EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

Textual Information Processing Based on Multi-Dimensional Indicator Weights

Yuliang Yang[1], Zhengping Lin[1], Yuzhong Zhou[1], Jiahao Shi[1], and Jie Lin[1,*]

[1]Electric Power Research Institute, China Southern Power Grid Company, Guangzhou, China.

Abstract

With the rapid advancement of artificial intelligence and wireless communication, the volume of textual information has grown significantly, accompanied by multidimensional metrics such as innovation, application prospects, key technologies, and expected outcomes. Extracting valuable insights from these multifaceted metrics and establishing an effective composite evaluation weighting framework poses a pivotal challenge in the text information processing. In this work, we propose a novel approach for textual information processing, leveraging multi-dimensional indicator weights (MDIWs). Our method involves extracting semantic information from text and inputting it into a long short term memory (LSTM)-based textual information processor (TIP) to generate MDIWs. These MDIWs are then processed to create a judgment matrix following by eigenvalue decomposition and normalization, capturing intricate semantic relationships. Our framework enhances the comprehension of multi-dimensional aspects within textual data, offering potential benefits in various applications such as sentiment analysis, information retrieval, and content summarization. Experimental results underscore the effectiveness of our approach in refining and utilizing MDIWs for improved understanding and decision-making.

Received on 29 August 2023; accepted on 29 November 2023; published on 04 December 2023

Keywords: Textual information processing, semantic information, multi-dimensional indicator weights, artificial intelligence

doi:10.4108/eetsis.3805

1. Introduction

The convergence of wireless communication has ushered in a transformative era in the realm of text information processing [1–3]. These interconnected technological advancements have collectively created an ecosystem where the acquisition, transmission, and analysis of textual data have reached unprecedented levels of efficiency and capability [4–6]. With the proliferation of wireless communication technologies, ranging from 4G to 5G and beyond, information can be exchanged in real-time, facilitating the seamless flow of text-based data across various communication platforms, social networks, and messaging applications. This surge in data generation, fueled by the widespread use of smartphones, tablets, and other smart devices, has led to an exponential increase in the volume and diversity of textual content available for analysis [7–10].

In parallel, the rise of edge computing has redefined the processing landscape by bringing computation closer to data sources [11–13]. Edge devices, ranging from edge servers to IoT endpoints, possess increasing computational power and intelligence, allowing them to perform complex text analytics and natural language processing tasks at the edge of the network [14–16]. This paradigm shift has not only reduced the latency associated with transmitting data to distant cloud servers but has also paved the way for enhanced data privacy and security [17–20]. By processing sensitive text data locally, edge computing mitigates potential risks associated with data exposure during transmission, catering to the stringent privacy requirements of various industries.

Additionally, the integration of IoT networks has contributed a plethora of data-generating devices to the ecosystem, each capable of producing a wealth of textual information [21–23]. These devices, which include sensors, wearables, smart appliances, and

*Corresponding author. Email: linjiecspg@126.com.

industrial machinery, communicate with each other and with centralized systems, generating a constant stream of textual data that reflects user interactions, environmental conditions, and operational statuses [24–27]. This vast and dynamic pool of data presents an invaluable resource for text analytics applications such as sentiment analysis, contextual understanding, and predictive modeling. Insights extracted from this text data offer a deep understanding of user behavior, preferences, and experiences, enabling organizations to tailor their offerings to meet evolving demands [28–30].

In this landscape, applications of text information processing have become increasingly diverse and impactful. Real-time language translation services have harnessed edge computing to provide instant and contextually accurate translations, improving cross-cultural communication. Sentiment analysis algorithms, capable of swiftly processing vast volumes of text from social media platforms and customer reviews, offer businesses insights into customer opinions, enabling them to make data-driven decisions. Moreover, edge devices within IoT networks can collaboratively process and analyze textual data, yielding timely insights that empower predictive maintenance, anomaly detection, and more.

