in masked self-attention like [$MASK$] token. Finally, we extract the last vector of final hidden state as an output vector to gain a sequential information representation for the transformer.

The output from the transformer, capturing the distilled sequential information, is then passed through a linear layer. This layer helps in additional transformation and scaling of the learned features, making them suitable for further analysis. Finally, we obtain the sequential information feature S from the output of this linear layer. Feature S encapsulates the sequential pattern of news propagation, serving as a critical input for subsequent steps in our fake news detection model.

D. CLASSIFICATION LAYER

Our model, TSNN, is designed to integrate two crucial features: topological information, denoted as T, and sequential information, referred to as S. By doing so, TSNN is capable of emphasizing the importance of both the structural patterns and the time-based sequences seen in news propagation. When T and S are combined as a concatenation, they form a comprehensive feature vector, $S||T$, which integrates the essence of both these information types.

After the formation of this comprehensive feature vector, it is then directed towards the classification layer of the model. This layer comprises two parts: a linear layer followed by a log softmax layer. A linear layer transforms feature vector to the 2-dimensional vector and the log softmax layer converts to the probability distribution over the class labels, providing an interpretable output $\hat{y}$ for the classification. The primary goal of the classification layer is to sort or categorize the incoming feature vector into one of two possible categories: real news or fake news.

For loss function, we employ cross-entropy loss function. During the training process, model utilizes the actual label of news and updates the trainable parameters through backpropagation and gradient descent. The gradient descent optimization aims to minimize the discrepancy between the model's predictions $\hat{y}$ and the actual labels y, thereby iteratively improving the model's ability to accurately classify between different categories. This iterative optimization process guides the model to learn and adapt its internal representations, fostering a more discriminative and nuanced understanding of the underlying features that distinguish real news from fake news.

V. EXPERIMENTS

In this section, the experimental settings and the environment utilized in our works are elaborated upon. We conducted an in-depth evaluation of our model's performance. When we compared our results with those of other well-known models, it became clear that our model has notable advantages and can handle a variety of situations effectively. In Table 2, we provide a clear comparison, showcasing how our model performs in relation to several other baseline models. This comparison helps highlight the strengths of our model and how it stands out in different settings.

TABLE 2. Fake news detection performance of TSNN and baselines. The evaluation results denoted as † are brought from the original paper because of cannot be reproducible.

Model	*PolitiFact*		*GossipCop*	
	Accuarcy	F1-score	Accuarcy	F1-score
TSNN (ours)	**92.15**	**92.11**	**97.91**	**97.88**
UPSR† [28]	91.4	91.0	97.7	97.6
UPFD-SAGE [27]	79.75	79.71	97.45	97.43
UPFD-GAT [27]	81.27	81.25	97.38	97.35
UPFD-GCN [27]	82.78	82.71	97.51	97.48
Bi-GCN [26]	82.53	82.45	96.84	96.80
GCNFN [25]	84.81	84.78	95.48	95.42

Moreover, a meticulous investigation was conducted to discern the effects of time units, such as seconds and minutes, and the influence of graph depth on the model's output. These variations in time-decay factors are extensively shown in Table 3. From these experiments, we observed which time units work best to improve the accuracy of decay calculations in our model. This knowledge helped us fine-tune our approach and get better results. Additionally, the intricacies of the model's design are laid out in Table 4. This table serves as a focal point for understanding the contributions of the sequential information layer, to the overall performance of the model. As a result of comparisons, we configure our time-decay GCN module as the minute-based time-decay score without Depth divide, and sequential-transformer as a 2-layered encoders-decoders transformer, and our TSNN model has about 2.7 million parameters.

A. EXPERIMENTAL SETUP

The environment for our experiments consisted of an Ubuntu 22.04 LTS, Intel Xeon Gold 6326 @ 2.90GHz CPU processor, coupled with an NVIDIA A10 GPU. To align our experiments with established practices, we incorporated a set of widely-accepted hyperparameters and techniques. We use Adam optimizer [34], L2 regularization weight of 0.001 was adopted. In terms of the model architecture, we configured our models to process a batch size of 128 and a hidden dimension d is 128. We fine-tuned our learning rate within the range of $\{0.01, 0.005, 0.001, 0.0005\}$. The final choice was

TABLE 3. A comparative analysis of the variants of the time-decay function in our time-decay GCNs module. It emphasizes the impact of changes in time unit and graph depth on the model's performance, with minute-based decay offering optimal performance.

