Computational Intelligence and Neuroscience

Advanced Signal Processing and Adaptive Learning Methods

Lead Guest Editor: Zoran Peric
Guest Editors: Vlado Delic, Zoran Stamenkovic, and David Pokrajac

Advanced Signal Processing and Adaptive Learning Methods

Computational Intelligence and Neuroscience

Advanced Signal Processing and Adaptive Learning Methods

Lead Guest Editor: Zoran Peric
Guest Editors: Vlado Delic, Zoran Stamenkovic, and David Pokrajac

Copyright © 2019 Hindawi Limited. All rights reserved.

This is a special issue published in "Computational Intelligence and Neuroscience." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Chief Editor

Andrzej Cichocki, Japan

Editorial Board

Ricardo Aler, Spain
Amparo Alonso-Betanzos, Spain
Pietro Aricò, Italy
Hasan Ayaz, USA
Sylvain Baillet, Canada
Theodore W. Berger, USA
Daniele Bibbo, Italy
Vince D. Calhoun, USA
Francesco Camastra, Italy
Michela Chiappalone, Italy
Jens Christian Claussen, United Kingdom
Silvia Conforto, Italy
Justin Dauwels, Singapore
Christian W. Dawson, United Kingdom
Carmen De Maio, Italy
Sergio Decherchi, Italy
Paolo Del Giudice, Italy
Maria Jose del Jesus, Spain
Arnaud Delorme, France
Thomas DeMarse, USA
Anastasios D. Doulamis, Greece
António Dourado, Portugal
Steven L. Fernandes, USA
Juan Carlos Fernández, Spain
Piotr Franaszczuk, USA
Leonardo Franco, Spain
Paolo Gastaldo, Italy
Samanwoy Ghosh-Dastidar, USA
Manuel Graña, Spain
Rodolfo E. Haber, Spain
Dominic Heger, Germany
Stephen Helms Tillery, USA
José Alfredo Hernández-Pérez, Mexico
Luis Javier Herrera, Spain
Etienne Hugues, USA
Ryotaro Kamimura, Japan
Pasi A. Karjalainen, Finland
Elpida Keravnou, Cyprus
Raşit Köker, Turkey
Dean J. Krusienski, USA
Fabio La Foresta, Italy
Antonino Laudani, Italy
Maciej Lawrynczuk, Poland
Mikhail A. Lebedev, USA
Cheng-Jian Lin, Taiwan
Giosuè Lo Bosco, Italy
Ezequiel López-Rubio, Spain
Bruce J. MacLennan, USA
Reinoud Maex, Belgium
Kezhi Mao, Singapore
Sergio Martinoia, Italy
Laura Marzetti, Italy
Elio Masciari, Italy
Paolo Massobrio, Italy
Gerard McKee, Nigeria
Michele Migliore, Italy
Paulo Moura Oliveira, Portugal
Debajyoti Mukhopadhyay, India
Klaus Obermayer, Germany
Karim G. Oweiss, USA
Massimo Panella, Italy
Fivos Panetsos, Spain
David M Powers, Australia
Sandhya Samarasinghe, New Zealand
Saeid Sanei, United Kingdom
Friedhelm Schwenker, Germany
Victor R. L. Shen, Taiwan
Fabio Solari, Italy
Jussi Tohka, Finland
Carlos M. Travieso-González, Spain
Lefteri Tsoukalas, USA
Pablo Varona, Spain
Roberto A. Vazquez, Mexico
Meel Velliste, USA
Mario Versaci, Italy
Francois B. Vialatte, France
Thomas Villmann, Germany
Ivan Volosyak, Germany
Cornelio Yáñez-Márquez, Mexico
Michal Zochowski, USA
Rodolfo Zunino, Italy

Contents

Advanced Signal Processing and Adaptive Learning Methods
Zoran Perić, Vlado Delić, Zoran Stamenković, and David Pokrajac
Editorial (2 pages), Article ID 5428615, Volume 2019 (2019)

RnRTD: Intelligent Approach Based on the Relationship-Driven Neural Network and Restricted Tensor Decomposition for Multiple Accusation Judgment in Legal Cases
Xiaoding Guo, Hongli Zhang, Lin Ye, and Shang Li
Research Article (18 pages), Article ID 6705405, Volume 2019 (2019)

Adaptive Learning Emotion Identification Method of Short Texts for Online Medical Knowledge Sharing Community
Dan Gan, Jiang Shen, and Man Xu
Research Article (10 pages), Article ID 1604392, Volume 2019 (2019)

Speech Technology Progress Based on New Machine Learning Paradigm
Vlado Delić, Zoran Perić, Milan Sečujski, Nikša Jakovljević, Jelena Nikolić, Dragiša Mišković, Nikola Simić, Siniša Suzić, and Tijana Delić
Review Article (19 pages), Article ID 4368036, Volume 2019 (2019)

Steady-State Motion Visual Evoked Potential (SSMVEP) Enhancement Method Based on Time-Frequency Image Fusion
Wenqiang Yan, Guanghua Xu, Longting Chen, and Xiaowei Zheng
Research Article (14 pages), Article ID 9439407, Volume 2019 (2019)

Concurrent, Performance-Based Methodology for Increasing the Accuracy and Certainty of Short-Term Neural Prediction Systems
Miljana Milić, Jelena Milojković, Ivan Marković, and Petar Nikolić
Research Article (12 pages), Article ID 9323482, Volume 2019 (2019)

Product Module Network Modeling and Evolution Analysis
Hu Qiao, Zhaohui Xu, Jiang He, and Ying Xiang
Research Article (8 pages), Article ID 2186916, Volume 2019 (2019)

Deep Learning Network for Multiuser Detection in Satellite Mobile Communication System
Guan Qing yang, Wu Shuang, and He Ya-Ru
Research Article (11 pages), Article ID 8613639, Volume 2019 (2019)

Using Morphological Data in Language Modeling for Serbian Large Vocabulary Speech Recognition
Edvin Pakoci, Branislav Popović, and Darko Pekar
Research Article (8 pages), Article ID 5072918, Volume 2019 (2019)

KeratoDetect: Keratoconus Detection Algorithm Using Convolutional Neural Networks
Alexandru Lavric and Popa Valentin
Research Article (9 pages), Article ID 8162567, Volume 2019 (2019)

Image Processing-Based Recognition of Wall Defects Using Machine Learning Approaches and Steerable Filters

Nhat-Duc Hoang

Research Article (18 pages), Article ID 7913952, Volume 2018 (2018)

Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 5428615, 2 pages
https://doi.org/10.1155/2019/5428615

Editorial

Advanced Signal Processing and Adaptive Learning Methods

Zoran Perić,[1] Vlado Delić,[2] Zoran Stamenković,[3] and David Pokrajac[4]

[1]*University of Niš, Niš, Serbia*
[2]*University of Novi Sad, Novi Sad, Serbia*
[3]*IHP–Leibniz-Institut für Innovative Mikroelektronik, Frankfurt, Germany*
[4]*Boeing, Everett, WA, USA*

Correspondence should be addressed to Zoran Perić; zoran.peric@elfak.ni.ac.rs

Received 25 August 2019; Accepted 29 August 2019; Published 3 November 2019

Copyright © 2019 Zoran Perić et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Research in the areas of signal processing and artificial intelligence (and developed methods and algorithms) has become increasingly important in the last two decades. In this special issue, we present new ideas and hybrid approaches based on techniques from the aforementioned scientific areas encouraging researchers from different fields to adopt them in accomplishing complex multidisciplinary tasks. This special issue includes a set of novel contributions, which covers a wide range of advanced signal processing techniques and adaptive learning methods for various engineering purposes.

In the paper entitled "RnRTD: Intelligent Approach Based on the Relationship-Driven Neural Network and Restricted Tensor Decomposition for Multiple Accusation Judgment in Legal Cases," X. Guo et al. describe a new method for judging multiple crimes in legal cases. This is a multilabel classification, and it is based on the relationship-driven recurrent neural network (rdRNN) and restricted tensor decomposition (RTD). The authors have demonstrated that the proposed RTD layer and the relation-driven cyclic neural network have remarkable optimization effects on various deep neural network algorithms.

Next, a novel emotion identification method based on mutual information feature weight, which captures the correlation and redundancy of features, is presented in the paper entitled "Adaptive Learning Emotion Identification Method of Short Texts for Online Medical Knowledge Sharing Community" by D. Gan et al.

The paper entitled "Speech Technology Progress Based on New Machine Learning Paradigm" by V. Delić et al. provides an overview of speech technologies development as a typical signal processing area. The authors provide an analysis of the nature of speech signal and processing, corresponding machine learning algorithms, and applied computational intelligence in order to give an insight into several fields, covering speech production and auditory perception, cognitive aspects of speech communication and language understanding, both speech recognition and text-to-speech synthesis in more detail, and consequently the main directions in development of spoken dialogue systems. Additionally, the article discusses the concepts and recent advances in speech signal compression, coding, and transmission, including cognitive speech coding.

"Using Morphological Data in Language Modeling for Serbian Large Vocabulary Speech Recognition" is another paper in the area of speech signal processing, presented by E. Pakoci et al. In the paper, the authors have demonstrated that using additional morphological knowledge for language model training can solve a large part of problems for highly inflective languages such as Serbian.

A product module network design based on complex network theory is described in the paper entitled "Product Module Network Modeling and Evolution Analysis" by H. Qiao et al. In order to analyze the change in the product module network caused by module evolution, a BBV (Barrat–Barthelemy–Vespignani) model of the product module network is established to dynamically determine the brittle risk of the product module network.

"Steady-State Motion Visual Evoked Potential (SSMVEP) Enhancement Method Based on Time-Frequency Image Fusion" is the title of the paper by W. Yan et al., in which the authors propose an SSMVEP enhancement method based on

T-F image fusion in order to explore whether T-F domain analysis can achieve better fusion effects than time-domain analysis. The authors have demonstrated that the key to the T-F image fusion algorithm is the fusion of the wavelet low-frequency components and that the online performance of the T-F image fusion method is better than that of the traditional spatial filtering methods, which indicates that the proposed method is feasible to fuse SSMVEP in the T-F domain.

The paper entitled "Deep Learning Network for Multiuser Detection in Satellite Mobile Communication System" by G. Q. Yang et al. proposes a multiuser detection (MUD) algorithm based on deep learning network for the satellite mobile communication system. The authors have demonstrated that through establishing a typical satellite communication system simulation platform, compared with the orthogonal signal tracking (OMP) and iterative sorting least squares (IORLS) algorithms, the proposed deep learning network algorithm has better performance in different conditions of SNR, CINR, and carrier frequency offset interference.

A novel methodology for increasing the predictions accuracy of different ANN-based systems is presented in the paper entitled "Concurrent, Performance-Based Methodology for Increasing the Accuracy and Certainty of Short-Term Neural Prediction Systems" by M. Milić et al. The authors have tested the method on GNI forecasting at national economy level, municipal traffic volume forecasting, and suburban daily electric load consumption forecasting.

The paper "KeratoDetect: Keratoconus Detection Algorithm Using Convolutional Neural Networks" by A. Lavric et al. proposes an algorithm facilitating the diagnostics of keratoconus disease. The algorithm analyzes the corneal topography of the eye using a convolutional neural network (CNN) that is able to extract and learn the features of a keratoconus eye. The authors have demonstrated that the algorithm can assist the ophthalmologists in rapid screening of patients, thus reducing diagnostic errors and facilitating treatment.

Finally, N.-D. Hoang proposes an automatic image processing approach for periodically evaluating the condition of wall structures in the paper entitled "Image Processing-Based Recognition of Wall Defects Using Machine Learning Approaches and Steerable Filters." In order to extract useful features from digital images, steerable filters and projection integrals are employed. The author has demonstrated that the proposed method can achieve a good classification performance with a high classification accuracy, and as such, it represents a promising aid in periodic building surveys by maintenance agencies.

By providing a variety of applications, from traditional signal processing areas such as speech, image, and medical signal processing to the applications for analysis of legal matters and civil engineering, we have shown that whatever our primary research area is, machine learning and signal processing techniques can significantly improve the performance of existing methods and enable emersion of novel approaches. Application of such advanced techniques may also contribute to the progress in a number of other areas, which is an additional mission of this special issue.

Conflicts of Interest

The guest editors declare no (existing and anticipated) conflicts of interest. The guest editors exercised their professional judgment in the review process, but their opinion does not necessarily reflect the official standings of their employers or organizations they may belong to.

Zoran Perić
Vlado Delić
Zoran Stamenković
David Pokrajac

Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 6705405, 18 pages
https://doi.org/10.1155/2019/6705405

Research Article

RnRTD: Intelligent Approach Based on the Relationship-Driven Neural Network and Restricted Tensor Decomposition for Multiple Accusation Judgment in Legal Cases

Xiaoding Guo, Hongli Zhang, Lin Ye, and Shang Li

Harbin Institute of Technology, Harbin, China

Correspondence should be addressed to Hongli Zhang; zhanghongli@hit.edu.cn

Received 14 January 2019; Accepted 17 June 2019; Published 7 July 2019

Guest Editor: David Pokrajac

Copyright © 2019 Xiaoding Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The use of intelligent judgment technology to assist in judgment is an inevitable trend in the development of judgment in contemporary social legal cases. Using big data and artificial intelligence technology to accurately determine multiple accusations involved in legal cases is an urgent problem to be solved in legal judgment. The key to solving these problems lies in two points, namely, (1) characterization of legal cases and (2) classification and prediction of legal case data. Traditional methods of entity characterization rely on feature extraction, which is often based on vocabulary and syntax information. Thus, traditional entity characterization often requires extensive energy and has poor generality, thus introducing a large amount of computation and limitation to subsequent classification algorithms. This study proposes an intelligent judgment approach called RnRTD, which is based on the relationship-driven recurrent neural network (rdRNN) and restricted tensor decomposition (RTD). We represent legal cases as tensors and propose an innovative RTD method. RTD has low dependence on vocabulary and syntax and extracts the feature structure that is most favorable for improving the accuracy of the subsequent classification algorithm. RTD maps the tensors, which represent legal cases, into a specific feature space and transforms the original tensor into a core tensor and its corresponding factor matrices. This study uses rdRNN to continuously update and optimize the constraints in RTD so that rdRNN can have the best legal case classification effect in the target feature space generated by RTD. Simultaneously, rdRNN sets up a new gate and a similar case list to represent the interaction between legal cases. In comparison with traditional feature extraction methods, our proposed RTD method is less expensive and more universal in the characterization of legal cases. Moreover, rdRNN with an RTD layer has a better effect than the recurrent neural network (RNN) only on the classification and prediction of multiple accusations in legal cases. Experiments show that compared with previous approaches, our method achieves higher accuracy in the classification and prediction of multiple accusations in legal cases, and our algorithm is more interpretable.

1. Introduction

In contemporary society, the demand for big data assistance in the judgment of legal cases, such as case intelligence research [1] and judgment [2], big data comprehensive supervision, and assistance in handling legal cases, is increasing with the development of big data and artificial intelligence technology. Researchers are committed to creating an "intelligent legal case judgment" project that combines big data and artificial intelligence. Legal case multiaccusation judgment business is an important part of the realization of such a project. Legal case multiaccusation judgment technology fully applies big data and artificial intelligence technology to service judgment making, legal case handling [3], and facilitation of the public. Big data provides judgments with recognized standards for judging legal cases and avoids the occurrence of different judgment results in similar legal cases. Artificial intelligence technology avoids the subjectivity of human beings, performs scientific and accurate analyses of cases from the perspective of cases and laws, and helps judges make objective judgment in legal cases.

The solution to using big data and artificial intelligence technology to accurately judge multiple accusations in

different legal cases involves two main points, namely, (1) construction of a comprehensive and accurate characterization method of legal cases and (2) realization of a classification and prediction algorithm for multiple accusations involved in a large number of legal cases. Figure 1 shows the process of multiaccusation classification for legal cases. Traditional methods of entity characterization are often used to model an entity by tagging it. However, these feature extraction methods are highly dependent on the vocabulary and syntax in the entity data set and require heavy manpower and material resources. The generality of the tagged model is poor. In addition, feature extraction methods based on vocabulary and syntax require strong expert knowledge as support. The resulting entity characterization considerably limits the subsequent classification algorithm, and the algorithm's accuracy becomes highly volatile.

This study proposes an intelligent legal case judgment technique called RnRTD, which is based on the relationship-driven recurrent neural network (rdRNN) and restricted tensor decomposition (RTD). Figure 2 shows the framework of our approach. We present legal case data as tensor χ and propose an RTD technique. RTD is less dependent on vocabulary and syntax than traditional feature extraction methods, and it focuses more on extracting the information of potential structures in legal case tensors. RTD maximizes the accuracy of rdRNN by combining text and structural information. RTD maps legal case tensor χ into specified feature space $\mathbb{Z}$, which decomposes the original tensor χ into core tensor $\widetilde{\chi}$ and its corresponding feature matrix set $\{C_k\}$ under the restricted condition η in RTD. The obtained core tensor $\widetilde{\chi}$ represents the tensor structure information that is most helpful in improving the accuracy of the rdRNN classification algorithm. Core tensor $\widetilde{\chi}$ can be interpreted as the most advantageous feature structure in χ for rdRNN. RTD is an important feature extraction and dimensionality reduction operation. This study uses rdRNN to update and optimize restricted condition η in RTD iteratively so that its feature space $\mathbb{Z}$ continually approaches an ideal region, thus enabling rdRNN to achieve an optimal effect in the classification of multiple accusations in legal cases.

Compared with traditional feature extraction methods, RTD obtains legal case characterization containing tensor element and structural information that is more conducive for improving the accuracy of the rdRNN classification algorithm, and it has lower dependence on vocabulary, syntax, and expert knowledge. That is, the RTD legal case characterization model has better universality and fewer requirements on the dataset format in comparison with traditional feature extraction methods. Compared with the direct use of the original legal case tensor χ as the input of the RNN classification algorithm, rdRNN with an RTD layer has a better effect on the classification of multiple accusations in legal cases. The main reason is that rdRNN constantly updates and optimizes RTD restricted condition η, thereby enabling RTD to point to feature space $\mathbb{Z}$ where rdRNN has the best effect in legal case classification.

The main contributions of this study are summarized in the following points:

(i) *This study uses a new method of characterizing legal cases.* This study expresses a legal case as a tensor and proposes an RTD method that maps the original legal case tensor into a new feature space. RTD extracts the favorable tensor structure and text information for the subsequent classification algorithm from the original legal case tensor. RTD also extracts valuable tensor features and reduces tensor dimensions. The core tensor obtained by RTD is interpreted as the most valuable tensor structure and textual feature information extracted from the original legal case tensor for the rdRNN classification algorithm.

(ii) *This study proposes rdRNN, which is a new approach for intelligent judgment of multiple accusations in legal cases.* We add a new gate and a similar case list to control the interaction between tensors of legal cases on the basis of the original neural networks. rdRNN is particularly used for the intelligent judgment of multiple accusations in legal cases. It fully considers the impact of the relationship between legal cases on the judgment results of such cases. For example, highly similar legal cases are likely to have similar judgment results and vice versa.

(iii) *This study proposes a neural network-based method for the optimization of the restricted tensor.* The restricted tensor is a bridge between the RTD algorithm and rdRNN. rdRNN controls the tensor decomposition process by optimizing the restricted tensor, which guides the core tensor along the direction that is most conducive for improving the accuracy of the classification model. We derive the partial derivative of the loss function in rdRNN for the restricted tensor and realize the optimization operation of the neural network for the restricted tensor.

Section 2 gives the recent research progress on the classification of multiple crimes in legal cases. Section 3 introduces related definitions and the concepts involved in this study. Section 4 introduces the proposed approach for the judgment of multiple accusations in legal cases. Section 5 provides the experimental results and analysis of this study, and Section 6 presents a detailed discussion of the proposed method.

2. Related Work

With the advent of the era of big data and the development of artificial intelligence technology [4], the emergence of deep neural networks provides great prospects for accurate classification and prediction [5]. Neural network-based knowledge representation and reasoning methods enable deep learning approaches to be applied to many scenarios [6]. For the legal field, the combination of artificial intelligence and law has become an inevitable trend [7]. However, current research in this area mainly focuses on

legal case modeling [8], legal case document retrieval [9], legal consultation question-and-answer systems [10], and legal case similarity reasoning work [2]. Little research has been conducted on the multiaccusation determination of cases in the legal field.

Bartolini et al. proposed a semantic annotation method for indexing and retrieving legal texts [11]. The method uses a specific segment extraction and text classification algorithm to automatically semantically mark legal documents. Aleven developed a computational model based on artificial intelligence algorithms and professional legal knowledge [2]. The model determines the correlation between cases based on the context and problem scenarios of the case. Joshi et al. proposed a text mining method for electronic evidence review of legal cases [12]. The method uses semantic topic and text classification technology to repeatedly detect the feature vocabulary in legal documents and then automatically segments and screens the documents, avoiding the manual work of legal analysts.

Sulea et al. proposed a legal case judgment system based on SVM classifier [13]. The method uses machine learning techniques to predict the legal field to which the legal case belongs and the outcome of its judgment. By accurately extracting the features of legal cases, the method can roughly predict the specific date of the case. Brninghaus and Ashley proposed a text classification method based on facts of legal cases [14]. The method uses artificial intelligence algorithms and legal background knowledge to predict the outcome of legal cases. The method extracts facts of legal cases, indexes and models them according to the features, and finally completes the classification of legal cases.

The critical part for the prediction of legal case judgments is case modeling and case classification. Traditional text modeling methods are based on feature tags, which rely heavily on the syntax and semantic information of the source data. Labeling features requires a lot of manual work and expert knowledge. Therefore, the text classification algorithm formed on this basis is not scalable, and the accuracy is highly volatile.

3. Preliminaries

This section introduces the related methods, definitions, and background knowledge involved in this study. Section 3.1 presents the basic notations and definitions. Section 3.2 provides a formal representation of the tensor decomposition problem. Section 3.3 introduces the calculation process of forward propagation in bidirectional long short-term memory (Bi-LSTM). Section 3.4 presents a formal description of the problem about intelligent legal judgment to be solved in this study.