Motivated by the above literature review, we introduce an innovative approach to the textual information processing in this paper, by leveraging the concept of multi-dimensional indicator weights (MDIWs). Our method encompasses the extraction of semantic information from text, which is subsequently input into an long short term memory (LSTM)-based textual information processor (TIP) to generate MDIWs. These MDIWs undergo a series of transformations, culminating in the creation of a comprehensive judgment matrix that incorporates eigenvalue decomposition and normalization techniques. This process effectively captures intricate semantic relationships inherent in the data. Simulation results are demonstrated to show the efficacy of our approach, revealing its capability to refine and harness MDIWs for enhanced understanding and decision-making.

2. Textual Information Processing

2.1. Semantic extraction

Semantic extraction in textual information processing involves the task of understanding and extracting the underlying meaning and relationships present in text data. It goes beyond simple keyword-based extraction by focusing on the context, structure, and relationships between words and phrases. This process is crucial for various natural language processing (NLP) applications like information retrieval, sentiment analysis, question answering, and more. The textual information processing is detailed as follows.

- **Word embeddings and named entity recognition (NER):** Word embeddings are numerical representations of words that capture their semantic meaning. One common method to generate word embeddings is Word2Vec. It uses the skip-gram model where given a target word w_t and a context window of surrounding words w_c, it maximizes the probability of context words associated with the target word, given by

$$\theta_{we} = \max_{w_c} \sum_{w_t} \log P(w_c|w_t). \tag{1}$$

NER identifies entities such as names of persons, organizations, locations, dates, etc., in text. It involves classifying words into predefined categories. A popular approach is using conditional random fields (CRFs) that models the conditional probability of a sequence of labels given the input sequence, given by

$$\theta_{ner} = \arg\max_{x} P(y|x), \tag{2}$$

where y denotes the sequence of labels and x is the input sequence.

- **Dependency parsing and semantic role labeling (SRL):** Dependency parsing reveals the grammatical relationships between the words in a sentence. A common approach is to represent the sentence as a dependency tree where words are nodes, and edges represent the syntactic relationship, given by

$$\text{Score}(w_i, R, w_j) = f(w_i, w_j) + g(w_i, w_j) + h(w_i, w_j), \tag{3}$$

where SRL aims to identify the predicate-argument structure in a sentence. Given a sentence and its predicate, it involves labeling the roles of words in the sentence with respect to that predicate, given by

$$\theta_{srl} = \arg\max_{x} P(y_1, y_2, ..., y_n|x, p), \tag{4}$$

where y_i is the semantic role label for word i, and p denotes the prediction result.

- **Coreference resolution:** Coreference resolution deals with identifying when different words refer to the same entity. It is typically modeled as a clustering problem, given by

$$\begin{aligned}\text{Score}(\text{mention}_i, \text{mention}_j) &= f(\text{mention}_i, \text{mention}_j) \\ &\quad + g(\text{mention}_i, \text{mention}_j)\end{aligned} \tag{5}$$

where mention_i is the mention of word i.

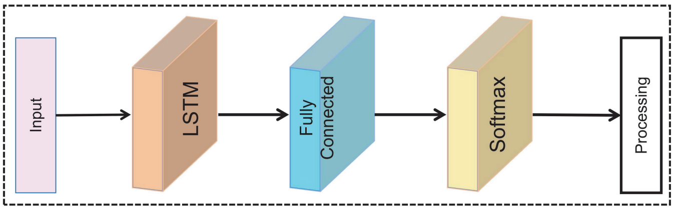

Figure 1. Framework of LSTM-based textual information processing.