Model	*PolitiFact*		*GossipCop*	
	Accuracy	F1-score	Accuracy	F1-score
minute-based	**92.15**	**92.11**	**97.91**	**97.88**
(*w/* Depth divide)	90.62	91.09	97.25	97.27
second-based	92.01	91.87	97.85	97.82
(*w/* Depth divide)	91.14	91.10	97.79	97.81
w/o time-decay	89.62	89.59	97.25	97.21
(*w/* Depth divide)	90.81	90.59	97.34	97.30

determined by the criterion of validation loss, allowing us to adaptively select the optimal learning rate that minimizes the validation loss.

Our models were trained for a maximum of 200 epochs. If validation loss does not decrease until 10 epochs, we made our model early stop training. The average execution time on 5 runs is 150 seconds in *PolitiFact*, and 180 seconds in *GossipCop* each runtime. As mentioned in Section III, for the data division, we split the available dataset into training and test sets at a proportion of 0.75:0.25. This was done to ensure that our model was tested on unseen data, contributing to a more objective evaluation. To ensure a comprehensive evaluation of our model's performance, we applied a 5-fold cross-validation strategy specifically to the training set. We partitioned the training data into five subsets, training the model on four of them while validating its performance on the remaining subset in each fold. This process helps us assess the model's robustness and generalization across different subsets of the training data. The reported evaluation metrics including accuracy and macro F1-score, are averaged results on five cross-runs. This repetitive testing reduces chances of anomalies linked to a single evaluation, thus presenting a more honest picture of a model's capability. Accuracy is a widely used evaluation metric that measures the ratio of correctly predicted instances to the total number of instances. F1-score is a useful metric specifically in class-imbalanced datasets, that is calculated by harmonic mean of precision and recall.

B. EXPERIMENT RESULTS

As comparing models fairly is very important, we took meticulous steps to ensure that every model, ours included, was tested under the same experimental conditions. A uniform random seed was maintained throughout all tests, ensuring reproducibility. Table 2 shows a detailed comparison of our model's performance against other principal baseline models. Notably, most of these baseline models are commonly cited in research, highlighting their importance in the field. We re-implemented these baselines to obtain evaluation results in same and fair environments, except the UPSR [28] model. Due to the unavailability of open-source code, we referred directly to the UPSR performance from the original paper by Wei et al. [28]. Moreover, most of our experiment settings follow as possible UPSR, such as dataset, data split strategy, and training setting.

Our TSNN outperforms the other solid baselines with a remarkable margin in both datasets. The gap between UPSR and other baselines was already anticipated results since reported from UPSR paper [28]. However, as TSNN overwhelms UPSR, our model achieves a noticeable performance improvement. We regard this achievement caused by advantages from the appropriate time-decay score function, supernode approach, and sequential-transformer architecture. To verify the effectiveness of our model, we conduct several ablation studies by empirical analysis in Section VI.

VI. ABLATION STUDY

To gain a deeper understanding of the role and impact of various components in our TSNN model, we undertake a comprehensive ablation study targeting each of these crucial elements. Our primary goal is to determine the contributions and significance of each piece, especially in relation to the model's overall performance. To verify the contributions of each layer in TSNN, we made adjustments to both the sequential and topological information layers and carefully observed the effects of these modifications on the model's efficiency.

A. EFFECT FOR TIME-DECAY GCNS

To deeply estimate the role of topological information layer, we closely examined two important details in time-decay GCNs module. Specifically, we looked at two main areas: how adjusting the time unit (changing from minutes to seconds) might influence the model performance, and the role of the graph depth level plays in time-decay. Our primary focus was on the original '**minute-based** time-decay score', expressed as $\tau_{j,i} = \frac{1}{\log(1+(|\tau_i - \tau_j|)/60)}$. Building on this, we also considered a factor of 60 to switch the time units from minutes to seconds, named '**second-based** time-decay score' as follows:

$$\tau_{j,i}^{(sec)} = |\frac{1}{\log(1+(|\tau_i - \tau_j|))}|. \tag{5}$$

As a comparison with the first and third rows in Table 3, the minute-based time-decay scoring function has more effect than the second-based method. The second-based approach empowers too excessively sensitive time differences to edge weight importance. The fifth row in Table 3 shows the result of removing the time-decay scoring function named '***w/o* time-decay**'. In this setup, the time-decay GCN module was simplified by substituting it with basic 2-layered GCNs without edge weight, which was omitted to discern its overall influence. Removing the time-decay score thoroughly decreases performance extremely and the time-decay score is a significant feature in our original model. We observed that an appropriate time-decay score can affect topological information propagation with time-weighted information.

As a result, we apply to divide the raw timestamp by the time scale factor 60.

Another aspect that took the minute-based decay and divided it by the depth of the graph, named '**Depth divide** decay score' is formulated as:

$$\tau_{j,i}^{(min\ w/\ Dep)} = \frac{1}{\log(1 + (|\tau_i - \tau_j|)/60)/D}, \tag{6}$$

where D indicated node's depth level in a graph. We expect that the deeper level node of graph might be less important than around of root node. However, as can show performance with the Depth divide score function in Table 3, the Depth divide method widens too much gap depending on the depth level. This scoring method makes edge weight weaker on the more deep depth level.

After testing these variants of the time-decay scoring function thoroughly, it becomes clear that the straightforward minute-based approach without depth dividing to time-decay is the standout performer. It is evident that embedding the time-decay score within the GCN significantly amplified the model's capacity to grasp intricate topological details. These comparative experiments emphasize the importance of selecting the correct time unit when working with GCN, and in our experiments, focusing on minutes provided the best performance.

B. EFFECT FOR SEQUENTIAL TRANSFORMER

To prove validity of sequential information layer, we adjust stacking layers of the transformer or alter the model. As shown in Table 4, we exhibit the performances of various stacked sequential transformers and, other recurrent neural network model families.

TABLE 4. Our ablation study demonstrates the impact of different model components on the overall performance of the TSNN model. Key findings show the superior performance of the 2-layered encoders-decoders transformer in capturing sequential information.

Model	*PolitiFact*		*GossipCop*	
	Accuarcy	F1-score	Accuarcy	F1-score
seq-transformer				
(2 enc-2 dec)	**92.15**	**92.11**	**97.91**	**97.88**
(3 enc-3 dec)	90.38	90.31	97.01	96.98
(4 enc-4 dec)	89.37	89.34	97.10	97.07
(4 encoders)	91.65	91.61	97.44	97.41
seq-LSTM	91.39	91.34	97.09	97.05
seq-GRU	91.14	91.10	97.05	97.01

First, we reduce the number of transformer encoders and decoders appropriately. We evaluate the adjusted transformer on various stacked layers to decide on transformer configuration. As shown from the first to fourth rows in Table 4, comparative targets are for each, **2, 3, 4 layered encoders-decoders** and only **4 layered encoders**. These configurations with reduced layers have fewer model parameters than the original transformer from Vaswani et al. [33]. The deep learning models with more parameters commonly have a greater capacity to memorize and generalize from training. However, it has disadvantages which include potential overfitting on smaller datasets, and the need for extensive tuning and regularization techniques. In our case, the data amount is insufficient to gain better representation and train the deeper model. The results in Table 4 reveal that the layer-diminished transformer shows better performance. Therefore, the transformer with 2 layered encoders-decoders is the most favorable configuration in our datasets. Furthermore, the 4 encoders variation has the second-best performance even though it removed decoder part in transformer and its performance is slightly reduced than the 2-layered encoders-decoders version.

Second, we shifted our focus to two specific models: the **sequential-LSTM** and the **sequential-GRU**. These networks are the most popular and widely used type of RNNs for modeling sequential data and alleviating the limitation of traditional RNN. We select these two models due to their prominence and the underlying mechanics they offered. The sequential-LSTM model incorporated a two-layered bidirectional LSTM [35], which involves stacking two layers of LSTM units and processing input sequences in both forward and backward directions. Likewise, the sequential-GRU is built upon a two-layered bidirectional GRU [36] which has more simple architecture and fewer parameters than LSTM [35]. In both instances, these architectures are employed as alternatives to the transformer setup typically found in recurrent neural network models. The 2-layered encoders-decoders sequential-transformer outperforms other RNNs. These results signify that transformer is more effective than RNNs in sequential data with long-range dependencies.