3.1. Definitions and Notations. This section describes the relevant notations and definitions required in this work. Tensors are actually multidimensional matrices [15], which we represent in Euler script letters, such as χ and ν. We refer to the dimensions as tensor modes and to the number of a tensors modes as order. We describe the scalars in lowercase letters (such as a, b) and the vectors in boldface lowercase letters (such as c, d). We declare the matrices in capital letters, such as A and B. We use A^{T} to represent the transpose of matrix A. We express the identity matrix as I, the identity tensor as τ, and the matrix with all elements of 1 as 1. Table 1 shows all the required notations and definitions.

Definition 1 (outer product). The outer product of vectors $\mathbf{a} \in \mathbb{R}^I$ and $\mathbf{b} \in \mathbb{R}^J$ is denoted as $A = \mathbf{a} \circ \mathbf{b}$, where $A \in \mathbb{R}^{I \times J}$ and $A(i, j) = \mathbf{a}(i)\mathbf{b}(j)$.

Definition 2 (elementwise multiplication). The elementwise multiplication of vectors $\mathbf{a} \in \mathbb{R}^I$ and $\mathbf{b} \in \mathbb{R}^I$ is denoted as $A = \mathbf{a} * \mathbf{b}$, where $A \in \mathbb{R}^I$ and $A(i) = \mathbf{a}(i)\mathbf{b}(i)$. In another case, the elementwise multiplication of vector $\mathbf{a} \in \mathbb{R}^I$ and matrix $A \in \mathbb{R}^{I \times J}$ is denoted as $Z = \mathbf{a} * A$, where $Z \in \mathbb{R}^{I \times J}$ and $Z(i, j) = \mathbf{a}(i)A(i, j)$.

Definition 3 (Kronecker product). Given vectors $\mathbf{a} \in \mathbb{R}^I$ and $\mathbf{b} \in \mathbb{R}^J$, their Kronecker product is denoted as $S = \mathbf{a} \otimes \mathbf{b}$, where $S \in \mathbb{R}^{IJ}$ and $S = [\mathbf{a}(1)\mathbf{b}^{\mathrm{T}}, \mathbf{a}(2)\mathbf{b}^{\mathrm{T}}, \cdots, \mathbf{a}(I)\mathbf{b}^{\mathrm{T}}]^{\mathrm{T}}$. Given matrices $A \in \mathbb{R}^{I \times J}$ and $B \in \mathbb{R}^{P \times Q}$, their Kronecker product is denoted as $A \otimes B$.

$$\begin{aligned} A \otimes B = [&A(:, 1) \otimes B(:, 1) A(:, 1) \otimes B(:, 2) \\ &\cdots A(:, J) \otimes B(:, Q-1) A(:, J) \otimes B(:, Q)]. \end{aligned} \tag{1}$$

Definition 4 (Khatri–Rao product). Given matrices $A \in \mathbb{R}^{I \times R}$ and $B \in \mathbb{R}^{J \times R}$, their Khatri–Rao product is denoted as $A \odot B$, which is calculated by combining the Kronecker product of each corresponding column in A and B, that is

$$A \odot B = [A(:, 1) \otimes B(:, 1) \cdots A(:, r) \otimes B(:, r) \cdots A(:, R) \otimes B(:, R)]. \tag{2}$$

Definition 5 (n-mode matricization). Given an N-mode tensor χ, $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. χ can be matrixed into N forms according to each mode. We denote the n-mode matricization of χ as $\chi_{(n)}$, where $\chi_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$. $\chi_{(n)}$ is obtained by keeping the nth mode unchanged while expanding and concatenating the slices of the remaining modes into a matrix.

Definition 6 (Frobenius norm of a tensor). Given an N-mode tensor χ, $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the Frobenius norm of χ is denoted as

$$||\chi||_{\mathrm{F}} = \sqrt{\sum_{i_1}^{I_1} \sum_{i_2}^{I_2} \cdots \sum_{i_N}^{I_N} \chi(i_1, i_2, \ldots, i_N)^2}. \tag{3}$$

Definition 7 (n-mode stretch). Given an N-mode tensor ν, $\nu \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$, and a weight matrix W, $W \in \mathbb{R}^{J_n \times \prod_{k \neq n}^{N} J_k}$. The n-mode stretch between ν and W is expressed as $\upsilon \times^n W = \kappa$, where $\kappa \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$.

Figure 1: Multiaccusation classification for legal cases.

Figure 2: Framework of the proposed RnRTD.

Table 1: Symbols.

Symbol	Definition
$\chi, X, \mathbf{x}, x$	Tensor, matrix, vector, scalar
A^{T}	Transpose of matrix A
$\chi(i_1, \ldots, i_N)$	The $(i_1, \ldots, i_N)$th entry of χ, same for vectors and matrices
$A(:, i)$	The ith column of matrix A
$\circ$	Outer product
$*$	Elementwise multiplication
$\otimes$	Kronecker product
$\odot$	Khatri–Rao product
$\times^n$	The n-mode stretch
$\times_n$	The n-mode product
$\chi_{(n)}$	The n-mode matricization of tensor χ
χ_{F}	The Frobenius norm of tensor χ

$$\kappa(j_1, j_2, \ldots, j_n, \ldots, j_N) = W\left(j_1, \prod_{k \neq n}^{N} j_k\right) v(j_1, j_2, \ldots, j_n, \ldots, j_N). \tag{4}$$

Definition 8 (n-mode product). Given an N-mode tensor $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a matrix $C \in \mathbb{R}^{I_n \times J}$, their n-mode product is denoted as $\lambda = \chi \times_n C$, $\lambda \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$.

$$\lambda(i_1, \ldots, i_{n-1}, j, i_{n+1}, \ldots, i_N) = \sum_{m=1}^{I_n} \chi(i_1, \ldots, i_{n-1}, m, i_{n+1}, \ldots, i_N) C(m, j). \tag{5}$$

Figure 3: Tensor decomposition.

3.2. Tensor Decomposition. Many tensor decomposition methods, such as PARAFAC and Tucker decomposition, are currently available [15]. As shown in Figure 3, tensor decomposition methods decompose the original tensor into a core tensor and a series of corresponding factor matrices. The essence of tensor decomposition is to approximate the original tensor by using the product of the core tensor and the factor matrices. The mathematical description of tensor decomposition is as follows:

Given an N-mode tensor χ, $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The following formula can be obtained by using the tensor decomposition method:

$$\chi \approx \tau \times_1 C_1 \times_2 C_2 \times \cdots \times_N C_N, \tag{6}$$

where τ is the core tensor, $\tau \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$, and $\{C_n\}$ is the corresponding factor matrix set, $C_n \in \mathbb{R}^{J_n \times I_n}$. Each element in $\{C_n\}$ is a column orthogonal matrix. τ and $\{C_n\}$ also minimize function φ, where

$$\varphi = \left\| \chi - \tau \prod_{n}^{N} \times_n C_n \right\|_{\mathrm{F}}. \tag{7}$$

3.3. Bi-LSTM. RNNs have far-reaching implications for the study of sequence data [16]. The nodes between the hidden layers of RNN are connected [17], that is, the input of the hidden layer contains not only the output of the input layer but also the output of the hidden layer at the last moment. In theory, RNN can process sequence data of any length. However, gradient disappearance or gradient explosion often occurs when RNN deals with long-distance dependence, thereby making RNN training difficult. The hidden layers of the original RNN has only one kind of state, which is very sensitive to short-term inputs. Long short-term memory (LSTM) deals with long-distance dependence by increasing the long-term memory state in the original RNN [18].

As shown in Figure 4, we represent the input value of LSTM at time t as x_t, the output value from the previous moment $t-1$ as h_{t-1}, and the long-term unit state at time $t-1$ as c_{t-1}. We record the unit status entered at time t as $\tilde{c}_t$. The output value of LSTM at time t comprises two parts, namely, the output value of LSTM at current time h_t and the unit state of current time c_t. LSTM sets up three control gates, which are forget, input, and output, to control the long-term unit state c. The forget gate is used to determine

Figure 4: Framework of LSTM.

how much of the long-term unit state at the previous moment is retained at the current moment. For example, the forget gate f_t at time t determines the weight of c_{t-1} in the calculation of c_t. The input gate is used to determine how much of the input of LSTM is retained in the current long-term unit state. For example, input gate i_t determines the weight that x_t takes while calculating c_t. The output gate is used to determine how much the long-term unit state at the current moment contributes to the output of LSTM at the current time. For example, output gate o_t determines the influence of the value of c_t on h_t.

The process of forward propagation calculation in LSTM is described as follows:

$$\begin{aligned} f_t &= \sigma\left(w_f \cdot \left[h_{t-1}, x_t\right]^{\mathrm{T}} + b_f\right), \\ i_t &= \sigma\left(w_i \cdot \left[h_{t-1}, x_t\right]^{\mathrm{T}} + b_i\right), \\ \tilde{c}_t &= \tanh\left(w_c \cdot \left[h_{t-1}, x_t\right]^{\mathrm{T}} + b_c\right), \\ o_t &= \sigma\left(w_o \cdot \left[h_{t-1}, x_t\right]^{\mathrm{T}} + b_o\right). \end{aligned} \tag{8}$$

The long-term unit state c_t at current time t is calculated by f_t, i_t, c_{t-1} and $\tilde{c}_t$, and the final output of LSTM h_t is calculated by o_t and c_t. That is,

$$\begin{aligned} c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t, \\ h_t &= o_t * \tanh\left(c_t\right), \end{aligned} \tag{9}$$

where h_{t-1} is the output of LSTM at time $t-1$, x_t is the input of LSTM at time t, σ is the sigmoid function, which is our selected activation function in LSTM, $\tilde{c}_t$ is the unit state input at time t, w_f, w_i, and w_o are the weight matrices of the

forget gate f_t, the input gate i_t, and the output gate o_t, respectively, and b_f, b_i, and b_o are the bias matrices of f_t, i_t, and o_t, respectively. The activation function used in calculating $\tilde{c}_t$ is the hyperbolic tangent function, where w_c is the weight matrix and b_c is the bias term.

Bi-LSTM is a bidirectional RNN [19]. The unit state of the hidden layer in Bi-LSTM is calculated from the outputs of forward and backward LSTM. We define the output unit state of Bi-LSTM at time t as $h_{\text{Bi-LSTM}_t}$, the output unit state of forward LSTM as h_{f_t}, and the output unit state of backward LSTM as h_{b_t}. The aforementioned forward propagation formula of LSTM implies that

$$\begin{aligned} h_{f_t} &= \text{LSTM}^{\rightarrow}\left(h_{f_{t-1}}, x_t\right), \\ h_{b_t} &= \text{LSTM}^{\leftarrow}\left(h_{b_{t+1}}, x_t\right), \\ h_{\text{Bi-LSTM}_t} &= \left[h_{f_t}, h_{b_t}\right]. \end{aligned} \tag{10}$$

3.4. Problem Description

Problem 1. We express the legal case as a tensor and classify the legal case according to the judgment result. The category of each legal case is indicated by a scalar, such as r. Given a legal case dataset Ω that contains legal cases with judgment results, $\Omega = \{(\chi^{(1)}, r^{(1)}), (\chi^{(2)}, r^{(2)}), \cdots, (\chi^{(N)}, r^{(N)})\}$. $\chi^{(n)}$ represents the nth legal case in the legal case dataset Ω. $r^{(n)}$ indicates the type of legal judgment result that corresponds to the nth legal case. Our goal is to train a case classification model $\phi(\chi)$ that can classify legal cases based on their judgment results.

In this study, legal cases are represented as three-dimensional tensors. As shown in Figure 5, the first dimension represents the basic components of the case, such as the defendant's statement, the plaintiff's statement, the public prosecution, and the court's trial. On this basis, the matrix slice that contains the last two dimensions represents the matrix form of the corresponding legal case component. The matrix slice is composed of the accumulation of word vectors inside the legal case component. Generally, case components are matrixed instead of including the word vectors of all the words in the matrix. We selectively extract words that are valuable for the legal case classification. These words can be divided into two categories. The first category usually includes nouns or pronouns, such as characters, times, places, and objects; the second category usually comprises adjectives, numerals, or verbs, such as the means of committing accusations, the degree of harm, and the number of accusations.

4. Our Approach

This study proposes RnRTD for the multiaccusation determination of legal cases. Figure 6 shows the RnRTD framework. First, we extract core tensors from the original tensors using the RTD method. The core tensor approximates the restricted tensor in terms of the tensor structure and elements. Second, we use rdRNN to optimize the restricted tensor so that it guides the core tensor along the direction that is most conducive for improving the accuracy of the classification model.

4.1. RTD Method. This study proposes a new tensor decomposition method called RTD method. The inputs of the RTD algorithm include the restricted condition tensor η and tensor χ that represents the legal case. The RTD outputs include core tensor $\tilde{\chi}$ and its corresponding factor matrix sets, namely, $\{C_k\}$ and $\{D_k\}$. RTD decomposes χ into a core tensor $\tilde{\chi}$ under the action of the restricted condition η. $\tilde{\chi}$ is approximated to η in terms of tensor structure and internal element values. RTD can be interpreted as a mapping of the original tensor χ to the core tensor $\tilde{\chi}$. In short, RTD achieves directional decomposition of tensors and extracts vital information from tensors while reducing their dimension. In this study, we define core tensor $\tilde{\chi}$ as the most favorable tensor structure and element value information for the subsequent legal case classification algorithm, namely, rdRNN. On this basis, we construct a deep neural network model for RnRTD that is dedicated to legal intelligence judgments.

RTD decomposes the original tensor under the restricted condition so that the obtained core tensor constantly approaches the restricted tensor in terms of tensor structure and element value. In Figure 7, the formal description of the problems to be solved by the RTD algorithm is shown as Problems 1 and 2.

Problem 2. Given tensor $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$, restricted tensor $\eta \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_K}$, and its weight w_η, we derive two factor matrix sets, namely, $\{C_k\}$ and $\{D_k\}$, $C_k \in \mathbb{R}^{I_k \times H_k}$, $D_k \in \mathbb{R}^{J_k \times H_k}$, that $\{C_k\}$ and $\{D_k\}$ minimize the following function:

$$\phi = \left\| \chi \prod_k^K \times_k C_k - W_\eta \times^n \eta \prod_k^K \times_k D_k \right\|_{\text{F}}^2. \tag{11}$$

Matrix W_η is preset according to the legal case, $W_\eta \in \mathbb{R}^{H_n \times \prod_{k \neq n}^K H_k}$. The elements in sets $\{C_k\}$ and $\{D_k\}$ are orthogonal matrices, that is, they meet the following conditions. For any elements C_k and D_k in sets $\{C_k\}$ and $\{D_k\}$,

$$\begin{cases} C_k C_k^{\text{T}} = I, \\ C_k^{\text{T}} C_k = I, \end{cases} \quad \begin{cases} D_k D_k^{\text{T}} = I, \\ D_k^{\text{T}} D_k = I. \end{cases} \tag{12}$$

In this study, we use the alternating least squares (ALS) algorithm to determine the solution of the objective function ϕ. The ALS algorithm can be divided into four steps: (1) randomly pick a variable as a parameter and randomly generate the values of other variables, (2) determine the partial derivative of the loss function ϕ in the specified parameter while fixing the values of other variables, (3) set the partial derivative of ϕ to the specified parameter as zero and calculate the value of the specified parameter, and (4) select another variable as a parameter and return to Step (2). The ALS algorithm continues to iterate Steps (2), (3),

Figure 5: Original tensor that represents the legal case.

Figure 6: Framework of the RnRTD method.

and (4) until the error of the loss function ϕ reaches the tolerable upper limit.

Problem 2 needs to be solved using Lemma 1. The specific definition and proof of Lemma 1 are provided as follows.

Lemma 1. *Given function* $tr(\alpha^T\alpha) = \sum_{i_1}^{I_1}\sum_{i_2}^{I_2}\cdots\sum_{i_n}^{I_N}\alpha(i_1, i_2, \ldots, i_n)^2$, $\alpha \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$, *$\alpha$ can be a vector, matrix, and tensor. The target function* $\varphi = tr((\chi\prod_k^K \times_k C_k)^T(\chi\prod_k^K \times_k C_k))$, *where* $\chi \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_K}$, $C_k \in \mathbb{R}^{I_k\times H_k}$, *and* C_k *satisfy equation (12). For any element* C_{k_0} *in* $\{C_k\}$, $k_0 \in [1, K]$, *the partial differential of function* φ *to* C_{k_0} *is* $\partial\varphi/\partial C_{k_0} = \varepsilon(\chi\prod_{k\neq k_0}^K \times_k C_k)^T(\chi\prod_k^K \times_k C_k)$, *where* ε *is a constant.*

The proof of Lemma 1 is shown in Proof 1.

Proof 1. We use μ to represent $\chi\prod_{k\neq k_0}^K \times_k C_k$; it can can be derived that $\varphi = \mathrm{tr}((\mu\times_{k_0}C_{k_0})^{\mathrm{T}}(\mu\times_{k_0}C_{k_0}))$. We abbreviate the target function $\varphi = \mathrm{tr}((\mu C_{k_0})^{\mathrm{T}}(\mu C_{k_0})) = \mathrm{tr}(C_{k_0}^{\mathrm{T}}\mu^{\mathrm{T}}\mu C_{k_0})$. We use ν to represent $\mu^{\mathrm{T}}\mu$, and we can get that $\varphi = \mathrm{tr}(C_{k_0}^{\mathrm{T}}\nu C_{k_0})$. According to the function derivation rule, we can obtain the following equation: $\partial\varphi/\partial C_{k_0} = \partial\mathrm{tr}(C_{k_0}^{\mathrm{T}}\nu C_{k_0})/\partial C_{k_0} = \nu C_{k_0} + \nu^{\mathrm{T}}C_{k_0}$. Since $\nu = \nu^{\mathrm{T}}$, the calculation formula for the partial derivative of the function φ to C_{k_0} is

$$\frac{\partial\varphi}{\partial C_{k_0}} = 2\nu C_{k_0} = \varepsilon\left(\chi\prod_{k\neq k_0}^{K}\times_k C_k\right)^{\mathrm{T}}\left(\chi\prod_{k}^{K}\times_k C_k\right), \tag{13}$$

where ε is a constant, $\varepsilon = 2$.

According to the iterative process of the ALS algorithm, the precondition for solving the value of $\{C_k\}$ and $\{D_k\}$ which minimize the function ϕ in equation (11) using the ALS algorithm is to calculate the value of $\partial\phi/\partial C_{k_0}$, where $k_0 \in [1, K]$. Equation (11) shows that $\partial\phi/\partial C_{k_0}$ and $\partial\phi/\partial D_{k_0}$ are solved in the same manner. Proof 2 provides mathematical proof of the calculation of $\partial\phi/\partial C_{k_0}$. □

Figure 7: Framework of the RTD method.

Proof 2. We use λ and ϖ to represent $\chi\prod_{k\neq k_0}^{K}\times_k C_k$ and $w_\eta\eta\prod_k^K\times_k D_k$, respectively. According to formula (11), we can obtain that $\phi = \mathrm{tr}((\lambda\times_{k_0}C_{k_0}-\varpi)^{\mathrm{T}}(\lambda\times_{k_0}C_{k_0}-\varpi))$. We abbreviate the aforementioned formula as $\phi = \mathrm{tr}((\lambda C_{k_0}-\varpi)^{\mathrm{T}}(\lambda C_{k_0}-\varpi))$. Then, we can determine the following formula: $\partial\phi/\partial C_{k_0} = (\partial\mathrm{tr}(\varpi^{\mathrm{T}}\varpi)-2\partial\mathrm{tr}(C_{k_0}^{\mathrm{T}}\lambda^{\mathrm{T}}\varpi)+\partial\mathrm{tr}\ (C_{k_0}^{\mathrm{T}}\lambda^{\mathrm{T}}\lambda C_{k_0}))/\partial C_{k_0}$. According to the function derivation rule, we derive that $\partial\mathrm{tr}(\varpi^{\mathrm{T}}\varpi)/\partial C_{k_0}=0$, $(\partial\mathrm{tr}(C_{k_0}^{\mathrm{T}}\lambda^{\mathrm{T}}\varpi)/\ \partial C_{k_0}) = \lambda^{\mathrm{T}}\varpi$. In combination with Lemma 1, we can obtain that $(\partial\mathrm{tr}(C_{k_0}^{\mathrm{T}}\lambda^{\mathrm{T}}\lambda C_{k_0})/\partial C_{k_0}) = 2\lambda^{\mathrm{T}}\lambda C_{k_0}$. Finally, the following formula is determined:

$$\frac{\partial\phi}{\partial C_{k_0}} = -2\lambda^{\mathrm{T}}\varpi + 2\lambda^{\mathrm{T}}\lambda C_{k_0}. \tag{14}$$

We set the value of equation (14) to 0 and obtain that

$$\lambda C_{k_0} = \varpi. \tag{15}$$

Let $Z = \lambda_{(k_0)}^{\mathrm{T}}\varpi_{(k_0)}$, then $C_{k_0}^{\mathrm{T}}Z = (\lambda C_{k_0})_{(k_0)}^{\mathrm{T}}\varpi_{(k_0)} = \varpi_{(k_0)}^{\mathrm{T}}\varpi_{(k_0)}$. By combining Equation (12), we derive that

$$\begin{cases} C_{k_0}^{\mathrm{T}}Z = Z^{\mathrm{T}}C_{k_0}, \\ Z = C_{k_0}Z^{\mathrm{T}}C_{k_0}. \end{cases} \tag{16}$$

We use the SVD matrix decomposition method to decompose Z, and find that $Z = PSQ^{\mathrm{T}}$. P and Q are orthogonal matrices, P is a left singular matrix, Q is a right singular matrix, and S is a diagonal matrix. After this analysis, the following solution can be obtained:

$$C_{k_0} = PQ^{\mathrm{T}}. \tag{17}$$

In summary, according to equations (11)–(17), we can derive a solution of $\partial\phi/\partial C_{k_0}$ and C_{k_0} which are described in equations (14) and (17), respectively. $\partial\phi/\partial D_k$ is calculated in the same manner as $\partial\phi/\partial C_k$. On this basis, we calculate the value of $\{C_k\}$ and $\{D_k\}$, which minimize the objective function ϕ in formula (11), by using the ALS algorithm.