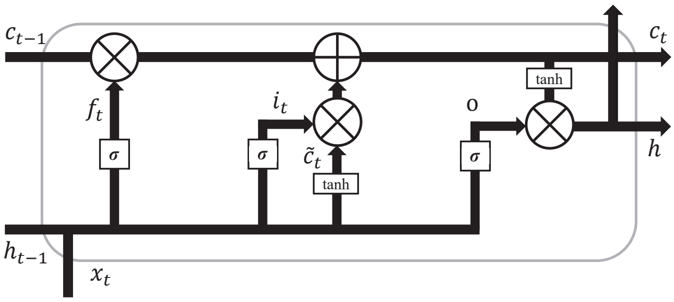

Figure 2. Structure of an LSTM cell.

2.2. LSTM-based Textual Information Processing

After obtaining the above semantic information in the text, we proceed to perform the LSTM-based textual information processing, where the framework is shown in Fig. 1. In this framework, the obtained semantic information θ_{we}, θ_{ner}, $\mathrm{Score}(w_i, R, w_j)$, θ_{srl}, and $\mathrm{Score}(\mathrm{mention}_i, \mathrm{mention}_j)$ are sequentially to LSTM network, fully connected network, softmax function and multi-dimensional indicator weights processing.

Specifically, the LSTM network is a type of recurrent neural network designed to process sequential data, such as text. In this step, the input semantic information is sequentially fed into the LSTM network. The LSTM is capable of capturing and learning the temporal dependencies within the input sequence, by utilizing gating mechanisms to address the vanishing gradient problem. In particular, Let x_t be the input at time step t, h_t be the hidden state at time step t, and c_t be the cell state at time step t.

Fig. 2 shows a typical LSTM cell, which consists of three main gates and their corresponding operations:

- **Forget gate (f_t):** The forget gate decides what information to forget from the cell state. It takes the previous hidden state h_{t-1} and the current input x_t as inputs. The forget gate output is in the range of $[0, 1]$, indicating how much information should be discarded from the cell state, given by,

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \tag{6}$$

- **Input gate (i_t) and candidate cell state ($\tilde{c}_t$):** The input gate determines which values to update in the cell state. Additionally, a new candidate cell state $\tilde{c}_t$ is computed, representing potential new values to add to the cell state, given by,

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{7}$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c). \tag{8}$$

- **Update cell state (c_t):** The cell state is updated by combining the information to forget and the new candidate information, given by

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t. \tag{9}$$

- **Output gate (o_t) and hidden state (h_t):** The output gate determines what information to output from the cell state. It takes into account the current input x_t and the previous hidden state h_{t-1}. The hidden state at the current step h_t is computed using the updated cell state, given by,

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{10}$$

$$h_t = o_t \cdot \tanh(c_t). \tag{11}$$

The LSTM's gating mechanisms allow it to selectively remember and forget information, making it effective for processing long sequences of data and capturing important patterns.

After processing the input through the LSTM, the output from the LSTM's last step is passed through a fully connected neural network layer. This layer consists of neurons that compute weighted combinations of the LSTM outputs. This step allows for higher-level feature extraction and transformation of the sequential information. The output of the fully connected layer is then usually passed through a softmax function. The softmax function converts the raw scores from the previous layer into probabilities. It normalizes the scores across different classes, making them interpretable as probabilities that the input belongs to each class. This is particularly useful for tasks involving classification or probability estimation.

Overall, this framework combines the power of LSTM-based sequential information processing with subsequent stages of transformation through fully connected layers, probability estimation through the softmax function, and weight adjustment to create a system capable of processing and understanding complex textual information. The described steps are typical of neural network architectures used for tasks like natural language processing, information extraction, and text understanding.

2.3. Multi-Dimensional Indicator Weights Processing

From the output of the LSTM-based textual information processing, we will investigate the multi-dimensional indicator weights processing in the following steps. Firstly, a judgment matrix needs to be created to compare the relative importance between different indicators. In the judgment matrix, we use scales of 1 to 9 to represent the relative importance between two indicators. Then, for each judgment matrix, we need to calculate the eigenvectors to obtain the weights of each indicator. After that, we need to perform the consistency testing to ensure the consistency of the judgment matrix, which can be achieved by calculating the consistency ratio (CR). In particular, the judgment matrix is consistent if CR is close to 0, or needs to be readjusted otherwise. Finally, for the weights of each first level indicator, we normalize them to make the sum equal to unity.