Finally, we found the superior configuration and model in the sequential information layer. Our meticulous research, the details of which are elaborated in Table 4, led to some key insights. Most notably, the 2-layered encoders-decoders transformer consistently outperformed its counterparts, namely those built on LSTM and GRU frameworks for adopting to sequential information layer. This result can show the validity of our sequential-transformer in designing the news-tweet diffusion path for sequential features.

VII. CONCLUSION

In response to the increasing prevalence of fake news, we present a topological and sequential neural network model (TSNN) for more accurate fake news detection. Our TSNN model incorporates both topological and sequential information in news propagation networks, thus offering a more comprehensive approach to identifying fake news. In our model, topological information is represented through a supernode which is extracted by using two-staged graph neural networks. The first stage involves the application of a leaf graph attention network (leafGAT) at the leaf nodes. The second stage introduces two layered time-decay graph convolutional networks (time-decay GCNs) that cover the full graph. Furthermore, the TSNN model utilizes a transformer architecture to capture the sequential features in the news diffusion path. This element of the model allows for identifying the sequence in which news is shared and is often a significant clue in detecting fake news. We evaluated the effect of our contribution by ablation study with sophisticated observation. TSNN reached 92.15% accuracy and 92.11% F1-score on *PolitiFact*, and 97.91% and 97.88% on *GossipCop*, respectively. These results clearly indicate that our model significantly outperforms other foundational baselines in fake news detection benchmark datasets. In summary, the TSNN model proposed in this paper represents a significant step forward in the field of fake news detection. By using both topological and sequential information, the model provides a more nuanced and effective approach to identifying fake news. The approach is expected to contribute to ongoing research and development in this area.

Nevertheless, this approach and task have an unattainable limitation that may be challenging to overcome. The dataset has a static status of fixed time points in this scenario. In this context, the dataset may not reflect real-time changes, as actual news and its dissemination network are subject to rapid updates and modifications. This static characteristic of the dataset could lead to a potential mismatch with the evolving news content, making it difficult to capture the latest information. In future work, we will explore fake news in constantly changing status and real-time environments by analyzing unstable news history. We expect that continual learning or online learning approaches are useful for adapting new and changeable data.

REFERENCES

[1] O. D. Apuke and B. Omar, "Fake news and COVID-19: Modelling the predictors of fake news sharing among social media users," *Telematics Informat.*, vol. 56, Jan. 2021, Art. no. 101475.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explor. Newslett.*, vol. 19, no. 1, pp. 22–36, 2017.

[3] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–40, Sep. 2021.

[4] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending against neural fake news," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[5] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid CNN-RNN based deep learning approach," *Int. J. Inf. Manage. Data Insights*, vol. 1, no. 1, Apr. 2021, Art. no. 100007.

[6] K. Shu, G. Zheng, Y. Li, S. Mukherjee, A. H. Awadallah, S. Ruston, and H. Liu, "Early detection of fake news with multi-source weak social supervision," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Ghent, Belgium, Sep. 2021, pp. 650–666.

[7] J. A. Leite, O. Razuvayevskaya, K. Bontcheva, and C. Scarton, "Detecting misinformation with LLM-predicted credibility signals and weak supervision," 2023, *arXiv:2309.07601*.

[8] X. Zhou, J. Wu, and R. Zafarani, "SAFE: Similarity-aware multi-modal fake news detection," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2020, pp. 354–367.

[9] Y. Zhou, Y. Yang, Q. Ying, Z. Qian, and X. Zhang, "Multimodal fake news detection via CLIP-guided learning," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2023, pp. 2825–2830.

[10] Z. Wu, D. Pi, J. Chen, M. Xie, and J. Cao, "Rumor detection based on propagation graph neural network with attention mechanism," *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113595.

[11] C. Song, K. Shu, and B. Wu, "Temporally evolving graph neural network for fake news detection," *Inf. Process. Manage.*, vol. 58, no. 6, Nov. 2021, Art. no. 102712.