Algorithm 1 shows the solution of Problem 2 by using ALS algorithm. The inputs of Algorithm 1 are χ which represents one legal case, the restricted tensor η, and its weight w_η. In line 2, we randomly initialize the values of $\{C_k\}$, $\{U_k\}$. max_iterations in line 2 represents the maximum number of iterations of ALS algorithm. The function calcu_Z in line 4 corresponds to equation (16). Line 5 and 6 show the calculation process of equation (17).

Another problem to be solved by the RTD algorithm is Problem 3, which is the formal description of the process of tensor decomposition on the original tensor under the action of the restricted tensor and its weight. On the basis of Problem 2, we can obtain factor matrix sets $\{C_k\}$ and $\{D_k\}$, which minimize the value of function ϕ in formula (11) while satisfying formula (12). □

Problem 3. Given a tensor $\chi\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_K}$ and factor matrix sets $\{C_k\}$ and $\{D_k\}$, $C_k\in\mathbb{R}^{I_k\times H_k}$, $D_k\in\mathbb{R}^{J_k\times H_k}$, $\{C_k\}$ and $\{D_k\}$ are derived from Problem 2. A core tensor $\tilde{\chi}$ is determined, where $\tilde{\chi}\in\mathbb{R}^{J_1\times J_2\times\cdots\times J_K}$ and $\tilde{\chi}$ minimize the following target function:

$$F_{\mathrm{RTD}} = \left\|\chi\prod_k^K\times_k C_k - \tilde{\chi}\prod_k^K\times_k D_k\right\|_{\mathrm{F}}^2. \tag{18}$$

Input: Tensor χ which represents the original legal case data, $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$, the restricted tensor η, $\eta \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_K}$, and its weight w_η.
Output: The factor matrix sets $\{C_k\}$ and $\{D_k\}$, $C_k \in \mathbb{R}^{I_k \times H_k}$, $D_k \in \mathbb{R}^{J_k \times H_k}$.
Initialize the factor matrix sets $\{C_k\}$ and $\{D_k\}$;
for $i = 1$ to max_iterations **do**
  // First, pick the elements in the factor set $\{C_k\}$ as variables
  for $k_0 = 1$ to K **do**
    $Z = \text{calcu_}Z(\chi, \eta, \{C_k\}, \{D_k\}, w_\eta, k_0)$;
    $P, S, Q^{\mathrm{T}} = \mathrm{SVD}(Z)$;
    $C_{k_0} = PQ^{\mathrm{T}}$;
  end
  // Then, pick the elements in the factor set $\{D_k\}$ as variables
  for $k_0 = 1$ to K **do**
    $Z = \text{calcu_}Z(\chi, \eta, \{D_k\}, \{C_k\}, w_\eta, k_0)$;
    $P, S, Q^{\mathrm{T}} = \mathrm{SVD}(Z)$;
    $D_{k_0} = PQ^{\mathrm{T}}$;
  end
end
return $\{C_k\}$, $\{D_k\}$;

Algorithm 1: The solution of Problem 2 by using ALS Algorithm.

Problem 3 needs to be solved using Lemma 2. The specific definition and proof process of Lemma 2 are as follows.

Lemma 2. *Given the target function* $\psi = tr((\widetilde{\chi}\prod_k^K \times_k D_k)\prod_k^K \times_k D_k^T \widetilde{\chi}^T)$, $\widetilde{\chi} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_K}$, $D_k \in \mathbb{R}^{J_k \times H_k}$, *each element in* $\{D_k\}$ *satisfies formula (12). Then, the partial derivative of the target function* ψ *to* D_{k_0} *where* $k_0 \in [1, K]$ *is* $\partial\psi/\partial D_{k_0}$, $\partial\psi/\partial D_{k_0} = \varepsilon(\widetilde{\chi}\prod_k^K \times_k D_k)\prod_k^K \times_k D_k^T$, *where* ε *is a constant.*

Proof 3. We use κ to represent $\tau\prod_k^K \times_k D_k$; τ is the identity tensor, $\tau \in \mathbb{R}^{J_1 \times J_2 \times \cdots J_K}$. Then, the target function ψ can be rewritten as $\psi = \mathrm{tr}(\widetilde{\chi}\kappa\kappa^{\mathrm{T}}\widetilde{\chi}^{\mathrm{T}})$. We use ρ to represent $\kappa\kappa^{\mathrm{T}}$. We can obtain that $\partial\phi/\partial\widetilde{\chi} = \partial\mathrm{tr}(\widetilde{\chi}\kappa\kappa^{\mathrm{T}}\widetilde{\chi}^{\mathrm{T}})/\partial\widetilde{\chi} = \partial\mathrm{tr}(\widetilde{\chi}\rho\widetilde{\chi}^{\mathrm{T}})/\partial\widetilde{\chi}$. From the function derivation rule, we can get the following formula: $\partial\mathrm{tr}(\widetilde{\chi}\kappa\kappa^{\mathrm{T}}\widetilde{\chi}^{\mathrm{T}})/\partial\widetilde{\chi} = \partial\mathrm{tr}(\widetilde{\chi}\rho\widetilde{\chi}^{\mathrm{T}})/\partial\widetilde{\chi} = (\rho\widetilde{\chi}^{\mathrm{T}} + \rho^{\mathrm{T}}\widetilde{\chi}^{\mathrm{T}})^{\mathrm{T}}$; thus,

$$\frac{\partial\phi}{\partial\widetilde{\chi}} = 2\widetilde{\chi}\kappa\kappa^{\mathrm{T}} = \varepsilon\left(\widetilde{\chi}\prod_k^K \times_k D_k\right)\prod_k^K \times_k D_k^{\mathrm{T}}, \tag{19}$$

where ε is a constant, $\varepsilon = 2$.

After the aforementioned analysis, Proof 4 gives the solution to Problem 3 and its mathematical proof process while combining Lemma 2 and Proof 3. □

Proof 4. We use υ to represent $\chi\prod_k^K \times_k C_k$ and γ to represent $\tau\prod_k^K \times_k D_k$, where τ is the identity tensor, $\tau \in \mathbb{R}^{J_1 \times J_2 \times \cdots J_K}$. Then, the function F_{RTD} can be rewritten as $F_{\mathrm{RTD}} = \|\upsilon - \widetilde{\chi}\gamma\|_{\mathrm{F}}^2$, that is $F_{\mathrm{RTD}} = \mathrm{tr}((\upsilon - \widetilde{\chi}\gamma)^{\mathrm{T}}(\upsilon - \widetilde{\chi}\gamma))$. Known by the definition of function tr, $\mathrm{tr}(\gamma^{\mathrm{T}}\widetilde{\chi}^{\mathrm{T}}\upsilon) = \mathrm{tr}(\widetilde{\chi}^{\mathrm{T}}\upsilon\gamma^{\mathrm{T}})$. Then, we can get the following equation: $\partial F_{\mathrm{RTD}}/\partial\widetilde{\chi} = (\partial\mathrm{tr}(\upsilon^{\mathrm{T}}\upsilon) - 2\partial\mathrm{tr}(\widetilde{\chi}^{\mathrm{T}}\upsilon\gamma^{\mathrm{T}}) + \partial\mathrm{tr}(\widetilde{\chi}\gamma\gamma^{\mathrm{T}}\widetilde{\chi}^{\mathrm{T}}))/\partial\widetilde{\chi}$. It can be derived from the function derivation rule and Lemma 2 that

$$\frac{\partial F_{\mathrm{RTD}}}{\partial\widetilde{\chi}} = \varepsilon\left(\widetilde{\chi}\gamma\gamma^{\mathrm{T}} - \upsilon\gamma^{\mathrm{T}}\right), \tag{20}$$

where ε is a constant, $\varepsilon = 2$. Let $(\partial F_{\mathrm{RTD}}/\partial\widetilde{\chi})$ be zero. We can get the final solution of Problem 3 by combining formula (12).

$$\widetilde{\chi} = \left(\chi\prod_k^K \times_k C_k\right)\prod_k^K \times_k D_k^{\mathrm{T}}. \tag{21}$$

Algorithm 2 implements RTD by using Algorithm 1. Function TSPA in line 1 represents the implementation of Algorithm 1, and the inputs are χ, η, and w_η. Function F_RTD in line 2 shows the calculation of $\widetilde{\chi}$ using equations (18)–(21). Finally, the core tensor of χ is obtained by using Algorithm 2, which approximates the restricted tensor η on the layer of tensor structure and elements information. □

4.2. rdRNN. This study proposes a new RNN called rdRNN. Unlike traditional RNN, rdRNN sets up a new gate based on the bidirectional RNN. The new gate uses the similarity matrix between samples as a parameter of the deep neural network training model. Compared with the original bidirectional RNN, the classification result of rdRNN is more accurate and stable. For the intelligent judgment of legal cases, the original deep neural network method does not consider the correlation between legal cases. This disregard may lead to bias in the final case classification. For example, the verdict of a legal case is inconsistent with the description

Input: Tensor χ which represents the original legal case data, $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$, the restricted tensor η, $\eta \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_K}$, and its weight w_η.
Output: The core tensor $\widetilde{\chi}$ which is close to the restricted tensor η in the layer of tensor structure and elements value, $\widetilde{\chi} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_K}$.
// Solving the factor matrix sets $\{C_k\}$ and $\{D_k\}$ using Algorithm 1
$\{C_k\}, \{D_k\} = \mathrm{TSPA}(\chi, \eta, w_\eta)$;
$\widetilde{\chi} = \mathrm{F_RTD}(\chi, \{C_k\}, \{D_k\})$;
return $\widetilde{\chi}$;

Algorithm 2: The restricted tensor decomposition method.

of the case. To solve this problem, rdRNN fully considers the judgment results of legal cases that are similar to the case to be judged. rdRNN uses these results as a parameter of the deep neural network training model and realizes an efficient and accurate classification of multiple accusations in legal cases.

The following section shows the training of rdRNN's deep neural network while using Rmsprop as its optimization function:

(i) We use the dataset $\left\{\widetilde{\chi}^{(n)}, L^{(n)}\right\}$ and the restricted tensor η as inputs of rdRNN. $\widetilde{\chi}^{(n)}$ is the core tensor of $\chi^{(n)}$, which represents the legal case. $\widetilde{\chi}^{(n)}$ is obtained by the RTD algorithm with $\chi^{(n)}$ and η as its inputs. $L^{(n)}$ represents the category label of $\chi^{(n)}$ according to the judgment result of legal case.

(ii) In this study, we combine rdRNN with the softmax layer to complete the classification of legal cases. For sample $\chi^{(n)}$, assuming $h^{(n)}$ is the output vector of rdRNN, the softmax layer implements the mapping of $h^{(n)}$ to the legal case category $L^{(n)}$.

(iii) We use cross entropy croen_F as the loss function to update rdRNN. rdRNN uses its forward propagation algorithm and error backpropagation formulas to iterate over the values of parameters in neural networks, such as weight matrices $\{w_d\}$ and bias terms $\{b_d\}$ that are associated with relationship gate, and restricted tensor η, where d is the number of hidden layers.

(iv) We select Adam as the optimization function of rdRNN, and Rmsprop completes the optimization and calculation of parameters $\{w_d\}$, $\{b_d\}$, and η by using $\partial\mathrm{crosen_F}/\partial w_d$, $\partial\mathrm{crosen_F}/\partial b_d$, and $\partial\mathrm{crosen_F}/\partial\eta$.

4.2.1. Calculation of Forward Propagation in rdRNN. In this study, we fully consider the relationship between legal cases and set up a new gate to complete the classification of legal cases, eliminate contingency errors as much as possible, and avoid inconsistencies between the predicted judgment result and the actual case. Relationship control gate r_t is used to control the similar relationship between legal cases. r_t helps the rdRNN deep neural network make an intelligent judgment by using the judgment results of cases that are similar to the case to be judged.

rdRNN can be divided into forward and backward LSTM. These networks do not have obvious differences, except for the opposite propagation direction. In the case of rdRNN forward LSTM propagation network, the formal description of relationship control gate r_t is as follows:

$$r_t = \sigma\left(w_r \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b_r\right), \tag{22}$$

where w_r and b_r are the weight matrix and bias term of the relational control gate r_t, respectively, σ is the activation function, i.e., the sigmoid function, h_{t-1} is the output unit state of the neuron at time $t-1$, and x_t is the input value of the neuron at time t.

In the forward LSTM network, the output of each neuron at time t is calculated by the following formula:

$$\begin{aligned} \begin{bmatrix} r_t \\ f_t \\ i_t \\ o_t \\ \widetilde{c}_t \end{bmatrix} &= \begin{bmatrix} \sigma\left(\mathrm{net}_{r,t}\right) \\ \sigma\left(\mathrm{net}_{f,t}\right) \\ \sigma\left(\mathrm{net}_{i,t}\right) \\ \sigma\left(\mathrm{net}_{o,t}\right) \\ \tanh\left(\mathrm{net}_{\widetilde{c},t}\right) \end{bmatrix} \\ &= \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(\begin{bmatrix} w_{rh}, w_{rx} \\ w_{fh}, w_{fx} \\ w_{ih}, w_{ix} \\ w_{oh}, w_{ox} \\ w_{ch}, w_{cx} \end{bmatrix} \cdot \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + \begin{bmatrix} b_r \\ b_f \\ b_i \\ b_o \\ b_c \end{bmatrix} \right), \end{aligned} \tag{23}$$

where r_t, f_t, i_t, and o_t are the relational control, forget, input, and output gates, respectively; $\widetilde{c}_t$ is the unit status of current inputs; $\mathrm{net}_{r,t}$, $\mathrm{net}_{f,t}$, $\mathrm{net}_{i,t}$, and $\mathrm{net}_{o,t}$ are the weighted inputs of their corresponding gates at time t; $\mathrm{net}_{c,t}$ is the weighted inputs of input state generation function tanh; σ is the activation function, i.e., the sigmoid function,

$\sigma(x) = 1/(1+e^{-x})$; tanh is the hyperbolic tangent function, $\tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$; w_r is the weight matrix of relationship control gate r_t, $w_r = [w_{rh}, w_{rx}]$; w_f, w_i, and w_o are expressed in the same manner as w_r; and b_r, b_f, b_i, and b_o are the bias terms of their corresponding activation functions.

Subsequently, the unit state of the current moment c_t is calculated by f_t, c_{t-1}, i_t, and $\tilde{c}_t$. The calculation formula is expressed as follows:

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t. \tag{24}$$

The final output of the forward LSTM neural network at time t is calculated by o_t, c_t, and r_t and the similar list of x. It is described as follows:

$$h_t = o_t * \tanh(c_t) + r_t * \left(\frac{1}{|\mathrm{List}(x)|} \sum_{x_0 \in \mathrm{List}(x)} \mathrm{Sim}(x_0, x) h_{x_0} \right), \tag{25}$$

where $\mathrm{List}(x)$ is composed of legal cases where the similarity with x is greater than a threshold Max_sim so far. h_{x_0} refers to the output of the forward LSTM neural network that corresponds to legal case x_0. $\mathrm{Sim}(\cdot)$ is a function that calculates the similarity between legal cases. In this study, we set function Sim as the weight of the Euclidean distance and the cosine distance between legal cases.

$$\mathrm{Sim}(x, x_0) = D_{\mathrm{Euclidean}}(x, x_0) + w_d D_{\mathrm{cosine}}(x, x_0), \tag{26}$$

where $D_{\mathrm{Euclidean}}$ and D_{cosine} refer to the Euclidean distance and the cosine distance between the vectors x and x_0, respectively. w_d is the weight matrix.

4.2.2. Calculation of Backpropagation in rdRNN. In this section, we describe in detail the backpropagation algorithm of the rdRNN neural network, including the backpropagation of the error along time and the hidden layer. In rdRNN, forward and backward LSTM neural networks have the same principle in the backpropagation algorithm. Therefore, this section mainly uses forward LSTM as an example.

Given the error term at time t δ_t, $\delta_t = (\partial \mathrm{crosen_F}/\partial h_t)$. Calculation of the backpropagation algorithm of the error term along time is to calculate the value of $\delta_{t-1} = (\partial \mathrm{crosen_F}/\partial h_{t-1})$. The full derivative formula shows that

$$\delta_{t-1} = \frac{\partial \mathrm{crosen_F}}{\partial h_t} \frac{\partial h_t}{\partial h_{t-1}}. \tag{27}$$

Equations (23)–(25) show that r_t, f_t, i_t, o_t, and $\tilde{c}_t$ are all functions of h_{t-1}. Then, we can obtain

$$\begin{cases} \delta_{o,t} = \delta_t \dfrac{\partial h_t}{\partial o_t} \dfrac{\partial o_t}{\partial \mathrm{net}_{o,t}}, \\ \delta_{f,t} = \delta_t \dfrac{\partial h_t}{\partial c_t} \dfrac{\partial c_t}{\partial f_t} \dfrac{\partial f_t}{\partial \mathrm{net}_{f,t}}, \\ \delta_{i,t} = \delta_t \dfrac{\partial h_t}{\partial c_t} \dfrac{\partial c_t}{\partial i_t} \dfrac{\partial i_t}{\partial \mathrm{net}_{i,t}}, \\ \delta_{\tilde{c},t} = \delta_t \dfrac{\partial h_t}{\partial c_t} \dfrac{\partial c_t}{\partial \tilde{c}_t} \dfrac{\partial \tilde{c}_t}{\partial \mathrm{net}_{\tilde{c},t}}, \\ \delta_{r,t} = \delta_t \dfrac{\partial h_t}{\partial r_t} \dfrac{\partial r_t}{\partial \mathrm{net}_{r,t}}, \end{cases} \tag{28}$$

$$\begin{cases} \dfrac{\partial \mathrm{net}_{o,t}}{\partial h_{t-1}} = w_{oh}, \\ \dfrac{\partial \mathrm{net}_{f,t}}{\partial h_{t-1}} = w_{fh}, \\ \dfrac{\partial \mathrm{net}_{i,t}}{\partial h_{t-1}} = w_{ih}, \\ \dfrac{\partial \mathrm{net}_{\tilde{c},t}}{\partial h_{t-1}} = w_{ch}, \\ \dfrac{\partial \mathrm{net}_{r,t}}{\partial h_{t-1}} = w_{rh}. \end{cases}$$

The formula on the left represents the variable declaration, and the formula on the right is calculated from equation (23). According to equations (27) and (28), we can further derive the following formula:

$$\delta_{t-1} = \delta_{o,t} w_{oh} + \delta_{f,t} w_{fh} + \delta_{i,t} w_{ih} + \delta_{\tilde{c},t} w_{ch} + \delta_{r,t} w_{rh}, \tag{29}$$

where $\delta_{o,t} = \delta_t * \tanh(c_t) * o_t * (1 - o_t)$, $\delta_{f,t} = \delta_t * o_t * (1 - \tanh(c_t)^2) * c_{t-1} * f_t * (1 - f_t)$, $\delta_{i,t} = \delta_t * o_t * (1 - \tanh(c_t)^2) * \tilde{c}_t * i_t * (1 - i_t)$, $\delta_{\tilde{c},t} = \delta_t * o_t * (1 - \tanh(c_t)^2) * i_t * (1 - \tilde{c}_t^2)$. According to relationship control gate r_t, equations (23) and (25), we determine that

$$\delta_{r,t} = \delta_t * \left(\frac{1}{|\mathrm{List}(x)|} \sum_{x_0 \in \mathrm{List}(x)} \mathrm{Sim}(x_0, x) h_{x_0} \right) * r_t * (1 - r_t). \tag{30}$$

From equation (29), we can finally figure out the calculation method of the error term in rdRNN is passed from the current moment t to any time k.

Then, we describe in detail the transmission of error between the hidden layers of rdRNN. The error term of the lth hidden layer in rdRNN is assumed to be the partial derivative of the error function crosen_F versus the weighted input net^l. In rdRNN, the input of the lth hidden layer at time t is x_t^l.

$$x_t^l = \text{active}F^{l-1}\left(\text{net}_t^{l-1}\right), \tag{31}$$

where $\text{active}F^{l-1}$ denotes the activation function of the $l-1$th hidden layer in rdRNN and net_t^{l-1} denotes the weighted input of the $l-1$th hidden layer at time t.Given the error term of the lth hidden layer at time t δ_t^l, $\delta_t^l = (\partial\text{crosen_F}/\partial\text{net}_t^l)$, the calculation of error propagation between hidden layers is to figure out the value of δ_t^{l-1}, where

$$\delta_t^{l-1} = \frac{\partial\text{crosen_F}}{\partial\text{net}_t^{l-1}}. \tag{32}$$

According to equations (23), (31), and (32), $\text{net}_{r,t}^l$, $\text{net}_{f,t}^l$, $\text{net}_{i,t}^l$, $\text{net}_{o,t}^l$, and $\text{net}_{\tilde{c},t}^l$ are all functions of x_t^l, and x_t^l is a function of net_t^{l-1}. Therefore, the full derivative formula shows that

$$\begin{aligned}\delta_t^{l-1} = {} & \frac{\partial\text{crosen_F}}{\partial\text{net}_{r,t}^l}\frac{\partial\text{net}_{r,t}^l}{\partial x_t^l}\frac{\partial x_t^l}{\partial\text{net}_t^{l-1}} + \frac{\partial\text{crosen_F}}{\partial\text{net}_{f,t}^l}\frac{\partial\text{net}_{f,t}^l}{\partial x_t^l}\frac{\partial x_t^l}{\partial\text{net}_t^{l-1}} \\ & + \frac{\partial\text{crosen_F}}{\partial\text{net}_{i,t}^l}\frac{\partial\text{net}_{i,t}^l}{\partial x_t^l}\frac{\partial x_t^l}{\partial\text{net}_t^{l-1}} + \frac{\partial\text{crosen_F}}{\partial\text{net}_{o,t}^l}\frac{\partial\text{net}_{o,t}^l}{\partial x_t^l}\frac{\partial x_t^l}{\partial\text{net}_t^{l-1}} \\ & + \frac{\partial\text{crosen_F}}{\partial\text{net}_{\tilde{c},t}^l}\frac{\partial\text{net}_{\tilde{c},t}^l}{\partial x_t^l}\frac{\partial x_t^l}{\partial\text{net}_t^{l-1}}.\end{aligned} \tag{33}$$

The following formula can be obtained by further calculation:

$$\begin{aligned}\delta_t^{l-1} = {} & \left(\delta_{r,t}^l w_{rx} + \delta_{f,t}^l w_{fx} + \delta_{i,t}^l w_{ix} + \delta_{o,t}^l w_{ox} + \delta_{\tilde{c},t}^l w_{cx}\right) \\ & * \text{gra}\left(\text{active}F^{l-1}\left(\text{net}_t^{l-1}\right)\right),\end{aligned} \tag{34}$$

where $\text{gra}(\text{active}F^{l-1}(\text{net}_t^{l-1}))$ is the derivative of function $\text{active}F^{l-1}$ at net_t^{l-1}.