3. Experimental Results and Discussions

In this part, we perform some experiments on the textual information processing, to show some results and provide some discussions on the system design. In particular, we consider the following three data sets, which are three typical data sets for textual information processing and classification:

- **IMDB dataset:** This dataset has been curated specifically for the binary sentiment classification task pertaining to movie reviews. IMDB dataset comprises an equitable distribution of affirmative and unfavorable reviews. The dataset is meticulously partitioned into training and test sets, with a balanced count of 25,000 reviews in each subset.
- **Stanford sentiment treebank (SST) dataset:** This dataset encompasses a corpus of 11,855 movie reviews systematically categorized into distinct subsets. The corpus is differentially allocated into 8,544 training samples, 1,101 development samples, and 2,210 test samples, thereby enabling comprehensive evaluation.
- **DBpedia dataset:** This dataset is a substantial and multilingual knowledge repository, which is a result of meticulously curating the most frequently utilized infoboxes within the Wikipedia ecosystem. DBpedia undergoes periodic updates, with subsequent releases witnessing the addition or removal of certain classes and properties. Notably, the prominent edition of DBpedia comprises an extensive training cohort of 560,000 samples and a testing cohort encompassing 70,000 samples, all annotated with 14-class labels.

Fig. 3 and Table 1 show the textual classification accuracy with IMDB dataset, where the performances of both convolutional neural network (CNN) and graph neural network (GNN) are also plotted for performance comparison. The investigation spans across a spectrum of deep network configurations encompassing 2, 3, 4, 5, and 6 layers. Notably, the proposed scheme consistently emerges as the vanguard, boasting unparalleled classification accuracy. Specifically, for networks with 3 layers, the proposed scheme achieves an accuracy of 0.7991, while CNN and GNN attain 0.7512 and 0.6998 respectively. Equally compelling, with 6 layers, the proposed scheme reaches a remarkable accuracy of 1.00, surpassing CNN's accuracy of 1.00 and GNN's accuracy of 0.9910. Evidently, the proposed scheme attains superiority over both CNN and GNN, and intriguingly, all three methods exhibit an upward trajectory in performance as the network's depth increases¡ªa testament to the potency of deeper architectures in enhancing classification accuracy.

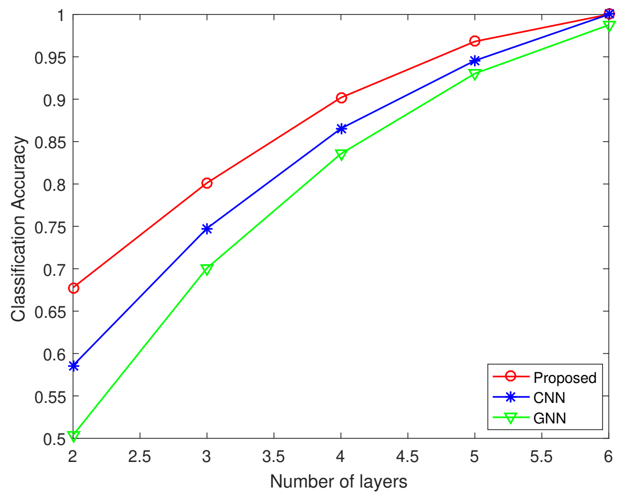

Figure 3. Textual classification accuracy with IMDB dataset.

Table 1 Numerical textual classification accuracy with IMDB dataset.

Number of layers	2	3	4	5	6
Proposed	**0.6778**	**0.8012**	**0.9018**	**0.9681**	**1.00**
CNN	0.5858	0.7477	0.8654	0.9455	1.00
GNN	0.5037	0.7008	0.8356	0.9302	0.9875

Table 2 Numerical textual classification accuracy with SST dataset.