[12] B. Xie, X. Ma, J. Wu, J. Yang, and H. Fan, "Knowledge graph enhanced heterogeneous graph neural network for fake news detection," *IEEE Trans. Consum. Electron.*, to be published.

[13] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news detection," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 797–806.

[14] V.-H. Nguyen, K. Sugiyama, P. Nakov, and M.-Y. Kan, "FANG: Leveraging social context for fake news detection using graph representation," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1165–1174.

[15] P. Bazmi, M. Asadpour, and A. Shakery, "Multi-view co-attention network for fake news detection by modeling topic-specific user and news source credibility," *Inf. Process. Manage.*, vol. 60, no. 1, Jan. 2023, Art. no. 103146.

[16] Z. Zhao, J. Zhao, Y. Sano, O. Levy, H. Takayasu, M. Takayasu, D. Li, J. Wu, and S. Havlin, "Fake news propagates differently from real news even at early stages of spreading," *EPJ Data Sci.*, vol. 9, no. 1, p. 7, Dec. 2020.

[17] Y. Yang, "COVID-19 fake news detection via graph neural networks in social media," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2021, pp. 3178–3180.

[18] O. Wieder, S. Kohlbacher, M. Kuenemann, A. Garon, P. Ducrot, T. Seidel, and T. Langer, "A compact review of molecular property prediction with graph neural networks," *Drug Discovery Today, Technol.*, vol. 37, pp. 1–12, Dec. 2020.

[19] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: A survey," *ACM Comput. Surv.*, vol. 55, no. 5, pp. 1–37, May 2023.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[21] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[22] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," Tech. Rep., 2017, pp. 1025–1035.

[23] Z. Su, Z. Hu, and Y. Li, "Hierarchical graph representation learning with local capsule pooling," in *Proc. ACM Multimedia Asia*. Red Hook, NY, USA: Curran Associates Inc., Dec. 2021, pp. 4805–4815.

[24] B. Knyazev, G. W. Taylor, and M. Amer, "Understanding attention and generalization in graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[25] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," 2019, *arXiv:1902.06673*.

[26] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and A. Huang, "Rumor detection on social media with bi-directional graph convolutional networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 549–556.

[27] Y. Dou, K. Shu, C. Xia, P. S. Yu, and L. Sun, "User preference-aware fake news detection," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2021.

[28] L. Wei, D. Hu, W. Zhou, and S. Hu, "Uncertainty-aware propagation structure reconstruction for fake news detection," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 2759–2768.

[29] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media," *Big Data*, vol. 8, no. 3, pp. 171–188, Jun. 2020.

[30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013.

[31] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, "DEFEND: Explainable fake news detection," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 395–405.

[32] A. Maas, A. Hannun, and A. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, vol. 30, no. 1. Atlanta, GA, USA, 2013, p. 3.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[36] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proc. 8th Workshop Syntax, Semantics Struct. Stat. Transl.*, 2014, pp. 103–111.

DONGIN JUNG received the B.S. degree in computer software engineering from Wonkwang University, Iksan-si, South Korea, in 2021, and the M.S. degree in artificial intelligence from Chung-Ang University, Seoul, South Korea, in 2023. His research interests include natural language processing and large-scale data science.

EUNGYEOP KIM received the B.A. degree in English and English literature and in software humanities convergence from Chung-Ang University, Seoul, South Korea, in 2021, where he is currently pursuing the M.S. degree in artificial intelligence. His research interests include multimodal video understanding, graph neural networks, and contrastive learning.

YOON-SIK CHO received the B.S. degree in electrical engineering from Seoul National University, South Korea, in 2003, and the Ph.D. degree in electrical engineering from the University of Southern California, USA, in 2014. He was an Academic Mentor of the RIPS Program, Institute for Pure and Applied Mathematics, University of California at Los Angeles, and a Postdoctoral Scholar with the Information Sciences Institute, University of Southern California, before he joined Sejong University. He is currently an Assistant Professor with the Department of AI, Chung-Ang University, South Korea. His research interests include large-scale data science, social network analysis, and cloud computing.