According to equations (27)–(34), we can derive the partial derivative of the loss function crosen_F to the weight matrix set $\{w_{\ d}\}$ and the bias term set $\{b_{\ d}\}$ in rdRNN. Given that $\partial\text{crosen_F}/\partial w_{rh} = \sum_{j=1}^t \partial\text{crosen_F}/\partial w_{rh,j} = \sum_{j=1}^t (\partial\text{crosen_F}/\partial\text{net}_{r,j})(\partial\text{net}_{r,j}/\partial w_{rh,j})$, we obtain

$$\begin{cases}\dfrac{\partial\text{crosen_F}}{\partial w_{rx}} = \displaystyle\sum_{j=1}^t \delta_{r,j} h_{j-1}, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial w_{fx}} = \displaystyle\sum_{j=1}^t \delta_{f,j} h_{j-1}, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial w_{ix}} = \displaystyle\sum_{j=1}^t \delta_{i,j} h_{j-1}, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial w_{ox}} = \displaystyle\sum_{j=1}^t \delta_{o,j} h_{j-1}, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial w_{cx}} = \displaystyle\sum_{j=1}^t \delta_{\tilde{c},j} h_{j-1},\end{cases}$$

$$\begin{cases}\dfrac{\partial\text{crosen_F}}{\partial b_r} = \displaystyle\sum_{j=1}^t \delta_{r,j}, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial b_f} = \displaystyle\sum_{j=1}^t \delta_{f,j}, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial b_i} = \displaystyle\sum_{j=1}^t \delta_{i,j}, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial b_o} = \displaystyle\sum_{j=1}^t \delta_{o,j}, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial b_c} = \displaystyle\sum_{j=1}^t \delta_{\tilde{c},j},\end{cases} \tag{35}$$

$$\begin{cases}\dfrac{\partial\text{crosen_F}}{\partial w_{rx}} = \delta_{r,j} x_t, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial w_{fx}} = \delta_{f,j} x_t, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial w_{ix}} = \delta_{i,j} x_t, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial w_{ox}} = \delta_{o,j} x_t, \\[2ex] \dfrac{\partial\text{crosen_F}}{\partial w_{cx}} = \delta_{\tilde{c},j} x_t.\end{cases}$$

4.2.3. Calculation of the Partial Derivative of Loss Function crosen_F to Restricted Tensor η. This study proposes a new intelligent method for judging legal cases called RnRTD, which combines rdRNN and RTD to complete the classification of legal cases. In the process of training the RnRTD neural network, a new problem is involved: updating of the value of the restricted tensor η so that it can continuously approximate the tensor value that is most beneficial for improving the classification accuracy of the RnRTD algorithm.

The crux to solving this problem is to calculate the partial derivative of the loss function crosen_F to the restricted tensor η, that is, $\partial\text{crosen_F}/\partial\eta$. Directly solving the value of $\partial\text{crosen_F}/\partial\eta$ is difficult. We can use the full derivative rule to obtain that

$$\frac{\partial\text{crosen_F}}{\partial\eta} = \sum_{n}^{N} \frac{\partial\text{crosen_F}}{\partial\widetilde{\chi}^{(n)}} \frac{\partial\widetilde{\chi}^{(n)}}{\partial\eta}. \tag{36}$$

The backward propagation formula of rdRNN shows that $\partial\text{crosen_F}/\partial\widetilde{\chi}^{(n)} = (\partial\text{crosen_F}/\partial\text{net}_1)(\partial\text{net}_1/\ \partial\widetilde{\chi}^{(n)}) = \delta_1 w_0$. According to equations (19)–(21), we determine that $\partial\widetilde{\chi}^{(n)}/\partial\eta = \partial[(\chi^{(n)}\prod_k^K \times_k C_k^{(n)})\prod_{k_1}^K \times_{k_1} U_{k_1}^{(n)}\mathrm{T}]/\partial\eta$. From equations (14)–(17), we know that $C_k^{(n)}$ and $D_k^{(n)}$ are all functions of η. Therefore, the function full derivative rule shows that $\partial\widetilde{\chi}^{(n)}/\partial\eta = \sum_k^K (\partial\widetilde{\chi}^{(n)}/\partial C_k^{(n)})(\partial C_k^{(n)}/\partial\eta) + \sum_k^K (\partial\widetilde{\chi}^{(n)}/\partial D_k^{(n)^{\mathrm{T}}})(\partial D_k^{(n)}\mathrm{T}/\partial\eta)$, that is

$$\frac{\partial\widetilde{\chi}^{(n)}}{\partial\eta} = \sum_{k}^{K} \mathrm{cal}C\left(C_k^{(n)}\right) + \sum_{k}^{K} \mathrm{cal}D\left(D_k^{(n)}\mathrm{T}\right), \tag{37}$$

while

$$\begin{cases} \mathrm{cal}C\left(C_k^{(n)}\right) = \chi^{(n)} \prod_{k_1 \neq k}^{K} \times_{k_1} C_{k_1}^{(n)} \times_k \dfrac{\partial C_k^{(n)}}{\partial\eta} \prod_{k_2}^{K} \times_{k_2} D_{K_2}^{(n)}\mathrm{T}, \\ \mathrm{cal}D\left(D_k^{(n)}\mathrm{T}\right) = \chi^{(n)} \prod_{k_1}^{K} \times_{k_1} C_{k_1}^{(n)} \times_k \dfrac{\partial D_k^{(n)}\mathrm{T}}{\partial\eta} \prod_{k_2 \neq k}^{K} \times_{k_2} D_{K_2}^{(n)}\mathrm{T}. \end{cases} \tag{38}$$

Algorithm 3 provides the optimization process of RnRTD proposed in this study. max_iter in line 2 represents the total number of training sessions of RnRTD. batch_size in line 6 represents the number of samples per batch while training RnRTD. The RTD tensor decomposition method in line 7 corresponds to Algorithm 2. rdRNN_forwardprop in line 8 and rdRNN_backprop in line 9 represent forward propagation and error backpropagation algorithms for rdRNN, respectively, which are the implementations of Sections 4.2.1–4.2.3. In line 11, we use the Adam algorithm to realize parameter optimization of RnRTD neural network.

4.2.4. Loss Function crosen_F and Softmax Layer. In Algorithm 3, we use the softmax function softmax to calculate the probability that $\chi^{(n)}$ belongs to each type of legal case according to judgment results.

Definition 9. Given a set of samples of legal cases and their corresponding outputs of RnRTD $\{(\chi^{(n)}, S^{(n)})\}$, the probability that $\chi^{(n)}$ belongs to each type of legal case is calculated by

$$L_{1q}^{(n)} = \mathrm{softmax}\left(S^{(n)}\right) = \frac{e^{S_q^{(n)}}}{\sum_r^Q e^{S_r^{(n)}}}, \tag{39}$$

where $L_{1q}^{(n)}$ represents the qth element of $L_1^{(n)}$.

In this study, cross entropy is used as the loss function crosen_F to calculate the error of RnRTD. We define crosen_F as follows:

Definition 10. A set of samples of legal cases and their corresponding legal case types $\{(\chi^{(n)}, L^{(n)})\}$ is given. The predicted legal case category of $\chi^{(n)}$ is $L_1^{(n)}$, which is calculated by RnRTD, and then

$$\text{crosen_F}\left(\left\{L^{(n)}\right\}, \left\{L_1^{(n)}\right\}\right) = \sum_{n}^{N} \sum_{q}^{Q} L_q^{(n)} \log L_{1q}^{(n)}, \tag{40}$$

where N represents the number of samples of legal cases and q represents the dimension of $L^{(n)}$ and $L_1^{(n)}$, that is, the number of types of legal cases.

5. Results

5.1. Description of Experimental Data. We use nearly 1.8 million historical legal cases obtained from a Chinese refereeing study network. These legal cases involve more than 200 types of accusations, including theft, intentional assault, smuggling, fraud, and deliberate destruction of public property. Approximately 400,000 cases involve theft, and about 200,000 cases involve intentional assault. The number of accusations involved in each case ranges from 1 to 23.

Figure 8(a) shows the distribution of various accusations in the legal case data used in this study. The abscissa indicates the accusation index. For example, index 1 corresponds to bribery, and index 2 corresponds to rape. The ordinate indicates the proportion of cases involving the accusation that occupy the overall cases. Figure 8(a) shows that the number of cases involving theft is the highest in the database used in this article.

Figure 8(b) shows the distribution of the number of accusations involved in each case. The abscissa indicates the number of accusations involved in the case, and the ordinate indicates the proportion of cases in the corresponding number of accusations. Figure 8(b) shows that the highest number of accusations involves three cases.

5.2. Baseline Approaches. Given that multiaccusation judgment based on deep neural network and tensor decomposition is rarely studied, according to the limited tensor decomposition method RTD and the relation-based recurrent neural network rdRNN, we use the following method for a comparison with RnRTD proposed in this study:

Input: $\{(\chi^{(n)}, L^{(n)})\}$, where $\chi^{(n)}$ represents the legal cases and $L^{(n)}$ represents the category of legal case corresponding to $\chi^{(n)}$ according to judgment results. The size of η, $\{w_r\}$, $\{w_f\}$, $\{w_i\}$, $\{w_o\}$, $\{w_c\}$, $\{b_r\}$, $\{b_f\}$, $\{b_i\}$, $\{b_o\}$ and $\{b_c\}$, where $w_r = [w_{rh}, w_{rx}]$, $w_f = [w_{fh}, w_{fx}]$, $w_i = [w_{ih}, w_{ix}]$, $w_o = [w_{oh}, w_{ox}]$ and $w_c = [w_{ch}, w_{cx}]$
Output: The optimal restricted tensor η, parameters of rdRNN $\{w_r\}$, $\{w_f\}$, $\{w_i\}$, $\{w_o\}$, $\{w_c\}$ and $\{b_r\}$, $\{b_f\}$, $\{b_i\}$, $\{b_o\}$, $\{b_c\}$.
Initialize the restricted tensor η, parameters of rdRNN $\{w_r\}$, $\{w_f\}$, $\{w_i\}$, $\{w_o\}$, $\{w_c\}$ and $\{b_r\}$, $\{b_f\}$, $\{b_i\}$, $\{b_o\}$, $\{b_c\}$
for iter = 1 to max_iter **do**
  Set batch_indices to zero;
  while batch_indices ≤ dataset_size **do**
    Set $\{\partial \text{crosen_F}/\partial\eta\}$, $\{\partial \text{crosen_F}/\partial w_r\}$, $\{\partial \text{crosen_F}/\partial w_f\}$, $\{\partial \text{crosen_F}/\partial w_i\}$, $\{\partial \text{crosen_F}/\partial w_o\}$, $\{\partial \text{crosen_F}/\partial w_c\}$ and $\{\partial \text{crosen_F}/\partial b_r\}$, $\{\partial \text{crosen_F}/\partial b_f\}$, $\{\partial \text{crosen_F}/\partial b_i\}$, $\{\partial \text{crosen_F}/\partial b_o\}$, $\{\partial \text{crosen_F}/\partial b_c\}$ to zero;
    for i = batch_indices to (batch_indices + batch_size) **do**
      $\tilde{\chi}^{(i)} = \text{RTD}(\chi^{(i)}, \eta)$;
      $L_1^{(i)} = \text{rdRNN_forwardprop}(\tilde{\chi}^{(i)}, \{w_r\}, \{w_f\}, \{w_i\}, \{w_o\}, \{w_c\}, \{b_r\}, \{b_f\}, \{b_i\}, \{b_o\}, \{b_c\})$;
      $\{\partial \text{crosen_F}/\partial\eta\}$, $\{\partial \text{crosen_F}/\partial w_r\}$, $\{\partial \text{crosen_F}/\partial w_f\}$, $\{\partial \text{crosen_F}/\partial w_i\}$, $\{\partial \text{crosen_F}/\partial w_o\}$, $\{\partial \text{crosen_F}/\partial w_c\}$, $\{\partial \text{crosen_F}/\partial b_r\}$, $\{\partial \text{crosen_F}/\partial b_f\}$, $\{\partial \text{crosen_F}/\partial b_i\}$, $\{\partial \text{crosen_F}/\partial b_o\}$, $\{\partial \text{crosen_F}/\partial b_c\}$+ = $\text{rdRNN_backprop}(\tilde{\chi}^{(i)}, L_1^{(i)}, L^{(i)}, \eta, \{w_r\}, \{w_f\}, \{w_i\}, \{w_o\}, \{w_c\}, \{b_r\}, \{b_f\}, \{b_i\}, \{b_o\}, \{b_c\})$;
    end
  η, $\{w_r\}$, $\{w_f\}$, $\{w_i\}$, $\{w_o\}$, $\{w_c\}$, $\{b_r\}$, $\{b_f\}$, $\{b_i\}$, $\{b_o\}$, $\{b_c\}$ = $\text{Adam}(\eta, \{w_r\}, \{w_f\}, \{w_i\}, \{w_o\}, \{w_c\}, \{b_r\}, \{b_f\}, \{b_i\}, \{b_o\}, \{b_c\}, \{\partial \text{crosen_F}/\partial\eta\}, \{\partial \text{crosen_F}/\partial w_r\}, \{\partial \text{crosen_F}/\partial w_f\}, \{\partial \text{crosen_F}/\partial w_i\}, \{\partial \text{crosen_F}/\partial w_o\}, \{\partial \text{crosen_F}/\partial w_c\}, \{\partial \text{crosen_F}/\partial b_r\}, \{\partial \text{crosen_F}/\partial b_f\}, \{\partial \text{crosen_F}/\partial b_i\}, \{\partial \text{crosen_F}/\partial b_o\}, \{\partial \text{crosen_F}/\partial b_c\})$;
  end
end
return η, $\{w_r\}$, $\{w_f\}$, $\{w_i\}$, $\{w_o\}$, $\{w_c\}$ and $\{b_r\}$, $\{b_f\}$, $\{b_i\}$, $\{b_o\}$, $\{b_c\}$;

Algorithm 3: The framework of RnRTD algorithm.

(i) This study uses a series of deep neural networks based on convolutional or cyclic neural networks, including TextCNN, TextCNN attention, TextRNN, TextRNN attention, LSTM, and Bi-LSTM, as comparison methods for RnRTD.

(ii) This study uses deep neural networks with only the RTD tensor decomposition layer as comparison methods for RnRTD. Through experiments, we can derive the contribution of the RTD tensor decomposition layer to all deep neural networks.

(iii) This study uses neural networks with only the rdRNN method as comparison methods for RnRTD. Through these comparison experiments, we can derive the contribution of the relationship-based rdRNN strategy to all RNNs.

5.3. Data Preprocessing. In this study, the data preprocessing operation can be divided into two parts, namely, the modular representation of legal cases and the construction of the original tensor. Our legal case data preprocessing process can be described as follows:

(i) We organize each case in the legal case database into our preestablished case model, which divides the original case file into the defendant's statement, the plaintiff's statement, the content of the public prosecutions allegations, and the court's judgment.

(ii) We filter and clean the contents of each module in the legal cases, extract the words that are meaningful to our multiaccusation judgment method, and filter out redundant words, stop words, noisy words, and modal particles.

(iii) We train the word-to-vector model to obtain the word vector of the aforementioned vocabulary. Then, we obtain a matrix representation of each case module and derive the tensor representation of the entire legal case.

For Step (1), each case module may be spread across different paragraphs, and cases in different regions have different case descriptions. We extract and integrate them separately to arrive at a modular representation of the cases based on the description rules of case documents in each region. For Step (2), the extraction and filtering of the vocabulary in legal cases often requires professional legal background knowledge; otherwise, error filtering will occur. We filter words in legal case modules by using the legal professional vocabulary and the stop word list. For Step (3), word vectors are the basis for the accuracy of the entire deep neural network method. We use a number of Chinese corpus, such as corpus on Baidu Encyclopedia, Zhihu Questions and Answers, Sohu News, and Sina Weibo, to train the word-to-vector model.

The tensor representation of legal cases and the subsequent deep neural network classification method require each case to have the same number of words, and the number of words in 95% of the cases is below 300. Therefore, we perform a padding operation for cases where the number of words is less than 300. For cases with a vocabulary number greater than 300, we use the TF-IDF weight of the vocabulary to tailor the case vocabulary.

Figure 8: Statistics on allegations in legal cases.

5.4. Experimental Hyperparameter Setting. This section describes the hyperparameter settings involved in our proposed method. These settings include the restricted tensor η and weight matrix W_η in RTD and the size of the similar case list in the rdRNN method (i.e., the size of list $|*|$ in equation (25)).

The setting of restricted tensor η directly affects the convergence speed and accuracy of RnRTD. Our experiments show that a large rank of restricted tensor η corresponds to a high accuracy of the subsequent deep neural network algorithms. Conversely, a strong linear relationship between column or row vectors in η results in a low accuracy of the subsequent classification algorithms.

In this study, weight matrix W_η is used to scale the elements of the last mode in the original tensors. W_η adjusts the weights of certain words in the legal cases. For different accusations, the same vocabulary may have different weights in different types of cases. For example, derailment occupies a large and small weight in cases that involve bigamy and smuggling, respectively.

The size of the similar case list in rdRNN is an important indicator that determines the impact of the relationship between cases on the final classification result. If the length of the similar case list is set too long, then it is equivalent to strengthening the weak similarity between cases and weakening the strong similarity between cases. Furthermore, if the length of the similar case list is set too short, then it is equivalent to weakening the weak similarity between cases and strengthening the strong similarity between cases. After many experiments, we set the case similar list length to 50.

5.5. Experimental Results and Analysis. This section shows the superiority of the proposed RnRTD method for multiple accusations in legal cases relative to the baseline listed in Section 5.2 and provides the corresponding analysis.

Figure 9 shows a series of experimental results based on Bi-LSTM. The abscissa indicates the number of batch iterations, and the ordinate indicates the accuracy of the multiaccusation judgment methods in legal cases. In contrast with the original Bi-LSTM method and Bi-LSTM with only the rdRNN layer, Bi-LSTM with only RTD achieves stable accuracy at the highest speed as the number of batches increases.

The characteristics of RTD are important factors in the aforementioned phenomenon. On the basis of the restricted tensor η, RTD extracts the tensor elements and structure information that are most relevant to the multiaccusation judgment of legal cases from the original tensor. The weight of the vocabulary unrelated to a particular accusation is considerably weakened, and the weight of the vocabulary associated with a particular accusation is strengthened. The tensor dimension is greatly reduced, and the influence of irrelevant vocabulary on the classification algorithm is reduced. Subsequent neural network algorithms continuously iterate and optimize the restricted tensor and continuously adjust and correct the element values of the core tensor. RTD optimizes the original deep neural network algorithm from the lexical level.

In Figure 9, as the number of batches increases, the accuracy of Bi-LSTM with only the rdRNN layer becomes ultimately higher than that of Bi-LSTM with only the RTD layer. The reason is that rdRNN fully considers the similarity between different cases and has better discrimination for similar cases expressed by different language description methods. By setting the appropriate similar case list size, rdRNN fully considers cases that are similar to the current case and weighs their corresponding output states according to their similarity. rdRNN corrects and optimizes the original deep neural network from the case level.

Figure 10 shows the experimental results of the TextRNN-based RnRTD method. Similar to what is shown in Figure 9, TextRNN with only the RTD layer has the highest convergence speed as the number of batches increases compared with the original TextRNN and TextRNN with only the rdRNN layer.

The accuracy of the deep neural network method with only the rdRNN layer is not always higher than that with

Figure 9: Experimental results of RnRTD based on Bi-LSTM.

Figure 10: Experimental results of RnRTD based on TextRNN.

only the RTD layer. Although the rdRNN layer implements the correction and optimization of subsequent classification algorithms at the case level through the setting of similar case lists, the RTD layer also optimizes classification algorithms at the vocabulary level by setting the restricted tensor. Both methods achieve the final accuracy optimization but have different effects for various contexts. RnRTD combines the advantages of RTD and rdRNN to achieve rapid convergence and high classification accuracy.

Table 2 provides an experimental comparison of RnRTD methods based on multiple deep neural networks. RnRTD remarkably improves the classification accuracy of original neural networks for the classification of multiple accusations in legal cases. RTD and rdRNN layers also have considerable optimization effects on the original neural networks. RTD is applicable to all deep neural networks and can extract the main information carried by the data at the input layer to realize dimension reduction. rdRNN is an optimization

Table 2: Performance of RnRTD methods based on multiple deep neural networks.

Optimization method	Basic algorithm						
	TextCNN	TextRNN	TextCNN attention	TextRNN attention	LSTM	Bi-LSTM	GRU
Original method	0.70	0.73	0.74	0.76	0.75	0.77	**0.79**
Only with RTD	0.73	0.77	0.79	0.82	0.80	0.81	**0.83**
Only with rdRNN	0.74	0.80	0.82	0.83	0.83	**0.84**	0.82
With RnRTD	**0.80**	**0.85**	**0.85**	**0.90**	**0.91**	**0.93**	**0.91**

strategy that is suitable for RNNs. It fully considers the similarity between cases within a certain period and optimizes algorithms at the case level. For algorithms based on convolutional neural networks, we remove the relational control gate in rdRNN while retaining the similar case list. Then, the optimization of these algorithms is realized by the rdRNN layer.