Number of layers	2	3	4	5	6
Proposed	**0.642**	**0.763**	**0.853**	**0.913**	**0.955**
CNN	0.554	0.713	0.822	0.900	0.951
GNN	0.477	0.663	0.797	0.884	0.940

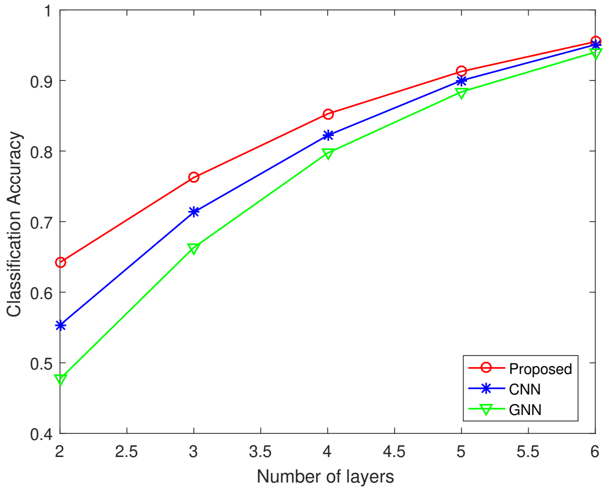

Figure 4. Textual classification accuracy with SST dataset.

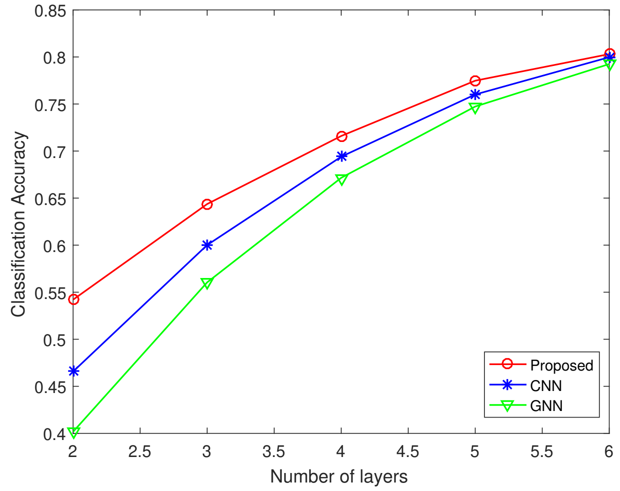

Figure 5. Textual classification accuracy with DBpedia dataset.

Table 3 Numerical textual classification accuracy with DBpedia dataset.

Number of layers	2	3	4	5	6
Proposed	**0.542**	**0.644**	**0.716**	**0.775**	**0.803**
CNN	0.466	0.600	0.694	0.760	0.799
GNN	0.402	0.561	0.672	0.747	0.793

Fig. 4 and Table 2 illustrate the textual classification accuracy of the proposed method with SST dataset, where the number of layers varies from 2 to 4. The analysis of the provided figure and table distinctly illustrates the consistent prominence of the proposed scheme, characterized by its unparalleled classification accuracy. In particular, for network configurations featuring 3 layers, the proposed scheme demonstrates a notable accuracy of 0.7625, while CNN and GNN achieve accuracy values of 0.7134 and 0.6633 respectively. Similarly compelling results emerge when considering networks with 6 layers, where the proposed scheme achieves a remarkable accuracy of 0.955, distinctly surpassing the accuracies of CNN (0.951) and GNN (0.940). This discernible superiority of the proposed scheme over CNN and GNN is evident. Notably, an intriguing trend is observed: all three methodologies exhibit an ascending performance trajectory with increasing network depth. This trend underscores the efficacy of deeper architectural configurations in enhancing the precision of classification tasks.