The convolutional neural network is less effective than RNN because legal case data are time series data. In addition, the attention layer only changes the encoding of the input and does not change the structure of neural networks. For the problem of judgment for multiple accusations in legal cases, the attention layer is still difficult to compensate for due to the lack of timing information of TextCNN and the gradient disappearance and gradient explosion of the TextRNN algorithm. From the perspective of the rdRNN layer, GRU has fewer adjustable parameters than LSTM and Bi-LSTM, and optimization on the restricted tensor is relatively limited. Therefore, the Bi-LSTM neural network with RnRTD performs better than other neural network algorithms.

6. Discussion

In this study, we propose a new method for multiaccusation judgment in legal cases called RnRTD. RnRTD is a multilabel classification method based on tensor decomposition and RNNs. RnRTD consists of the tensor decomposition method with constraints and relation-driven RNN.

We propose a tensor decomposition method with constraints, namely, RTD. We use this method to extract the tensor structure and element information that are most favorable to the subsequent classification algorithm from the original tensor that represents the legal case. RTD continuously corrects and optimizes the values of elements in the core tensor through the weight matrix and restricted tensor; hence, it continues to improve the classification accuracy of the neural network. RTD optimizes neural network classification algorithms at the lexical level. We also propose a relation-driven RNN strategy called rdRNN. Unlike traditional recurrent and LSTM neural networks, rdRNN sets up a new gating switch, that is, the similarity list window. It controls the impact of cases similar to the current case on the output status of the current case. rdRNN optimizes neural network classification algorithms at the case level.

According to our experimental results, the RTD layer and the relation-driven cyclic neural network rdRNN have remarkable optimization effects on various deep neural network algorithms. However, no obvious relationship exists between the two. RTD and rdRNN have their own advantages in different contexts. In Figures 9 and 10, the accuracy of rdRNN is higher than that of RTD. The accuracy of rdRNN is not always higher than that of RTD. RTD achieves stable accuracy the fastest as the number of batches increases in both figures. The reason is the principal component extraction and dimensionality reduction of RTD itself.

RTD is suitable for almost all deep neural networks. It performs principal component extraction and dimensionality reduction on the original data at the input layer. It is similar to traditional principal component analysis methods, such as PCA [20] and SVD [21]. Several decomposition methods [22], such as Tucker and CP [23], are currently used for high-dimensional data. These methods extract the main elements and structural information of the matrix or tensor at the logical level according to the linear relationship of the elements in the matrix or tensor. However, the resulting new matrix or tensor structure is often unexplained. According to the traditional matrix or tensor decomposition method, supervising the completion of the principal component extraction work is difficult.

The proposed restricted tensor provides interpretability for the tensor decomposition operation. Under the influence of the restricted condition tensor, RTD retains the information in the original tensor that is beneficial to the subsequent neural network and removes useless information. For the overall classification algorithm, RTD reduces the weight of weak correlation information and improves the influence of strong correlation information on the classification model. In addition, the subsequent deep neural network algorithm will continuously update and optimize the constraints in RTD to guide the core tensor to retain information that is conducive for the classification model. RTD optimizes the classification model at the vocabulary level by combining the weight matrix with the restricted tensor.

rdRNN fully considers the similarities between cases and uses it as a factor that influences the output status of the current case. rdRNN optimizes the entire classification model at the case level. Generally, different regions may use different legal case description vocabularies, and rdRNN sets the output status of similar historical cases within a certain period as the reference value of the current case output state by setting a similar case window. Moreover, it sets the weight according to the similarity. RnRTD combines RTD and rdRNN to optimize the classification results from the perspective of case and vocabulary. When we use rdRNN to optimize algorithms based on the convolutional neural network, we remove the relationship control gate, retain only the similar case list in rdRNN, and realize the optimization operation of the rdRNN layer on the neural network.

7. Conclusion

In this study, we propose a new method for judging multiple crimes in legal cases, namely RnRTD. RnRTD consists of RTD and rdRNN. RTD is a tensor decomposition method with constraints. RTD decomposes the original tensor that represents a legal case into a core tensor under the guidance of restricted tensor. The resulting core tensor represents the main tensor structure and element information that is most favorable for improving the accuracy of subsequent classification algorithm. We propose the rdRNN algorithm and train it using obtained core tensors. rdRNN guides the tensor decomposition process in RTD by continuously optimizing the restricted tensor and finally makes RTD develop in the direction that is most beneficial to improve the classification accuracy of rdRNN.

Nevertheless, this study has several problems. For example, even with the RTD tensor decomposition layer, algorithms based on RNNs usually run very slowly. In our future work, we will attempt to reduce the computational complexity of the algorithm and increase its speed.

Data Availability

The legal cases data after processing used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 6/12 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

In the research and writing work of this thesis, we would like to thank the teachers and students of the lab team (Network and Information Security Research Center, Harbin Institute of Technology) for their selfless help. This work was supported by the National Key Research and Development Program of China (2018YFC0830602 and 2016QY03D0501).

References

[1] U. Hahn and N. Chater, *Understanding Similarity: A Joint Project for Psychology, Case-Based Reasoning, and Law*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1998.

[2] V. Aleven, "Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment," *Artificial Intelligence*, vol. 150, no. 1-2, pp. 183–237, 2003.

[3] T. Bench-Capon and G. Sartor, "A model of legal reasoning with cases incorporating theories and values," *Artificial Intelligence*, vol. 150, no. 1-2, pp. 97–143, 2003.

[4] A. S. A. Garcez, L. C. Lamb and D. M. Gabbay, *Neural-Symbolic Cognitive Reasoning*, Springer, Berlin, Germany, 2009.

[5] T. R. Besold, A. S. A. Garcez, S. Bader et al., "Neural-symbolic learning and reasoning: a survey and interpretation," 2017, https://arxiv.org/abs/1711.03902.

[6] A. A. Garcez, T. Besold, L. de Raedt et al., "Neural-symbolic learning and reasoning: contributions and challenges," in *Proceedings of the 2015 AAAI Spring Symposium Series*, Stanford, CA, USA, March 2015.

[7] A. S. A. Garcez, D. M. Gabbay, and L. C. Lamb, "A neural cognitive model of argumentation with application to legal inference and decision making," *Journal of Applied Logic*, vol. 12, no. 2, pp. 109–127, 2014.

[8] M. G. Venkateshmurthy, T. V. Geetha, and R. K. Subramanian, "Extraction and representation of facts from legal briefs," *Journal of Intelligent and Robotic Systems*, vol. 3, no. 2, pp. 167–182, 1990.

[9] T. D. Breaux, A. I. Antn, and E. H. Spafford, "A distributed requirements management framework for legal compliance and accountability," *Computers and Security*, vol. 28, no. 1-2, pp. 8–17, 2009.

[10] D. A. Waterman, J. Paul, and M. Peterson, "Expert systems for legal decision making," *Expert Systems*, vol. 3, no. 4, pp. 212–226, 1986.

[11] R. Bartolini, A. Lenci, S. Montemagni, V. Pirrelli, and C. Soria, *Automatic Classification and Analysis of Provisions in Italian Legal Texts: A Case Study*, in *Proceedings of the On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops*, Agia Napa, Cyprus, October 2004.

[12] S. Joshi, P. M. Deshpande, and T. Hampp, "Improving the efficiency of legal e-discovery services using text mining techniques," in *Proceedings of the Srii Global Conference*, San Jose, CA, USA, April 2011.

[13] O.-M. Sulea, M. Zampieri, S. Malmasi, M. Vela, L. P. Dinu, and J. van Genabith, "Exploring the use of text classification in the legal domain," 2017, https://arxiv.org/abs/1710.09306.

[14] S. Bruninghaus and K. D. Ashley, "Progress in textual case-based reasoning: predicting the outcome of legal cases from text," in *Proceedings of the National Conference on Artificial Intelligence*, Boston, MA, USA, 2006.

[15] M. Welling and M. Weber, "Positive tensor factorization," *Pattern Recognition Letters*, vol. 22, no. 12, pp. 1255–1261, 2001.

[16] R. J. Williams and D. Zipser, *A Learning Algorithm for Continually Running Fully Recurrent Neural Networks*, MIT Press, Cambridge, MA, USA, 1989.

[17] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 2002.

[18] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with lstm," in *Proceedings of the International Conference on Artificial Neural Networks ICANN*, pp. 850–855, Madrid, Spain, August 2002.

[19] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[20] E. Oja, "The nonlinear pca learning rule in independent component analysis," *Neurocomputing*, vol. 17, no. 1, pp. 25–45, 1997.

[21] M. Vozalis and K. Margaritis, "Using svd and demographic data for the enhancement of generalized collaborative filtering," *Information Sciences*, vol. 177, no. 15, pp. 3017–3037, 2007.

[22] A. Cichocki, R. Zdunek, and S. I Amari, "Nonnegative matrix and tensor factorization," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 54–65, 2009.

[23] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 1604392, 10 pages
https://doi.org/10.1155/2019/1604392

Research Article

Adaptive Learning Emotion Identification Method of Short Texts for Online Medical Knowledge Sharing Community

Dan Gan,[1] **Jiang Shen**,[1] **and Man Xu**[2]

[1]*College of Management and Economics, Tianjin University, Tianjin 300072, China*
[2]*Business School, Nankai University, Tianjin 300071, China*

Correspondence should be addressed to Man Xu; td_xuman@nankai.edu.cn

Received 9 January 2019; Revised 10 April 2019; Accepted 8 May 2019; Published 25 June 2019

Guest Editor: Zoran Stamenkovic

Copyright © 2019 Dan Gan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The medical knowledge sharing community provides users with an open platform for accessing medical resources and sharing medical knowledge, treatment experience, and emotions. Compared with the recipients of general commodities, the recipients in the medical knowledge sharing community pay more attention to the intensity or overall evaluation of emotional vocabularies in the comments, such as treatment effects, prices, service attitudes, and other aspects. Therefore, the overall evaluation is not a key factor in medical service comments, but the semantics of the emotional polarity is the key to affect recipients of the medical information. In this paper, we propose an adaptive learning emotion identification method (ALEIM) based on mutual information feature weight, which captures the correlation and redundancy of features. In order to evaluate the proposed method's effectiveness, we use four basic corpus libraries crawled from the Haodf's online platform and employ Taiwan University NTUSD Simplified Chinese Emotion Dictionary for emotion classification. The experimental results show that our proposed ALEIM method has a better performance for the identification of the low-frequency words' redundant features in comments of the online medical knowledge sharing community.

1. Introduction

More and more comments, opinions, suggestions, ratings, and feedback are produced on social networks with the rapid development of the Internet [1]. While those on social networks are meant to be useful, this part of the contents requires adopting text mining and emotion analysis techniques. Until now, emotional analysis and evaluation still face several challenges [2], which are shown in Table 1. These challenges become obstacles to accurately analyze emotional polarity.

In recent years, more and more research has been done on emotion analysis. Among them, unstructured natural language texts have received the widest attention of scholars [9]. Emotion analysis is the inference of users' opinions, positions, and attitudes through written or spoken contents [10]. Solving emotion analysis tasks typically uses dictionary-based and learning-based approach [11, 12]. The dictionary-based approach analyzes the relevance of each word to a particular emotion by using the predefined dictionary [13]. Learning-based methods typically use labeled samples to train the specific-purpose models under supervision [14].

Emotional analysis is increasingly used to analyze human emotions, but the fatal shortcoming of current emotion analysis methods is lack of aspect level granularity improvement, and also they are rarely applied to online knowledge communities, especially medical knowledge communities, so it is necessary to find an emotional classification method for medical knowledge communities. In light of these considerations, we propose an adaptive learning emotion identification method (ALEIM) based on mutual information feature weight, which captures the correlation and redundancy of features. Its effectiveness is verified on the datasets crawled from the Haodf's online platform, in which the eigenvalues corresponding to the feature nouns are assigned according to the emotional dictionary NTUSD compiled by Taiwan

Table 1: Summary of typical previous studies for the emotion analysis challenges.

Author	Year	Domain oriented	Challenge type	Review structure
Jia et al. [3]	2009	Health/medical domain	Theoretical	Semi-structured
Hogenboom et al. [4]	2011	Movie reviews	Theoretical	Unstructured
Alexandra and Ralf [5]	2009	Online news reviews	Theoretical	Semistructured/unstructured
Mukherjee and Bhattacharyya [6]	2012	Products	Technical	Semistructured
Chetan and Atul [7]	2014	Tweets	Technical	Unstructured
Doaa and Osama [8]	2015	Scientific papers	Theoretical + technical	Structured

University. Finally, the experimental results show that our proposed ALEIM method achieves a better performance.

The remainder of this paper is organized as follows. Section 2 reviews the related work of our study. Section 3 presents our proposed ALEIM method, which contains problem description and assumptions, feature selection based on mutual information, and emotional polarity selection based on mutual information weight. Section 4 presents the datasets, evaluation measures, experimental performance, and the discussion. Finally, Section 5 presents the conclusions.

2. Related Work

2.1. Feature Extraction. Natural language processing and text analysis techniques are used to extract emotion features in emotion comments [9]. However, the feature selection method based on mutual information is developed to obtain the true feature, which is an information entropy estimation method independent of classifiers and datasets and superior to other feature extraction methods [15, 16]. A redundant algorithm for constructing the mutual information feature subset was proposed and used to improve the emotion classification accuracy [17]. The maximal relevance and minimal redundancy (mRMR) algorithm was proposed on the basis of the principle of mutual information, which was compared with the SVM classification [18, 19] and the recommended three ratio classification methods; the proposed accuracy is superior to traditional method, and recognition speed is faster than the intelligent method [20].

2.2. Emotion Analysis. Emotion analysis has been widely used in many fields [21, 22], such as consumer management, precision marketing, social network, etc. Unsupervised learning algorithm and the foremost supervised learning algorithm were used to classify emotion polarity of comments [23]. Moreover, emotion analysis is divided into many levels: document level [24], sentence level [25], word/term level, or aspect level [26].

Until now, the emotion classification methods can be roughly divided into three fields: machine learning methods, emotion dictionary-based methods [27], and deep learning emotion classification approaches [28]. Some common classifiers for machine learning methods are decision trees [29], Bayes [30], and support vector machines [31]. Emotion dictionary-based approach is to achieve classification by using the different granularity of emotion words polarity. The common emotion lexicons include the following: SentiWordNet [32], General Inquirer [33], SenticNet [34], Opinion Lexicon, HowNet Emotional Dictionary, Subjective Lexicon, DUTIR emotional vocabulary ontology library, and NTUSD [35]. However, it is very difficult to construct a complete emotion dictionary, which may have polarity of all emotion words. Therefore, it is necessary to obtain the polarity of emotional words by context. Deep learning emotion classification approaches are usually used to achieve emotion classification at aspect level. In terms of natural language processing, deep learning has far superior performance to machine learning [18], and it has been proved in the fields of text recognition [36] and semantic mining [37]. More recently, deep learning, especially convolutional neural network is widely used to improve the emotion analysis accuracy [38–40].

3. Semisupervised Emotion Classification Method

3.1. Problem Description and Assumptions. Let the basic corpus denote $\Theta = (U, A, V, f)$, the domain U indicate the source review set exists N comments, $U = \{u_i \mid i = 1, 2, \ldots, N\}$, u_i be the nth comment, and N be the total number of comments. The feature noun set of comments denotes $A = \{a_j \mid j = 1, 2, \ldots, J\}$, a_j is the jth comment feature, and J is the total number of feature noun. Among them, the overall characteristics of the review (patient satisfaction, efficacy) are also known as the identification category, which is recorded as c. The range of eigenvalues is V; it forms an information function with U and A: $f : U \times A \longrightarrow V$. Let $V = \{V^{(u_i)} \mid u_i \in U\}$; then, $V^{(u_i)}$ is the eigenvalue vector of the comment u_i and $V^{(u_i)} = \{V_{ij} \mid j = 1, 2, \ldots, k,\ j \le k\}$, and k is the number of eigenvalues of the feature noun a_i. V_{ij} is the jth eigenvalue of the u_i comment (the eigenvalue is related to the adjective corresponding to the noun). The new comment is recorded as T; the comment feature matrix can be defined as $V = [V_{\mathrm{U}} \ldots V_{\mathrm{T}}]^T$. The data in the comments are multi-isomerized, so it is necessary to normalize the eigenvalues.

$$v_j^{(u_i)} = \frac{V_{ij} - \min_i V_{ij}}{\max_i V_{ij} - \min_i V_{ij}}. \quad (1)$$

We number all the adjectives contained in each feature and substitute the number as eigenvalues into the matrix for subsequent calculations.

Let $M(c; f_\lambda)$ be the jth eigenvalues of the comment u_i; then, V is converted to V^*.

$$V^* = \left[\frac{V_{\mathrm{U}}}{V_{\mathrm{T}}}\right] = \begin{bmatrix} v_1^{(u_1)} & v_2^{(u_1)} & \cdots & v_k^{(u_1)} \\ v_1^{(u_2)} & v_2^{(u_2)} & \cdots & v_k^{(u_2)} \\ \vdots & \vdots & \vdots & \vdots \\ v_1^{(u_N)} & v_2^{(u_N)} & \cdots & v_k^{(u_N)} \\ v_1^{(t)} & v_2^{(t)} & \cdots & v_k^{(t)} \end{bmatrix}. \quad (2)$$

Due to the uncertainty of the adjective language selection in the commentary library, the probability is used to describe its distribution characteristics. P_{il} is the probability of feature a_i values $v_l^{(u_i)}$; after the commentator's emotional polarity is determined, the word is uncertain, and use the probability to eliminate the influences of the commenters' decision.

The uncertainty of the emotional polarity of comments is concentrated in the feature redundancy of the comment set. Mutual information can effectively measure the redundancy between variables in a feature set. It is thus possible to find a set of input features that has a large mutual information value with the identification category and low redundancy between other features. The feature Relation-Redundancy Coefficient ($\mathrm{R}^2\mathrm{C}$) is used for discrimination considering both the range of feature values and the distribution of values.

In the feature selection process, the joint action of multiple candidate features on the category c, due to the redundancy. In this paper, the redundancy between a_{λ} and selected feature S and the redundancy between all features in S are collectively referred to as the redundancy of the feature, denoted by $M(c; a_{\lambda}; S)$. The eigenvalue number of feature a_i is k; then, its information entropy can be denoted as

$$E(a_i) = \sum_{l=1}^{k} P_{\mathrm{il}} \log_2 \frac{1}{P_{\mathrm{il}}}. \quad (3)$$

If $a_i \in A$, $a_j \in A$, and $a_j \neq a_i$, according to the joint distribution rate, the conditional entropy can be denoted as

$$E(a_i \mid a_j) = -\sum_{a_i, a_j \in A} \sum_{a_j \neq a_i} P(a_i, a_j) \log_2 P(a_i \mid a_j). \quad (4)$$

Definition 1. Comment space mutual information.

In Θ, the mutual information between $a_i \in A$ and $a_j \in A$ in feature set A can be expressed as

$$M(a_i; a_j) = \sum_{a_i, a_j \in A} \sum_{a_j \neq a_i} P(a_i, a_j) \log_2 \frac{P(a_i \mid a_j)}{P(a_i) P(a_j)}. \quad (5)$$

The larger $M(a_i; a_j)$ is, the closer the relationship between the feature random variables a_i and a_j is; when $M(a_i; a_j)$ approaches zero, the two are independent of each other. The relationship between mutual information and information entropy can be expressed as

$$M(a_i; a_j) = E(a_i) - E(a_i \mid a_j). \quad (6)$$

3.2. Feature Selection Based on Mutual Information

Definition 2. Let $\hbar_{\mathrm{s}}$ be the ratio of the mutual information $M(c; a_{\mathrm{s}})$ between selected feature a_{s} and identification category c to the information entropy $E(a_{\mathrm{s}})$ of the feature a_{s}; then, $\hbar_{\mathrm{s}} = M(c; a_{\mathrm{s}})/E(a_{\mathrm{s}})$, $0 \le \hbar_{\mathrm{s}} \le 1$.

$\hbar_{\mathrm{s}}$ meets the following characteristics:

(i) When the range of feature values is the same, the more uniform the value is, the less important it is

(ii) When the feature values are evenly distributed, the larger the value range is, the less important it is

(iii) Then, the mutual information formula of feature redundancy in the MIFS-U method is expressed as

$$M(c; a_{\lambda}; S) = \beta \sum_{a_{\mathrm{s}} \in S} \{\hbar_{\mathrm{s}} \cdot M(a_{\mathrm{s}}; a_{\lambda})\}. \quad (7)$$

The ratio of mutual information between maximum correlation and minimum redundancy denotes the ratio of feature correlation and redundancy. When $\sum_{f_{\mathrm{s}} \in S} \{\hbar_{\mathrm{s}} \cdot M(a_{\mathrm{s}}; a_{\lambda})\} > 0$,

$$\vartheta = \frac{M(c; a_{\lambda})}{\delta \sum_{a_{\mathrm{s}} \in S} \{\hbar_{\mathrm{s}} \cdot M(a_s; a_{\lambda})\}}, \quad (8)$$

δ is a constant used to measure the influence degree of redundancy between features in the feature set on classification accuracy, and it can be set according to the actual situation. The parameter called the feature Relation-Redundancy Coefficient ($\mathrm{R}^2\mathrm{C}$) that measures the redundancy of the selected feature set is expressed by a nonnegative number $\mathfrak{R}$:

$$\mathfrak{R} = \begin{cases} \infty, & \sum_{a_{\mathrm{s}} \in S} \{\phi_{\mathrm{s}} \cdot M(a_{\mathrm{s}}; a_{\lambda})\} = 0 \text{ or } S = \varnothing, \\ \vartheta, & \text{others.} \end{cases} \quad (9)$$

In Θ, the Relation-Redundancy Coefficient has the following four effects:

(i) When $\mathfrak{R} = 0$, the correlation of candidate features a_{λ} and the identification category c is zero; a_{λ} is an irrelevant feature of Θ.

(ii) When $0 < \mathfrak{R} < 1$, the redundancy of the candidate features a_{λ} and a_s is stronger than a_{λ} and other features; then, it is a redundancy feature.

(iii) When $\mathfrak{R} > 1$, the correlation between the candidate feature a_{λ} and the identification category c is stronger than the redundancy of the candidate features a_{λ} and a_{s} and brings new information for classification; then, it is called the correlation feature. Here we set a threshold θ ($\theta > 1$) based on the actual values for the correlation features. The features are strong correlation features when $\mathfrak{R} \ge \theta$; otherwise, they are weak correlation features.

(iv) When $\mathfrak{R} = \infty$, it only needs to analyze the mutual information $M(c; a_{\lambda})$ between a_{λ} and the identification category c; the corresponding a_{λ} of the maximum value $\mathfrak{R}$ can be selected into the set S.

According to the abovementioned effects, the optimal feature set $S = \{a_j \mid j = 1, 2, \ldots, \varphi\}$ including the φ features is finally obtained.

Given the mutual information and redundancy of the features, the empirical index α is given by the expert. Using the mutual information method to obtain the comprehensive weight w_j of the feature a_j in the comment space Θ,

$$w_j = \alpha \times \frac{M(c; a_{\lambda}; S)}{\sum_{a_{\lambda} \in S} M(c; a_{\lambda}; S)} + (1 - \alpha) \frac{M(c; a_j)}{\sum_{a_j \in S} M(c; a_j)}. \quad (10)$$

As an important parameter of the model, w_j plays an important role in the accuracy of the classification.

3.3. Emotional Polarity Selection Based on Mutual Information Weight. Based on the corpus data in the basic database, we obtain the optimal subset with the least feature redundancy and the relative weights of each feature in the feature set and calculate the emotion value of the unmarked corpus in the marked feature based on this weight. The specific steps are as follows:

(i) Extract the emotional words from unmarked corpus and convert them to a basic corpus.

(ii) According to the basic corpus, the optimal features including weights that remove redundant features are filtered out.

(iii) The eigenvalues corresponding to the feature nouns are assigned according to the emotional dictionary NTUSD compiled by Taiwan University; the positive word is assigned 1, the negative word is assigned −1, and the emotional value according to the weight is calculated (which ignores the impact of the adverb or grammatical structure for the emotional value) and the emotional threshold based on the basic corpus is set.

(iv) Judge the polarity and accuracy of the test corpus according to the weights and emotional thresholds based on the training library.

4. Numerical Experiment

Our experimental analysis is performed between mutual information method and emotion lexicon, TI-IDF and SVM. Using four datasets crawled from the Haodf's online platform to evaluate the performance of our proposed ALEIM method, the experiment is divided into four aspects:

(i) The datasets used in the experiments.

(ii) The overall flow and evaluation measures of the experiments.

(iii) The description of the experimental details by using the four datasets crawled from the Haodf's online platform.

(iv) The discussion of the experiments.

4.1. Datasets. The experimental datasets are crawled from the Haodf's online platform. These medical service comments are extracted using the octopus, and then the word segmentation is reorganized using Java programming, and each sentence in the comment is split into the metamatrix structure of "noun + verb." We first select 100 doctors and randomly collect 750 data in their comment areas and construct four basic corpus training libraries with different comments based on the above data, which is shown in Figure 1.

The number of positive and negative comments in the four basic corpus training libraries varies, and the positive comments ratio is higher than the negative ones. Due to the random extraction of the comment data as a corpus training library, the distribution of positive and negative comments in the training library is uncertain. Such randomly extracted data are used as training corpus data, which can test not only the dependence of different classification algorithms based on different category numbers but also the learning ability of the specific marked category based on a small sample. 400 data are prepared as the test data in Table 2, including 200 positive comments and 200 negative comments to test the accuracy of the training library for emotion classification under different algorithms.

When feature extraction is performed, the features extracted from the corpus with 100 data are all included in the other volumes corpus; the features extracted from the corpus with 150 data are all included in the corpus of 200, 300, and 400 data; the corpuses of 200 and 300 data extract the same features; the feature number extracted from the test corpus with 400 data is 42, and an additional feature is extracted from the corpus with the 200 and 300 data.

Since the data are randomly crawled, the corpus has low data repeatability between each other, so it can approximate that the probability of new features appearing decreases rapidly as the selected comment corpus data increase. Therefore, the amount of comment data for a suitable training corpus is determined, and the extracted features can contain almost all the features included in the medical comments (some special features extracted by small probability often not related to the medical service itself). This indicates that the features in comments often have limitations compared with traditional commodity comments due to the uniformity and standardization of medical services. The general commodity comments are not fixed due to the feature attributes of products; the products are highly different, and different products often contain unique features, which often affect the overall polarity of the comments. Therefore, commodity comments have high requirements for feature extraction, and it is necessary to continuously update the extracted features based on a large amount of data to achieve accurate classification of emotional polarity. Since the medical service does not have the variability of general commodity, the features of the comments are limited, so selecting a certain amount of data extract features can almost involve all the features in the medical service comments.

4.2. Experimental Design and Evaluation Measures. We employ Taiwan University NTUSD Simplified Chinese

Figure 1: The preparation process of experimental datasets in this paper.

Table 2: The test data for emotion classification under different algorithms.

The number of data	Positive	Negative	The number of feature	Used for
100	70	30	37	Training corpus
150	100	50	39	Training corpus
200	120	80	41	Training corpus
300	180	120	41	Training corpus
400	200	200	42	Test corpus

Emotion Dictionary Corpora for emotion and emotion classification. The overall flowchart of experiments in this paper is shown in Figure 2.

The SVM and feature weight algorithm used in this experiment are implemented by using MATLAB. Among them, the mutual information algorithm and the IDF algorithm calculate the feature weight by using the basic corpus and then combine the emotion dictionary NTUSD to calculate the emotion value of the corpus in the training library and set the emotional threshold according to the corpus data (calculate the positive and negative comments, respectively, and then use the weighted average of the two types emotional mean as the emotional threshold). The emotional polarity of the test corpus based on the feature weight and the threshold is judged. We have selected the following indicators as evaluation indicators:

(i) True positive: originally positive emotions, classified as positive emotions.

(ii) True negative: originally negative emotions, classified as negative emotions.

(iii) False positive: originally negative emotion, classified as positive emotion.

(iv) False negative: originally positive emotion, classified as negative emotion.

The accuracy reflects the ability of the classifier to determine the entire sample: the positive decision can be positive and the negative decision negative and can be expressed as

$$A = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN})}. \tag{11}$$

The precision reflects the proportion of the true positive sample in the positive case determined by the classifier and can be expressed as

$$P = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FP})}. \tag{12}$$

The recall reflects the proportion of positive cases that are correctly judged as the total positive examples and can be expressed as

$$R = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})}. \tag{13}$$

4.3. Implementation Details of Experiments. Figure 3 shows that the accuracy of the classification algorithm of IDF and mutual information considering the feature weight and the emotion dictionary-based classification algorithm are significantly higher than the SVM algorithm using the Gaussian kernel function for four basic corpus libraries. As the number of samples increases, the accuracy of emotion lexicon maintains constant basically. However, as the number of samples increases, the accuracy of mutual information

Figure 2: The overall flowchart of experiments in this paper.

Figure 3: The accuracy of the four methods used in this paper.

increases rapidly and is higher than the other three methods. As can be seen from Figure 3, the performance of mutual information method is better than the other three methods. SVM algorithm requires that the number of different types in the training database must be substantially the same to achieve optimal learning. However, the online medical service comments have a large proportion of positive and negative polarities; support vector machine algorithm is difficult to achieve the optimal data ratio. Constructing the training library according to the actual ratio often leads to the identification of negative polarity data with less proportion and leads to lower overall accuracy.

Table 3 illustrates the detailed significant test results of accuracy between mutual information and other three methods in terms of the p value on the four basic corpus libraries. As can be seen from the table, the mutual information method is superior to the other three methods on 150 data, 200 data, and 300 data. The results show that when the sample size increases, p values between mutual information and other three methods are less than 0.05. This means the classification results of mutual information method are significantly better than the other three methods.

Figure 4 shows that the precision of the classification algorithms of IDF and mutual information considering the feature weight are slightly higher than the other two algorithms. The mutual information algorithm has lower precision when the training data are less, and the precision is improved with the training data increase but is slightly lower than the IDF weighting algorithm.

Table 4 illustrates the detailed significant test results of precision between mutual information and other three methods in terms of the p value on the four basic corpus libraries. As can be seen from the table, there is a significant difference among the mutual information method, emotion lexicon, and SVM methods because p values between mutual information and other two methods are less than 0.05, but when the number of samples increases, there is no significant difference between the mutual information method and TI-IDF method.

Figure 5 indicates that our proposed algorithm which considers the weight of each feature has the superior performance than other two comparison approaches. Since the negative emotion polarity data of the training inventory are less, the recall of the other two algorithms is extremely low, and the weight of the feature weight algorithm is not dependent on the weight of the data category, so the learning effect on the limited negative polarity data is better, and the recognition of the negative emotion data in the test data is higher. The recall rate of the mutual information algorithm is significantly higher than that of the IDF algorithm. It shows that the mutual information algorithm considering the feature weight has strong recognition ability for negative emotion.

Table 5 illustrates the detailed significant test results of recall between mutual information and other three methods in terms of the p value on the four basic corpus libraries.

Table 3: The detailed significant test results of accuracy between MI and other methods.

Datasets	Metrics	Methods		
		MI and emotion lexicon	MI and TI-IDF	MI and SVM (RBF)
100 data	p value	0.0906	0.0063	0.1304
150 data		0.0487	0.0197	0.0043
200 data		0.0435	0.0437	0.0226
300 data		0.0255	0.0432	0.0021

Figure 4: The precision of the four methods used in this paper.

Table 4: The detailed significant test results of precision between MI and other methods.

Datasets	Metrics	Methods		
		MI and emotion lexicon	MI and TI-IDF	MI and SVM (RBF)
100 data	p value	0.0413	0.0043	0.0343
150 data		0.0387	0.0667	0.0342
200 data		0.0234	0.0731	0.0106
300 data		0.0055	0.0902	0.0049

Figure 5: The recall of the four methods used in this paper.

Table 5: The detailed significant test results of recall between MI and other methods.

Datasets	Metrics	Methods		
		MI and emotion lexicon	MI and TI-IDF	MI and SVM (RBF)
100 data	p value	0.0313	0.1025	0.0034
150 data		0.0478	0.0706	0.0147
200 data		0.0443	0.0831	0.0321
300 data		0.0142	0.0502	0.0079

From this table, it can be seen that there is a significant difference among the mutual information method, emotion lexicon, and SVM methods because p values between mutual information and other two methods are less than 0.05, but there is no significant difference between the mutual information method and TI-IDF method.

Figure 6 shows the comparison of 41 feature weights in 300 training library corpora under mutual information weighting algorithm and TI-IDF algorithm. It can be seen that the weights of the two algorithms of feature 1, feature 25, feature 35, and feature 5 and feature 41 are quite different, corresponding to the condition, attitude, doctor, and side effects and consultation.

The mutual information algorithm weights are significantly higher than the IDF weights for the first three features. These three features are commonly found in medical comments. IDF algorithm believes that these comments with high frequency are of lower importance and are filtered to give small weights, while mutual information algorithm according to the high mutual information value and low redundancy of the identification feature gives the high weight, and such weight causes the mutual information algorithm to have a lower accuracy than the IDF in identifying the positive emotional polarity. These features are used as the basic features of comments; it tends to have a lower guiding effect on the emotional polarity of the reviewer in the positive comments and a primary role in the orientation of the emotional polarity in the negative comments.

In the latter two features, the mutual information algorithm weight is significantly lower than IDF algorithm. These features belong to the low-frequency features and appear 6 times and 7 times in 300 data, respectively. The IDF algorithm assumes that low-frequency words can more affect the emotional polarity of the comments for the comment library as a whole, and the mutual information algorithm considers these features to be small mutual information values with high redundancy and low weight. The experiments show that these two features actually weaken the emotional polarity of the comments. The IDF algorithm classifies all the errors in the six test comment categories with the above two features, and the mutual information recognition rate is 100%.

4.4. Discussion of Experiments. From the above experimental analysis, we can obtain that mutual information is the most appropriate method to solve such problem. It shows good performance in terms of accuracy when the number of samples increases and only requires a moderate computational cost for solving emotion classification problems of short texts for online medical knowledge sharing community. However, in terms of precision and recall, there is no significant difference between the mutual information method and TI-IDF method, but Figure 6 shows that the accuracy of IDF algorithm in identifying negative emotion polarity is significantly lower than that of mutual information algorithm. Experiments show that low-frequency words existing in medical review data are often redundant features, and mutual information algorithm has higher accuracy for the identification of such redundant features. However, our experiments need to be further improved due to only four basic corpus libraries involved in the experiment. Therefore, we plan to crawl more different types comments on online medical knowledge sharing community to achieve parameter optimization and method performance promotion.

Figure 6: The difference between mutual information algorithm and TI-IDF algorithm.

5. Conclusions

Emotion analysis has been widely used in many fields and becomes an important tool for extracting emotional information of the comments. Emotional analysis in medical knowledge sharing community is still relatively lacking compared with the general commodities. The information recipients in the medical knowledge sharing community are more concerned with the intensity of the emotional words in the comments or the overall evaluation. In this research, we propose an adaptive learning emotion identification method based on mutual information feature weight, which captures the correlation and redundancy of features. Its effectiveness is verified on the dataset crawled from the Haodf's online platform, and we employ Taiwan University NTUSD Simplified Chinese Emotion Dictionary Corpora for emotion classification. Finally, the experimental results show that the proposed ALEIM method can achieve good performance, especially in terms of the low-frequency words feature extraction in comments of the online medical knowledge sharing community.

Data Availability

The experimental data come from the Haodf's online platform and can be crawled from https://www.haodf.com.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (grant no. 71571105) and the

Fundamental Research Funds for China's Central Universities (grant no. 63172074).

References

[1] Y. Wang, Y. Rao, X. Zhan, H. Chen, M. Luo, and J. Yin, "Sentiment and emotion classification over noisy labels," *Knowledge-Based Systems*, vol. 111, pp. 207–216, 2016.

[2] D. M. El-Din and M. Hussein, "A survey on sentiment analysis challenges," *Journal of King Saud University Engineering Sciences*, vol. 30, no. 4, pp. 330–338, 2018.

[3] L. Jia, C. Yu, and W. Meng, "The effect of negation on sentiment analysis and retrieval effectiveness," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management-CIKM '09*, Hong Kong, China, November 2009.

[4] A. Hogenboom, P. Van Iterson, and B. Heerschop, "Determining negation scope and strength in sentiment analysis," in *Proceedings of the IEEE International Conference on Systems*, IEEE, Anchorage, AK, USA, October 2011.

[5] B. Alexandra and S. Ralf, "Rethinking sentiment analysis in the news: from theory to practice and back," in *Proceedings of the WOMSA 2009*, pp. 1–12, Seville, Spain, Novmber 2009.

[6] S. Mukherjee and P. Bhattacharyya, *Feature Specific Sentiment Analysis for Product Reviews*, Springer, Berlin, Germany, 2012.

[7] K. Chetan and M. Atul, "A scalable lexicon based technique for sentiment analysis," *International Journal in Foundations of Computer Science & Technology*, vol. 4, no. 5, pp. 267–307, 2014.

[8] M. Mohey, M. O. Hoda, and O. Ismael, "Online paper review analysis," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 9, pp. 99–107, 2015.

[9] B. Agarwal, N. Mittal, P. Bansal, and S. Garg, "Sentiment analysis using common-sense and context information," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 715730, 9 pages, 2015.

[10] G. Katz, N. Ofek, and B. Shapira, "Consent: context-based sentiment analysis," *Knowledge-Based Systems*, vol. 84, pp. 162–178, 2015.

[11] A. Weichselbraun, S. Gindl, and A. Scharl, "Enriching semantic knowledge bases for opinion mining in big data applications," *Knowledge-Based Systems*, vol. 69, pp. 78–85, 2014.

[12] F. Bravo-Marquez, M. Mendoza, and B. Poblete, "Meta-level sentiment models for big social data analysis," *Knowledge-Based Systems*, vol. 69, pp. 86–99, 2014.

[13] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, vol. 28, no. 2, pp. 15–21, 2013.

[14] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up: sentiment classification using machine learning techniques," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, PA, USA, July 2002.

[15] C. Pascoal, M. R. Oliveira, A. Pacheco, and R. Valadas, "Theoretical evaluation of feature selection methods based on mutual information," *Neurocomputing*, vol. 226, pp. 168–181, 2017.

[16] Y. Wu, B. Liu, and W. Wu, "Grading glioma by radionics with feature selection based on mutual information," *Journal of Ambient Intelligence & Humanized Computing*, vol. 9, pp. 1–12, 2018.

[17] G. U. Chao, Y. Yang, and X. X. Zhang, "Feature selection for transformer fault diagnosis based on maximal relevance and minimal redundancy criterion," *Advanced Technology of Electrical Engineering & Energy*, vol. 37, no. 7, pp. 84–89, 2018.

[18] S. Poria, A. Gelbukh, E. Cambria, A. Hussain, and G.-B. Huang, "Emosenticspace: a novel framework for affective common-sense reasoning," *Knowledge-Based Systems*, vol. 69, pp. 108–123, 2014.

[19] L. Shen, H. Chen, Z. Yu et al., "Evolving support vector machines using fruit fly optimization for medical data classification," *Knowledge-Based Systems*, vol. 96, pp. 61–75, 2016.

[20] V. Loia and S. Senatore, "A fuzzy-oriented sentic analysis to capture the human emotion in web-based content," *Knowledge-Based Systems*, vol. 58, pp. 75–85, 2014.

[21] T. Chalothom and J. Ellman, *Simple Approaches of Sentiment Analysis via Ensemble Learning, Information Science and Applications*, Springer, Berlin, Germany, 2015.

[22] F. N. Ribeiro, M. Araújo, P. Gonçalves et al., "SentiBench-a benchmark comparison of state-of-the-practice sentiment analysis methods," *EPJ Data Science*, vol. 5, no. 1, 2016.

[23] P. D. Turney, "Thumbs up or thumbs down: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, USA, July 2002.

[24] Y. Ainur, Y. Yisong, and C. Claire, "Multi-level structured models for document-level sentiment classification," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, USA, October 2010.

[25] N. Farra, E. Challita, and R. Assi, "Sentence-level and document-level sentiment mining for arabic texts," in *Proceedings of the IEEE International Conference on Data Mining Workshops*, IEEE, Vancouver, Canada, December 2011.

[26] H. Zhou and F. Song, "Aspect-level sentiment analysis based on a generalized probabilistic topic and syntax model," in *Proceedings of the FLAIRS 2015*, Hollywood, FL, USA, May 2014.

[27] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: a survey," *AIN Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[28] H. H. Do, P. Prasad, A. Maag, and A. Alsadoon, "Deep learning for aspect-based sentiment analysis: a comparative review," *Expert Systems With Applications*, vol. 118, pp. 272–299, 2019.

[29] N. Liu, E.-S. Qi, M. Xu, B. Gao, and G.-Q. Liu, "A novel intelligent classification model for breast cancer diagnosis," *Information Processing & Management*, vol. 56, no. 3, pp. 609–623, 2019.

[30] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, no. 6, pp. 14–46, 2015.

[31] N. Liu, J. Shen, M. Xu et al., "Improved cost-sensitive support vector machine classifier for breast cancer diagnosis," *Mathematical Problems in Engineering*, vol. 2018, Article ID 3875082, 13 pages, 2018.

[32] B. Agarwal and N. Mittal, "Prominent feature extraction for review analysis: an empirical study," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 3, pp. 485–498, 2014.

[33] H. Zhang, D. Wang, H. Xu, and S. Sun, "Sentiment classification of micro-blog public opinion based on convolution neural network," *Journal of the China Society for Scientific and Technical Information*, vol. 37, no. 7, pp. 695–702, 2018.

[34] C. Hung, "Word of mouth quality classification based on contextual sentiment lexicons," *Information Processing & Management*, vol. 53, no. 4, pp. 751–763, 2017.
[35] B. Rojas, "Deep learning for sentiment analysis," *Language & Linguistics Compass*, vol. 10, no. 12, pp. 205–212, 2016.
[36] Z. Yang, R. Salakhutdinov, and W. Cohen, "Multi-task cross-lingual sequence tagging from scratch," 2016, https://arxiv.org/abs/1603.06270.
[37] D. Marcheggiani, A. Frolov, and I. Titov, "A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling," in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, Vancouver, Canada, August 2017.
[38] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, August 2014.
[39] M. Saeidi, G. Bouchard, M. Liakata, and SentiHood, "Targeted aspect based sentiment analysis dataset for urban neighbourhoods," in *Proceedings of the 26th International Conference on Computational Linguistics*, Osaka, Japan, May 2016.
[40] S. Poria, I. Chaturvedi, E. Cambria, and L. D. A. Sentic, "Improving on LDA with semantic similarity for aspect-based sentiment analysis," in *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Vancouver, Canada, July 2016.

Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 4368036, 19 pages
https://doi.org/10.1155/2019/4368036

Review Article

Speech Technology Progress Based on New Machine Learning Paradigm

Vlado Delić,[1] Zoran Perić,[2] Milan Sečujski,[1] Nikša Jakovljević,[1] Jelena Nikolić,[2] Dragiša Mišković,[1] Nikola Simić,[2] Siniša Suzić,[1] and Tijana Delić[1]

[1]*University of Novi Sad, Faculty of Technical Sciences, 21000 Novi Sad, Serbia*
[2]*University of Niš, Faculty of Electronic Engineering, 18000 Niš, Serbia*

Correspondence should be addressed to Nikša Jakovljević; jakovnik@uns.ac.rs

Received 22 February 2019; Revised 29 April 2019; Accepted 9 May 2019; Published 25 June 2019

Academic Editor: Paolo Gastaldo

Copyright © 2019 Vlado Delić et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Speech technologies have been developed for decades as a typical signal processing area, while the last decade has brought a huge progress based on new machine learning paradigms. Owing not only to their intrinsic complexity but also to their relation with cognitive sciences, speech technologies are now viewed as a prime example of interdisciplinary knowledge area. This review article on speech signal analysis and processing, corresponding machine learning algorithms, and applied computational intelligence aims to give an insight into several fields, covering speech production and auditory perception, cognitive aspects of speech communication and language understanding, both speech recognition and text-to-speech synthesis in more details, and consequently the main directions in development of spoken dialogue systems. Additionally, the article discusses the concepts and recent advances in speech signal compression, coding, and transmission, including cognitive speech coding. To conclude, the main intention of this article is to highlight recent achievements and challenges based on new machine learning paradigms that, over the last decade, had an immense impact in the field of speech signal processing.