Fig. 5 and Table 3 demonstrate the textual classification accuracy of the proposed method with DBpedia dataset, where the number of layers varies from 2 to 4. Derived from the graphical representation and the tabular data, it becomes conspicuously apparent that the proposed methodology consistently assumes a leading position, characterized by an unmatched degree of classification accuracy. Specifically, in scenarios involving networks comprising three layers, the proposed scheme attains a precision level of 0.64, while in parallel, the CNN and GNN models achieve accuracies of 0.60 and 0.56, respectively. Notably, this trend extends to networks with six layers, where the proposed scheme remarkably achieves an accuracy of 0.803, thus transcending the precision of CNN (0.799) and GNN (0.7925). This discernible pattern firmly establishes the proposed scheme's superiority over both CNN and GNN. Furthermore, it is of particular intrigue that all three approaches evince an ascending trajectory in performance as the network's depth escalates. This phenomenon serves as a substantiation of the efficacy inherent in deeper architectural configurations for the enhancement of classification accuracy.

4. Conclusions

In this paper, we introduced a novel approach to the textual information processing through the utilization of MDIWs. Our method harnessed the power of LSTM-based textual information processing to generate MDIWs by extracting and processing semantic information from text. Subsequently, the MDIWs were transformed into a comprehensive judgment matrix using eigenvalue decomposition and normalization. This framework facilitated the identification and understanding of intricate semantic relationships embedded within textual data. The simulation results showcased the effectiveness of our approach in refining and effectively utilizing MDIWs, enhancing the comprehension of multi-dimensional attributes in the sentiment analysis, information retrieval, and content summarization.

4.1. Copyright

References

[1] A. E. Haddad and L. Najafizadeh, "The discriminative discrete basis problem: Definitions, algorithms, benchmarking, and application to brain's functional dynamics," *IEEE Trans. Signal Process.*, vol. 71, pp. 1–16, 2023.

[2] R. Gabrys, S. Pattabiraman, and O. Milenkovic, "Reconstruction of sets of strings from prefix/suffix compositions," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 3–12, 2023.

[3] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Hierarchical federated learning with quantization: Convergence analysis and system design," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 2–18, 2023.

[4] T. Häckel, P. Meyer, F. Korf, and T. C. Schmidt, "Secure time-sensitive software-defined networking in vehicles," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 35–51, 2023.

[5] Y. Song, Z. Gong, Y. Chen, and C. Li, "Tensor-based sparse bayesian learning with intra-dimension correlation," *IEEE Trans. Signal Process.*, vol. 71, pp. 31–46, 2023.

[6] H. Wan and A. Nosratinia, "Short-block length polar-coded modulation for the relay channel," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 26–39, 2023.

[7] K. N. Ramamohan, S. P. Chepuri, D. F. Comesaña, and G. Leus, "Self-calibration of acoustic scalar and vector sensor arrays," *IEEE Trans. Signal Process.*, vol. 71, pp. 61–75, 2023.

[8] Q. Lu, S. Li, B. Bai, and J. Yuan, "Spatially-coupled faster-than-nyquist signaling: A joint solution to detection and code design," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 52–66, 2023.

[9] B. Han, V. Sciancalepore, Y. Xu, D. Feng, and H. D. Schotten, "Impatient queuing for intelligent task offloading in multiaccess edge computing," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 59–72, 2023.

[10] H. Hou, Y. S. Han, P. P. C. Lee, Y. Wu, G. Han, and M. Blaum, "A generalization of array codes with local properties and efficient encoding/decoding," *IEEE Trans. Inf. Theory*, vol. 69, no. 1, pp. 107–125, 2023.

[11] X. Zhou, D. He, M. K. Khan, W. Wu, and K. R. Choo, "An efficient blockchain-based conditional privacy-preserving authentication protocol for vanets," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 81–92, 2023.

[12] D. Malak and M. Médard, "A distributed computationally aware quantizer design via hyper binning," *IEEE Trans. Signal Process.*, vol. 71, pp. 76–91, 2023.