1. Introduction

According to Kuhn's theory of scientific revolutions [1], the science makes progress through the revolutionary changes of prevailing scientific paradigms, where a paradigm represents a set of beliefs and values and technical and methodological procedures common to a scientific community. Paradigms define frames and models for solving scientific challenges. New solutions come with new generations who are ready to accept new truths and interdisciplinary approaches. New paradigms appear suddenly and provide new lights to a scientific problem, based on synergy of particular and specialized knowledge consolidated into a functional and coherent unity. Speech technology community investigates spoken language processing as an interdisciplinary research area (Figure 1), [2]. After a short retrospective of the main scientific paradigms based on the knowledge of speech production and auditory perception, this article presents new achievements and perspectives based on the new machine learning paradigm related to neuroscience and advanced signal processing.

The roots of speech signal processing research were closely related to the needs of speech signal digitization. The pioneering solutions were deployed during the World War II due to a need of secure communication between the Allies. The system was named SIGSALY, and it utilized pulse-code modulation (PCM) to enable the first transmission of voice using digital equipment [3]. In the next decades, the focus of researchers was on standardizing rules of digital telephony in order to provide high quality of reconstructed speech signal in the wide range of speech signal variances [4–7]. The compression paradigms regarding these systems have not changed significantly for decades. Particularly, the focus of research has slightly been moved toward improving the signal quality at the receiving end or toward reducing the required bit rate [8–13]. However, the significant

Figure 1: Interdisciplinary nature of speech technologies, i.e., spoken language processing (adopted from [2]).

development of computer technology in the last decade has enabled research into new approaches to advanced speech signal processing including adaptive machine learning methods [14]. Recent trends include cognitive speech coding so that there is a paradigm shift from perceptual (auditory) toward cognitive (auditory plus cortical) speech signal processing [15].

Modern speech technology systems rely on interdisciplinary research in the areas of multimodal signal processing and artificial intelligence, and a number of methods and algorithms have been developed with the aim of solving various problems: dialogue systems based on speech recognition and synthesis, including emotional speech, speaker identification and verification, as well as speech signal coding and transmission, denoising and detection of signals in the presence of noise, quality enhancement, and medical diagnostics based on the analysis of human voice. Recent progress in most of these speech technology topics will be discussed in more details in the following sections.

Spoken language processing (SLP) is an interdisciplinary research area that has attributes of computational intelligence. SLP lies in the intersection of linguistics, psychology, engineering, and artificial intelligence (AI) [2]. Advanced signal processing and machine learning methods are positioned in the adopted view to the interdisciplinary character of SLP, and both interconnections and intersections of different disciplines are shown and presented in a novel point of view (Figure 1). Instead of using the original term "pattern processing" in Figure 1, we have opted for the more common term "signal processing and machine learning (SP&ML)," which represents the overlap between the community of engineering and AI disciplines. With linguistic aspects included, they compose the natural language processing (NLP) field. Human-computer interaction (HCI) draws experience and methodology from the fields of engineering and psychology, and with the knowledge from linguistics included, they form a basis for the study and development of dialogue systems.

The interconnection of psycholinguistics and AI is the foundation of cognitive science or neurolinguistics. Neurolinguistics has been treated here as the neuroscience of speech. Neurolinguistics is presented in Figure 1 as dominantly linguistics discipline but connected to AI through computer linguistics which is on the intersection between AI and linguistics from one side and also connected to psychology through psycholinguistics, which is on the intersection between psychology and linguistics from other side. Neurolinguistics is on the opposite side from the engineering point of view. The neuroscience of speech can also be considered as an area of cognitive science, and cognition is inherent part of both speech perception (in the phase of understanding, the meaning of the message conveyed by spoken language) and speech production (in the phase of composing, a message intended to convey a certain meaning). Finally, SLP combines knowledge from the interdisciplinary areas of SP&ML, HCI, psycholinguistics, and computer linguistics, or more precisely NLP, cognitive sciences, dialogue systems, and information access.

Speech technologies are based on speech signal processing that spans a wide range of topics, while the focus in this review article is on three areas where the authors have the most expertise:

(i) Fundamental topics (speech analysis and synthesis, sound waves and speech features, speech production, auditory perception, and cognition including the linguistic aspect)

(ii) Dialogue systems based on speech recognition and text-to-speech synthesis (emotional speech recognition and text-to-speech synthesis including voice and style conversion)

(iii) Speech coding, compression, and transmission

Speech technology fields within the scope of the paper are presented in Figure 2 as a unified framework that encompasses covered topics, showing their complementarity, ranges and borders, interconnections, and intersections in the interdisciplinary area of SLP.

A brief retrospective and some perspectives of the speech technology fields shown in Figure 2 are presented in the following sections. Fundamental topics are shown in the middle of Figure 2 and presented in Section 2, covering speech production and perception analysis, including cognitive and linguistic point of views. More details related to the progress in speech recognition and speech synthesis, as well as their contributions to a new generation of human-machine speech dialogue systems, are presented in Section 3. Finally, the progress in speech signal compression, coding, and transmission is presented in Section 4, including contributions of the authors to the area. Most of these advancements are based on the new deep learning paradigm and our better understanding of neuroscience

Figure 2: Unified framework that encompasses speech signal processing fields in the scope of the article.

and modelling of cognitive aspects of spoken language communication.

2. Progress in Speech Analysis and Knowledge of Spoken Language Nature

Knowledge related to the nature of spoken language is essential for efficient coding and transmission as well as satisfactory real-time human-machine speech interaction. Speech models based on either speech production or auditory perception were inherent parts of most successful algorithms. Most recent neuro-inspired computational models are based on knowledge of cognitive speech processing models [16]. After a brief review of sound pressure waves and speech signal features, speech production and auditory perception including cognitive and linguistic points of view will be elaborated in more detail in the following subsections.

2.1. Sound Pressure Waves and Speech Signal Features. Sound propagates as a continuum of acoustic waves (sound pressure), and, once received, it can be recorded, digitized, coded, transmitted, processed, and reproduced. In case of speech sounds, frequencies relevant for recognizing what was said and who has said it are located mostly below 4 kHz and hardly ever above 7-8 kHz, which is just a portion of the entire frequency range of the human auditory sense [17]. This fact was the basis of the design of analogue telephone communication systems, including the choice of microphones used. For that reason, a speech signal is sampled at 8 kHz (for a basic level of quality) or 16 kHz (if a higher level of quality is desired). It is also well known that the dynamic range from the softest to the loudest sounds in average human speech is approximately 40 dB. Even if whisper and elevated voice are included, this dynamic range is rarely above 50 dB [14]. For these reasons, the requirements for a microphone needed to record voice are typically less strict than in case of recording, e.g., music. As to quantization, it is known that each bit contributes to the signal-to-noise ratio (SNR) by 6 dB, which means that the quantization noise is practically inaudible if 8 bits are used for coding every sound sample [4]. Thus, the typical case of using the sampling rate of 8 kHz and 8 bits per sample produces the bit rate of 64 kbits/s. A lot of effort has been invested to reduce this rate without significant loss of quality at the receiving side, and Section 4 is devoted to this subject.

Digitalization including quantization is the basis of all digital speech processing techniques. If the aim is to represent speech compactly and robustly, as is the case in automatic speech recognition or most types of speech coding for transmission, one of the basic questions is the selection of relevant features that will enable fast, accurate, and robust recognition of speech (or the speaker, language, or even emotion), and/or fast and efficient speech coding for transmission without significant loss of quality. Linear predictive coding (LPC) and LPC analysis have fundamental significance in speech signal modelling and speech feature estimation [18]. Many speech coding schemes are based on LPC including Low Delay-Code Excited Linear Prediction (LD-CELP) coding scheme defined by G.728 standard, Conjugate Structure Algebraic Code-Excited Linear Prediction (CS-ACELP) coding scheme defined by G.729A standard, Algebraic Code-Excited Linear Prediction coding scheme defined by G.723.1 standard, and Adaptive Multi-Rate Wide-Band (AMR-WB) coding scheme defined by G.722.2, standards which are used in today mobile voice communication and VoIP [5, 7].

One of speech production models is also based on LPC analysis and provides speech feature sets describing speech spectrum, which is most important for speech recognition [19]. The main scope of speech signal and data processing in real time (or limited time) is to reduce the amount of data (speech features), while providing high quality of representation of such a reduced signal, that is, data source. The realization of this goal is supported by statistical signal and data processing as well as methods and algorithms which deal with signal and data reduction [20]. The most efficient methods and algorithms incorporate adaptation, and these topics will be elaborated in more details in the next sections.

2.2. Speech Production and Auditory Perception. Figure 3 shows a block diagram of both speech production and perception. Text-to-speech synthesis (TTS) and automatic speech recognition (ASR) are shown in parallel as corresponding processes performed by machines. Speech and language are learned, while the sense of hearing is innate. There are a lot of differences among human and machine speech production and perception, but the increase in the ability of machine learning paradigms to simulate human speech production mechanism, as well as auditory perception and cognition abilities, will inevitably bring about an increase in the accuracy of ASR and naturalness of TTS.

Figure 3: Block diagram of speech production and speech perception and corresponding processes performed by machines carrying out text-to-speech synthesis (TTS) and automatic speech recognition (ASR).

Speech communication between humans begins and ends at the cognitive level of message composition and interpretation. Taking into account the average speech rate of 10–12 phones per seconds and the number of phones in a language, which typically corresponds to 5 or 6 bits needed to encode them, a speech message conveyed as text could be considered to correspond to a bit rate of 50–60 bits per second. The speaker plans not only what to say but also how to say it—(s)he controls the volume, speech rate, and intonation (prosody)—any of which can carry linguistic, and also paralinguistic and extralinguistic information [21]. With that information added, the bit rate can be considered to increase to several hundreds of bits per second.

Once the speaker decides what to say and how to say it, an appropriate sound wave is produced through nervous and muscular activity [22, 23]. In that, phones are not pronounced in isolation, but the articulatory targets required for corresponding phonemes are rarely reached, leading to the coarticulation effect, which aggravates the task of ASR. Most often, the entire speech apparatus is considered through the source-filter model, where the activity of vocal folds defines the excitation and the remainder of vocal tract acts as a filter and shapes the sound spectrum [19, 21]. Besides being dependent on the phone, the acoustic features of the speech signal at a particular moment also carry information relevant to the speaker and thus represent a biometrical feature which can reveal the speaker's identity [24] and possibly other factors related to the speaker or to the message. Including the influence of speaker variability, the bit rate at this level increases to several thousand bits per second. This segment of speech communication is studied by articulatory and acoustical phonetics, and its machine counterpart is TTS, namely, the module charged with the production of the artificial speech signal itself.

Distribution of speech sample amplitudes is nonuniform, and this knowledge is used in nonuniform speech signal coding defined by μ-law and A-law [25], while some new research results provide better solutions based on adaptive algorithms. The speech production mechanism articulates a series of phonemes nonuniformly, according to an empirical statistical law formulated by George Kingsley Zipf, a linguist [26], referring to the principle of the least effort from evolutionary biology field: interlocutors try to understand each other using phonemes and words that are easier for production and perception in a particular context. The knowledge of phoneme and word statistics has been introduced into ASR algorithms long ago, and stochastic speech models like Hidden Markov model (HMM) [27] were the prevailing scientific paradigm and represented the state of the art in speech recognition and synthesis community for decades.

On the other side, the continuum of acoustic waves reaches the ear of the listener and certain frequencies excite the eardrum, and over the malleus, incus, and stapes, they excite the cochlea, where spectral analysis is performed, based on the movement of the basilar membrane, whose length is about 35 mm [17, 22, 23, 25, 28]. The hair cells in the cochlea respond to different sounds based on their frequency so that high-pitched sounds stimulate the hair cells in the lower part of the cochlea, while low-pitched sounds stimulate the upper part of the cochlea [28]. Thus formed neural impulses are sent to the central auditory system in the brain [22], and based on spectral differences, the brain recognizes relevant acoustic differences and attempts to recover the string of phones that the original message was composed of, taking into account its language model (at the level of morphology, syntax, semantics, and pragmatics). It can thus be considered that the task of ASR is to reduce the bit rate of, e.g., 64 kbits/s (digitized speech) to a bit rate of 50–60 bits/s (plain text), which would correspond to the textual contents of the message without speech prosody.

However, speech perception, which principally relies on the sense of hearing, is a nonlinear process. As is the case

with other human senses (vision, taste, touch, and smell), auditory perception of both sound pressure level (SPL) and fundamental frequency (f0, pitch) follows the Weber–Fechner law [28] from psychophysics: a change perceived as linear corresponds to an exponential change in the physical stimulus. Apart from SPL and pitch, perception of sound is affected by the distribution of sound energy across frequencies, i.e., the spectrum of the sound, which usually represents a mixture of a sequence of discrete frequency components (timbre), as in the case of periodic sounds, and a continuous mix of nonharmonic or random frequency components, as in the case of various types of noise [22, 28]. This is why common speech features like cepstral coefficients are considered to be located at frequencies rescaled from Hz to mel-scale–MFCC; they are estimated by cepstral analysis from speech frames of 20–30 ms together with their first and second derivatives calculated from several successive frames [29].

Auditory scene analysis is the process by which the auditory system separates individual sounds in natural-world situations [30, 31]. Regardless of whether sound is received by a human ear or a microphone, the incident sound pressure wave represents a sum of pressure waves coming from different individual sources, which can be either human voices or any other sound sources. These sounds usually overlap in both time and frequency. Nevertheless, the human auditory system is usually able to concentrate on an individual sound source at a time [23, 31]. While listening and separating one source, the listener constructs a separate mental description for that source. Although he/she cannot actively listen to two sound sources simultaneously, he/she can switch immediately his/her attention from one to the other [30]. For example, if a student listens to the teacher, he ignores the noise from LCD projector and a colleague who may be speaking to him; if he switches the focus to his colleague, he cannot actively listen to the teacher anymore. Furthermore, if a human listener follows the context, he/she is able to reconstruct some phonemes or entire words that he/she may not be able to hear for some reason. Humans are as successful in sound separation as they are more experienced in real-word situations and they always analyse the incoming signal using heuristic processes. As the ultimate step of the hearing process, human auditory cortex constructs a cognitive representation of the received sound wave. Without the cognition step, sound waves coming to the ears are not perceived. Heuristic analysis is based on (ir)regularities in the sum of underlying sounds.

Individual sounds differentiate from each other in at least one of the following dimensions: time, space, and frequency spectrum [28, 31]. Temporal and spatial sensations in the human auditory system are presented in more details in [32]. In the time dimension, two sounds can have some onset/offset asynchrony. In a specific environment, binaural hearing enables the localization of sound sources, which is easier, but also often more important, in the horizontal plane where human ears are positioned than in the vertical plane. The spectrum of frequency components can determine the perceived pitch, timbre, loudness, and the difference in the spectra of sounds received by both ears enables the localization of sound sources [23, 31, 32]. Pitch is related to the fundamental frequency f0 in periodic sound waves such as musical tones or vowels in speech; their spectrum consists of f0 and its harmonics. Temporal variation of f0 results in melody in music and intonation in speech. Timbre represents a specific distribution in the intensities of f0s and its harmonics in the spectrum. Two renditions of the same tone from two different musical instruments, having the same f0, will have different timbres due to the difference in the relative intensities of particular harmonics (the spectral envelope), and as a result, they will sound different [22]. If a sound spectrum does not contain just harmonic tones (f0s and their harmonics), the spectrum is not discrete; sound spectrum is rich with frequency components in parts or in the entire frequency range of the human auditory sense. Such sounds, with a spectrum that is more or less continuous, are much more frequent in nature (e.g., noise of a car or a machine or any transient noise). Magnitudes of spectral components contribute to the loudness; sound pressure level is defined in dB relative to the threshold of hearing at 1 kHz (20 μPa) and has range 0–120 dB to the threshold of pain [17, 22]. To conclude, two sounds can be separated from each other in an auditory scene analysis according to the differences in loudness, pitch (f0, if present), and timbre or spectrum as a whole, as well as in their temporal and/or spatial variations that can create a variety of sound impressions.

Acoustic signals are received by a listener and transformed into linguistic and nonlinguistic categories, but it is not known exactly how. There is ongoing research on neurophysiology of speech communication using the latest advances in invasive and noninvasive human recording techniques, with the aim to uncover fundamental characteristics of cortical speech processing [16]. The research team in question has studied phonetic feature encoding and mechanisms of noise robust representation of speech in auditory cortex based on the evidence that humans and animals can reliably perceive behaviourally relevant sounds in noisy and reverberant environments.

Neuro-inspired computational models try to provide progress in artificial deep neural network (DNN) performance, based on better understanding of the representation and transformation performed by these models. A case study in ASR given in [33] attempts to identify the mechanisms that normalize the natural variability of speech and compares these mechanisms with findings of speech representation in the human auditory cortex. The aim is to compare DNNs with their biological counterparts, identify their limitations, and reduce the performance gap between biological systems and artificial computing. For example, a human is able to concentrate on one speaker voice and ignore other sounds and voices [23, 31], based on their differences in spatial positions, pitch, and timbre, coherence of changes in level and/or frequency, and time characteristics (onset/offset asynchrony) [30]. An algorithm aimed at focusing on one speaker in a group of many speakers based on deep attractor network is proposed in [34], based on similar principles. It has been shown that switching attention

to a new speaker instantly changes the neural representation of sound in the brain. An adaptive system should change the sensory representation in real time to implement novel, task-driven computations that facilitate the extraction of relevant acoustic parameters.

Human listeners have a remarkable ability to understand quickly and efficiently the world around them based on behaviour of known sound sources. Moreover, they are able to pay attention and focus on the meaning of speech of a particular speaker. Attentional focus can be integrated into HCI dialogue strategy [35], while data related to human cognitive effort can be used in postprocessing and improvement of the performance of ASR systems [36]. Humans are able not only to separate one speaker or concentrate only to one sound source but also to group more sound sources and hear, e.g., the entire orchestra as one musical sound based on harmonicity and synchrony of particular sound sources. Concurrent and sequential grouping processes are described in more details in [37].

The role of the nonlinearities in DNN in categorization of phonemes by their nonuniform and nonlinear warping of the acoustic space are studied in [38], as well as the way perceptual invariant categories are created. Biological neurons are able to dynamically change the synaptic efficacy in response to variable input conditions. It is called synaptic depression and when it is added to the hidden layers of a DNN trained for phoneme classification, ASR system becomes more robust to noisy conditions without explicitly being trained for them. The results from [39] suggest that more complete neuron models may further reduce the gap between the biological performance and artificial computing, resulting in networks that better generalize to novel signal conditions.

2.3. Engineering vs. Linguistic Point of View to NLP as a Typical AI Topic. The mechanism of speech production and the physical component of sound perception are relatively well-studied topics [22, 31], while cognitive aspects of speech communication still represent a widely open research area. All aspects of human-machine speech communication that are related to linguistics, such as natural language processing (NLP), cognitive sciences—neurolinguistics, and dialogue management (see Figure 1), represent great challenges to the scientific community. In the recent past, the development of speech technology and spoken dialogue systems has gained most momentum from the engineering disciplines, through the possibility of automatic learning from vast quantities of data, in terms of development of computational facilities, complex learning algorithms, and sophisticated neural model architectures addressing particular phenomena and problems of cognitive linguistics. At the same time, cognitive speech sciences mostly remain outside of the scope of the immediate interest of engineering disciplines relevant to speech technology development. Nevertheless, the knowledge in these areas overlaps in the concept and scope with machine learning, which, inspired by neurosciences, has brought about progress not only in human-computer interaction and computational linguistics but also in the area of spoken language processing, which lies in their intersection. This is indicated in Figure 1, which also shows a relatively wide gap between cognitive sciences (neuroscience) and psycholinguistics on one side and predominantly engineering disciplines on the other.

As regards the role of machine learning in the development of speech technology, it has offered a powerful alternative to models dependent on linguistic resources and modules performing particular linguistically motivated subtasks. Linguistic resources such as dictionaries and speech databases are typically quite expensive and time-consuming to collect and annotate, while the development of modules that compose a speech technology system requires deep domain knowledge and expert effort. In the last two decades, some of the tasks performed by rule-based systems or simpler machine learning methods have, one by one, been overtaken by neural networks. Namely, in the case of acoustic speech recognition, neural networks have been shown to outperform hidden Markov models (HMMs) in acoustic modelling [40] but have also outperformed classical N-gram language models in terms of generalization, using either architectures based on long short-term memory (LSTM) neurons [41] or recurrent neural networks (RNN) [42]. Solutions based on neural networks have been shown to reach human parity in tasks as complex as casual conversational speech recognition [43]. In combination with a range of data-synthesis techniques for obtaining large quantities of varied data for training, it is now possible to obtain an end-to-end ASR capable of outperforming state-of-the-art pipelines in recognizing clear conversational speech as well as noisy one [44, 45]. They have also been used in multimodal speech recognition, i.e., recognition of speech from audio and video [46]. The task of speech synthesis is a more language-dependent one, and in that it is more challenging since it aims to reintroduce the redundancy which is lost when speech is converted into text, and to do it in such a way that, among a multitude of prosodic renditions of a particular utterance, it produces one that the listener will consider acceptable in a given context. Here again, neural networks have shown to overperform classical models working on parameterized speech such as HMMs [47, 48] in acoustic modelling, and they have also been employed for prosody modelling [49] as well as modelling of acoustic trajectories [50]. Neural networks have also addressed the problem of a somewhat muffled character of synthesized speech due to the use of a vocoder, by performing synthesis of raw speech waveforms instead [51]. Finally, to overcome the need for sophisticated speech and language resources that require deep domain expertise, a range of end-to-end architectures were proposed, with the ultimate end that the system should be trained on pairs of text and audio, exploiting the capability of neural networks to automatically develop higher-level abstractions [52]. The flexibility of such a powerful data-driven approach in comparison with classical speech concatenation synthesizers has also brought significant progress in the area of multispeaker TTS and speaker adaptation [53–55] as well the ability to conform to a particular speech style or emotion [56]. This is particularly relevant as it coincides with the emergence of applications

such as smart environments, virtual assistants, and intelligent robots, demanding high-quality speech synthesis in different voices and different styles and conveying different emotional states of the perceived speaker [57]. Other language technology tasks have also been successfully addressed by neural networks, such as question answering [58], text classification [59, 60], machine translation [61, 62], and sentiment analysis [63]. Neural networks have also been used as a powerful linguistic tool, for modelling sentence syntax [64] or exploring particular linguistic phenomena such as establishing word representations in vector spaces [65]. However, rather than providing a decomposition of the problem and a clear analytical insight into it, neural networks provide an alternative, data-driven point of view, and thus cannot be considered a classical tool of theoretical linguistics. On the other hand, their performance in solving these problems justly makes neural networks state of the art in the development of speech technology.