[13] Y. Xiong, S. Sun, L. Liu, Z. Zhang, and N. Wei, "Performance analysis and bit allocation of cell-free massive MIMO network with variable-resolution adcs," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 67–82, 2023.

[14] Z. Feng, M. Yu, S. A. Evangelou, I. M. Jaimoukha, and D. Dini, "Mu-synthesis PID control of full-car with parallel active link suspension under variable payload," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 176–189, 2023.

[15] S. Khirirat, X. Wang, S. Magnússon, and M. Johansson, "Improved step-size schedules for proximal noisy gradient methods," *IEEE Trans. Signal Process.*, vol. 71, pp. 189–201, 2023.

[16] C. Chaieb, F. Abdelkefi, and W. Ajib, "Deep reinforcement learning for resource allocation in multi-band and hybrid OMA-NOMA wireless networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 187–198, 2023.

[17] T. Gafni, B. Wolff, G. Revach, N. Shlezinger, and K. Cohen, "Anomaly search over discrete composite hypotheses in hierarchical statistical models," *IEEE Trans. Signal Process.*, vol. 71, pp. 202–217, 2023.

[18] A. Gupta, M. Sellathurai, and T. Ratnarajah, "End-to-end learning-based full-duplex amplify-and-forward relay networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 199–213, 2023.

[19] H. Ma, Y. Fang, P. Chen, S. Mumtaz, and Y. Li, "A novel differential chaos shift keying scheme with multidimensional index modulation," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 237–256, 2023.

[20] B. Wang, "Internal boundary between entanglement and separability within a quantum state," *IEEE Trans. Inf. Theory*, vol. 69, no. 1, pp. 251–261, 2023.

[21] L. Hu, H. Li, P. Yi, J. Huang, M. Lin, and H. Wang, "Investigation on AEB key parameters for improving car to two-wheeler collision safety using in-depth traffic accident data," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 113–124, 2023.

[22] Y. Liu, Z. Tan, A. W. H. Khong, and H. Liu, "An iterative implementation-based approach for joint source localization and association under multipath propagation environments," *IEEE Trans. Signal Process.*, vol. 71, pp. 121–135, 2023.

[23] K. Ma, S. Du, H. Zou, W. Tian, Z. Wang, and S. Chen, "Deep learning assisted mmwave beam prediction for heterogeneous networks: A dual-band fusion approach," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 115–130, 2023.

[24] Z. Zhang, Z. Shi, and Y. Gu, "Ziv-zakai bound for doas estimation," *IEEE Trans. Signal Process.*, vol. 71, pp. 136–149, 2023.

[25] S. Guo and X. Zhao, "Multi-agent deep reinforcement learning based transmission latency minimization for delay-sensitive cognitive satellite-uav networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 131–144, 2023.

[26] X. Fang, W. Feng, Y. Wang, Y. Chen, N. Ge, Z. Ding, and H. Zhu, "Noma-based hybrid satellite-uav-terrestrial networks for 6g maritime coverage," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 138–152, 2023.

[27] R. Gabrys, V. Guruswami, J. L. Ribeiro, and K. Wu, "Beyond single-deletion correcting codes: Substitutions and transpositions," *IEEE Trans. Inf. Theory*, vol. 69, no. 1, pp. 169–186, 2023.

[28] S. Mosharafian and J. M. Velni, "Cooperative adaptive cruise control in a mixed-autonomy traffic system: A hybrid stochastic predictive approach incorporating lane change," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 136–148, 2023.

[29] X. Niu and E. Wei, "Fedhybrid: A hybrid federated optimization method for heterogeneous clients," *IEEE Trans. Signal Process.*, vol. 71, pp. 150–163, 2023.

[30] R. Yang, Z. Zhang, X. Zhang, C. Li, Y. Huang, and L. Yang, "Meta-learning for beam prediction in a dual-band communication system," *IEEE Trans. Commun.*,

vol. 71, no. 1, pp. 145–157, 2023.