3. Progress in Speech Recognition and Synthesis, as well as Dialogue Systems

Apart from automatic speech recognition (ASR) and text-to-speech synthesis (TTS), a human-machine speech dialogue system also includes a dialogue management module with corresponding dialogue strategies and language technologies for spoken language understanding (SLU) and spoken language generation (SLG), as illustrated in Figure 4.

This section presents some achievements in the field of speech technologies such as ASR and TTS. They have been developed with an effort to combine interdisciplinary knowledge from different areas such as linguistics, acoustics, computer science, and mathematics. Signal processing engineers usually have integrating roles among linguists from one side and mathematicians from the other side.

3.1. Progress of Automatic Speech Recognition Systems. Research and development of ASR systems began in the 1950s in Bell Labs, with simple digit recognition systems, and since then the recognition tasks have become more complex—from the recognition of isolated digits, then isolated words, then continuously spoken words in a silent environment, up to the recognition of spontaneous speech in a noisy environment. Consequently, the complexity of the algorithms used also increased drastically. A brief review of historical development of ASR can be found in [66]. There were three important moments in the development of ASR systems: introduction of mel-frequency cepstral coefficients [67], introduction of statistical methods (hidden Markov models (HMM) with Gaussian mixture models (GMM)) [68], and introduction of deep neural networks (DNN) [69]. This development was also supported by the technological development in the computer industry as well as the increase in the amount of data available for training these systems.

The domination of DNNs in ASR started with [40], which showed that feedforward DNN outperforms GMM in the task of estimation of context-dependent HMM state emitting probabilities. For a small database, such as English

Figure 4: Components of a human-machine speech dialogue system.

Broadcast News (about 30 h of training data), the difference in word error rates (WER) was not significant, but for the Switchboard database, which is bigger (about 300 h of training data), the difference became substantial. Further improvement of DNN was based on better optimization, new activation functions, new network architectures, new speech preprocessing methods, and leveraging multiple languages and dialects [70]. One of the important findings was that layer-by-layer pretraining using restricted Boltzmann machines (RBM) is not obligatory and that back-propagation algorithm is sufficient for training in case of a large quantity of available training data as well as a large number of units in the hidden layers. Additionally, LeCun et al. showed in [71] that in case of sufficiently wide DNN (large number of units in a layer), there is no problem with the local minima and that the values of local minima are very close. The next big step was a complete elimination of HMM from the model. Graves and Jaitly in [72] reported a speech recognition system that directly transcribes audio data with text, without requiring an intermediate phonetic representation. The system is based on a combination of the deep bidirectional long-short term memory (LSTM) recurrent neural network architecture and the connectionist temporal classification (CTC) objective function. Such a direct mapping of an audio signal into a grapheme sequence allows easy application of the system on new languages such as Serbian [73]. Inspired by CTC, Povey at al. in [74] developed lattice-free maximum mutual information using phone n-gram language model starting from randomly initialized neural networks. This method was also successfully applied to Serbian [75]; i.e., the relative reduction of WER was about 25% with respect to the best previous system.

3.2. Progress of Speech Emotion Recognition. Since humans are not always rational and logical beings—emotions play very important aspects in acceptance of new products and technologies [76]. The earliest attempts to recognize speaker emotional state on the basis of voice characteristics date back to the 1980s [77]. The initial motive for this research direction was the adaptation of an ASR system to emotionally

stressed speech [78], but another motive appeared with the development of spoken dialogue systems, where it was useful to modify the dialogue strategy based on, e.g., user annoyance [79]. There are a number of emotions that can be easily represented in the activation-evaluation space [80], but classification of such a large number of emotions is difficult. Hence, classification space has been reduced to neutral and 6 archetypal emotions: anger, disgust, fear, joy, sadness, and surprise, which are the most obvious and distinct emotions [80]. It should be noted that archetypal emotions are not primary emotions in so-called "pallet theory," where each emotion can be represented as a combination of the primary ones.

One of the important steps in the design of a speech emotion recognition system is the extraction of features that efficiently discriminate between emotions independently of lexical content, speaker, and acoustic environment. It is well known that prosodic features are correlated with emotions [80], which is why standard features used in emotion recognition systems include pitch, energy, and phone duration [81]. These features are also related to the voice quality that is related to the emotions [82]. Emotions affect speech energy distribution across a wide range of frequencies, thus spectral features such as MFCCs, linear prediction cepstral coefficients, log frequency power coefficients, and formants were further proposed [83, 84]. Feature extraction procedure starts with the segmentation of the input signals into 20–30 ms long frames shifted by 10 ms, since speech is a nonstationary signal. After that, the features extracted from a chosen segment (corresponding to a particular phoneme, syllable, word, or sentence) are mapped into a single vector using functionals such as mean, second moment, contour slope, and range. Hereafter, features "condensed" in such a way represent the input of standard classification algorithms such as linear Bayes [85], k nearest neighbours [85, 86], support vector machines [87], GMM [86], and artificial neural networks [88]. On the other hand, such frame-based features can also be classified as a sequence using HMM [84] and RNN [89]. Besides low-level acoustic features, individual words or sequences of words obtained by ASR can also be used to perform emotion classification [90]. After a huge success of convolutional neural networks (CNN) in image classification, where lower layers of the network perform feature extraction, some research groups have tried to implement CNN in the same manner to obtain features [91, 92]. Since speech emotion recognition is a scarce data problem, one of the future trends will be the application of semisupervised learning [93]. More details about features, classification algorithms, and databases can be found in [94–97].

3.3. Progress in the Development of Text-to-Speech Synthesis. The very first "speech machines" were mechanical devices capable of producing single phonemes, and some of their combinations were introduced by Christian Kratzenstein and Wolfgang von Kempelen at the end of eighteenth century [98]. The VODER, presented in 1939 by Homer Dudley, can be considered as the first synthesizer which could generate whole sentences [99]. The first full TTS system for English was introduced in 1968 by Teranishi and Umeda [100]. It was an articulatory-based system which could perform text analysis and determine pauses in text using a sophisticated parser [101].

However, it was not until concatenative synthesizers were invented, that TTS gained widespread usage. The idea of concatenative TTS is to concatenate appropriate parts of a prerecorded database [102]. If the goal is domain-specific synthesis or a very large speech database is available, this approach can produce high-quality speech. However, there are audible glitches at the concatenation points if the appropriate units cannot be found in the database. The method is also extremely inflexible in terms of changing the speaking style or the voice of the speaker; it requires a whole new database to be recorded and annotated.

As applications of TTS became more popular and more widely used, the necessity of algorithms that could produce different voices and speaking styles from smaller databases has grown. From around 2000, statistical parametric speech synthesis, where the spectrum, fundamental frequency, and duration of speech were modelled by multispace probability distribution HMMs and multidimensional Gaussian distributions [103], became popular. The HMM synthesis enables transformation of speaker-independent system toward a target speaker using a very small amount of speech data [104], creating expressive voices [105], as well as multilingual voices [106]. However, this method never achieved the naturalness of concatenative TTS. One of the main problems is the signal smoothness caused by modelling similar contexts with the same Gaussian mixtures. Another big problem introduced with parametric methods is the usage of vocoder, a system that produces speech waveforms from predicted acoustic features. Vocoders, although significantly improved over the time, introduce some artefacts which affect the overall quality of generated speech. A detailed review of HMM-based speech synthesis can be found in [107].

The first attempts to use neural networks in speech synthesis can be found in [108]. However, the recent development of hardware, especially graphical processing units (GPUs), has popularized this approach and established its dominant status in the TTS research society. Deep neural networks (DNN) replaced decision trees and Gaussian mixture models in mapping input linguistic features to output acoustic features, enabling nonlinear mappings [109]. Although simple feedforward NN with several hidden layers and sigmoid or tangent hyperbolic activations are sufficient for the production of intelligible and natural sounding synthetic speech, introduction of LSTM (long short-term memory) units has brought further improvement into the quality of synthesized speech [110]. Some improvements were also reported by introducing generative adversarial networks [111] and stacked bottleneck features [112].

DNNs have not only just enabled generating synthetic speech of high quality but also introduced many possibilities for production of speech in different voices and speaking styles. A majority of methods for creating new DNN voice using limited amount of training data is based on usage of multispeaker models. In multispeaker modelling, a large

database consisting of multiple speakers is required. Each speaker is usually represented with less data than in case of single-speaker modelling. Due to a variety of contextual information and better network generalization, the quality of speech produced with multispeaker models is similar or even better than speech obtained with single speaker models. Speaker identity in multispeaker systems can be represented in several ways. One group of approaches is based on the use of a unique vector for each speaker. This vector can be represented as *i* vector [113] or just one-hot vector [54] and is used in training as extension of standard input or additional input to any of the hidden layers. Another group of methods for representing speaker identity is based on splitting network to speaker-specific and shared parts. In [53], separate output layers for each speaker have been introduced. In [114], even language-dependent parts of the network have been added, but this approach requires data from the same speaker in multiple languages. Creating a new voice, whose samples have not been seen in the training phase, in a multispeaker framework is based on adapting only the speaker-dependent part of the network [53], estimating the speaker-specific vector for the new speaker [55] or adjusting the parameters of neurons in starting models [113]. As opposed to the usage of multispeaker models as starting models for adaptation, in [115], adaptation starting from a single speaker model is investigated. It has been shown that only ~10 min of target speaker voice is required in order to produce synthetic speech in target speaker's voice reaching the quality of conventional methods built on several hours of speech database. The hypothesis was that the models of speakers A and B are more similar than a randomly initialized model and the model of speaker B, consequently requiring less data to train the model of speaker B starting from the model of speaker A than starting from a randomly initialized model.

Synthetic speech should convey not only just information but also paralinguistic information such as emotional state. There is also a need to support some task-specific speech styles such as news, commercials, storytelling, and warnings [116]. It has been shown that emotion, mood, and sentiment affect attention, memory, performance, judgement, and decision-making in humans [117], which supports the necessity of using different speaking styles in synthetic speech for many applications. Three different methods for style modelling are compared in [118]. The presented methods are based on ideas introduced in multispeaker modelling using input codes, network adaptation, and separated output layers. It has been shown that only ~5 min of speech per style is sufficient in order to produce speech of acceptable quality in a specific style. Using input codes for representing different styles is also presented in [119, 120]. There have also been attempts at style transplantation, i.e., producing speech in the voice of speaker A in style X without having any sentence from speaker A in style X in the training data, in which case the network is forced to learn the style X from other speakers in the training database [121, 122].

Although DNNs have shown to be extremely powerful and flexible, for a long time, one of their main disadvantages in speech synthesis has been their dependence on the usage of a vocoder. For the first time in 2016, raw audio samples were directly predicted by DNN using WaveNet architecture [51]. This model is fully probabilistic and autoregressive, with the predictive distribution for each audio sample conditioned on all previous ones. When conditioned on linguistic features derived from text and speaker identity, it can be used as TTS and it significantly outperforms all other TTS systems. The main drawbacks of this system are its need for extremely large databases and extreme computational power, although the synthesis has since been accelerated by the introduction of approaches such as Parallel WaveNet [123]. A similar model called DeepVoice was introduced in 2017 [124]. In DeepVoice, every part of TTS pipeline is replaced by a corresponding DNN. Its main drawback is the fact that all components of TTS system are trained independently, and it leads to a cumulative error in synthesized speech in the end.

As opposed to WaveNet and DeepVoice systems, which use lexical features as inputs, there are systems which use raw orthographic text as input, such as Tacotron [52], Tacotron 2 [125], and Deep Voice 3 [126]. Tacotron outputs spectrograms that are transformed to speech samples using Griffin–Lim algorithm, which also introduces artefacts in generated speech. On the other hand, the Tacotron 2 system-generated spectrograms are used for conditioning standard WaveNet architecture, which generates speech samples. DeepVoice 3 architecture can output spectrograms or other features which can be used as input to some waveform synthesis models. Adaptation to new speakers has also been investigated in end-to-end systems [127, 128] as well as synthesis in different styles [129, 130].

The main advantage of an integrated end-to-end TTS system is that requires minimal human effort since there is no need to label input data. Since in end-to-end systems, direct sample values are often predicted [29], the usage of 16 bit samples would make the prediction complicated and some type of quantization is performed. For this reason, improved coding and compression algorithms are important for TTS.

3.4. Dialogue Systems. Automatic speech recognition and speech synthesis are technologies with a long history. During the last five decades, a wide spectrum of algorithms shaped our knowledge within the speech technology field. With the recent advances in the world of deep learning and artificial neural networks, we are able to imitate to some extent the human auditory system sensitivity, recognition accuracy, human voice intelligibility and naturalness, emotions in synthesized speech, etc. As the result, machines are able to identify particular speakers, recognize human words in a noisy environment or to perform large-vocabulary continuous speech recognition with high accuracy. Furthermore, with a small amount of speech data from target speakers, they synthesize high quality speech, good enough to become a threat to automatic speaker verification systems.

One question that naturally arises is whether this is enough to achieve intelligent-like, natural and long-term human-machine speech interaction. Unfortunately, it is not.

Speech recognition and synthesis are only two of the six modules in a typical architecture of a speech dialogue system, depicted in Figure 4, and each of the six modules corresponds to certain cognitive aspects underlying the human language processing system. As a result, dialogue management becomes a complex structure that involves cooperation of several, quite different functional units. Leaving aside the division of dialogue systems into task-oriented and non-task-oriented, we must emphasize that the biggest challenge scientists face during the research in this area is to emulate human ability to understand meaning and conduct a conversation that is forward-looking, informative, and coherent. Regarding the dialogue initiatives, machines are successful in handling conversations that are system initiative (or single initiative). In such cases, the system completely controls the conversation and maintains the processes of speech recognition, meaning extraction, answer generation, and speech production. However, as it is known, natural dialogue is not deterministic and real improvements in human-machine speech interaction can be achieved only if adaptive behaviour with respect to the intention, the current context, and history of interaction are provided.

As a traditional paradigm shift, recent works in this area have addressed a series of data-driven, end-to-end trainable, non-goal-driven systems based on generative probabilistic models [131]. As such, these models can be viewed as artificial cognitive systems, aimed at grouping and carrying out traditional dialogue management tasks: language understanding, reasoning, decision-making, and natural language generation. They are corpus-based, data-driven dialogue systems, based on machine learning algorithms using corpora created from real word data. The statistics observed in dialogue corpora is the main knowledge for the optimization of parameters and variables.

It is worth pointing out that, besides the importance of domain knowledge, linguistic context has the crucial impact for active and engaging conversation. However, one of the main drawbacks of these approaches is related to sparsity issues that can be expected during integration of contextual information into statistical models. In the work of Sordoni et al. [132], the neural network architecture is used to address this problem, allowing the system to take into account the previous dialogue utterances. While modelling contextual information, the authors identify models for three linguistic entities in a conversation: the context (c), the message (m), and response (r). On that basis, they suggest three context-based generation models to estimate a response $r = r_1, \ldots, r_{\mathrm{T}}$ as follows:

$$p(r \mid c, m) = \prod_{t=1}^{\mathrm{T}} p(r_t \mid r_1, \ldots, r_{t-1}, c, m). \quad (1)$$

This work extends the recurrent neural network language models (RNNLM) as a generative model of sentences [133]. As the basic principle in this neural network model, input vector, representing the current word at time instant t, is concatenated to the output from neurons in network context layer at time $t-1$. In order to capture long-span dependencies together with semantic and syntactic similarities, the authors select word embedding as a continuous representation of words and phrases. Similar approaches already advanced classical language modelling, based on traditional n-gram language models [134, 135].

In line with this, many researches are trying to take advantage of combining neural network and end-to-end training possibilities with the large amount of freely available text or audio material from social media, movie scripts, etc. [136]. Serban et al. [137], for example, demonstrated introduction of latent variables to hierarchical recurrent encoder-decoder architecture. The research presented in [138] extended the hierarchical structure with the attention mechanism (word level attention and utterance level attention), taking into account that words and utterances in the context are differentially important.

Although end-to-end, statistical models have drawn most of the recent research on dialogue systems, many problems remain unresolved [131]. Neural network-based models are capable of handling large amount of data, but still it is hard to design an intelligent system based on imitating responses (especially if we take into account that the dialogue data for a specific domain are quite limited). Hence, to reduce these limitations, Mišković et al. [36] proposed a different, representational approach. This work extends the focus tree model, a cognitively inspired computational model of working memory that allows for adaptive dialogue management in human-machine interaction. The research not only is focused on improvements of speech recognition module but also points to possible new architectural aspect of dialogue systems. Following the assumption of the hierarchical and associative nature of human memory system and facts that the processing of the user's dialogue acts in human-machine interaction is always context-dependent, this model enables, to some extent, understanding of language and real word data.

4. Progress in Speech Signal Compression, Coding, and Transmission

In general, speech coders can be classified into three major categories depending on the applied coding technique: waveform coders, parametric coders, and hybrid coders. The primary idea behind a waveform coder design is to preserve the shape of a speech signal waveform, thus encoding information about the original time-domain waveform [4–6, 14, 139]. Such coders are broadly used in embedded applications due to several reasons: low cost of manufacturing, low computational resource usage, and high speech quality [4–7, 14, 139]. The simplest and most well-known type of waveform coders is pulse code modulation (PCM) coder, which is considered as a standard in digital telephony. One of the key advantages of PCM coders is that they are instantaneous, implying a coding delay of no more than one sample period [4]. Unlike waveform coders, which tend to reconstruct the original shape of the speech signal in time-domain, parametric coders reconstruct the speech signal from certain parameters that model the source signal, making no attempt to preserve the original shape of the waveform [4–7, 14, 139]. Due to this limitation, parametric coders are

more signal dependent and less versatile. Additionally, compared to waveform coders, they provide a lower quality of speech signals. In parametric coders, human speech production mechanism is modelled with a time-varying filter, having coefficients commonly determined by the linear prediction analysis procedure. In the end, hybrid coders represent a class of coders, which combine features of both previously described classes of coders, namely, hybrid coders tend to preserve the shape of the signal in time domain and also exploit perceptive characteristics, that is, parametric approach [4–6, 14, 140]. Performance comparison of these three classes of coders is presented in Figure 5, where mean opinion score (MOS) is used as one of the standard subjective measures of reconstructed speech signal quality [4].

From Figure 5, one can conclude that waveform coders provide excellent quality of reconstructed speech signal and that they represent the best choice at bit rates higher than 16 kbits/s, whereas parametric coders cannot provide high quality regardless of the bit rate. On the other hand, parametric coders provide much better quality than waveform coders at low bit rates. Finally, hybrid coders are most suitable at medium bit rates. As for the purposes of speech synthesis and automatic speech recognition, the highest possible quality of reconstructed signal is desirable and waveform coders are usually considered as an adequate choice. Thus, what follows is focused on PCM and adaptive PCM (ADPCM) coding techniques.

Figure 5: Speech signal quality according to MOS versus bit rate for various speech signal coding techniques.

4.1. Adaptive PCM. Speech signal can be considered as a nonstationary process, whose average power significantly fluctuates in time domain, resulting in a wide dynamic range [4]. However, speech signal can be considered as almost stationary in a short period of time (up to 30 ms). This means that speech signal has a highly predictable characteristics during short periods of time, which is suitable for utilizing adaptive quantization [4, 8–10, 141–147]. Commonly, adaptive quantization is frame-based, where frames are formed by dividing an input speech signal into sets of samples.

There are two fundamentally distinct categories of adaptive quantization techniques: forward and backward adaptive quantization techniques [141]. Forward adaptive techniques require transmission of additional information about the estimated gain, which is used for adaptation. Moreover, forward adaptive techniques require a longer processing delay than backward adaptive techniques as samples within a frame have to be stored in a buffer, in order to estimate predictable characteristics of every frame. When the gain is estimated and the quantizer is adapted, samples can be quantized and further transmitted to the decoder along with the quantized gain.

A general forward adaptive PCM model is presented in Figure 6 [8, 10]. The encoder is formed of two parts: a fixed (nonadaptive) part, consisting of a fixed quantizer Q_f, and an adaptive part, consisting of a buffer, a gain estimator, one divider, and a fixed gain quantizer Q_g. If Q_f is a piecewise linear μ-law quantizer designed for 8 bit/sample and $\mu = 255$, the general forward adaptation model becomes a forward adaptive PCM model defined by G.711 standard [148].

Unlike forward adaptation, backward adaptation does not estimate characteristics of samples in a frame while encoding, which means that there is not additional information that has to be transmitted [149]. In fact, gain estimation is performed at the receiver after decoding, considering previously quantized samples. The simplest backward adaptive quantization model is based on uniform quantization with one codeword memory exploited for gain estimation and it is commonly referred as Jayant's model [4].

Advanced backward adaptive models commonly incorporate more sophisticated gain estimation methods, or variance ($\widehat{\sigma}_y^2(n)$) estimation methods, which, for quantization of a current sample $x(n)$, typically exploit a larger number of previously decoded samples $y(n-i)$ [4]:

$$\widehat{\sigma}_y^2(n) = \frac{1-\alpha}{1+\alpha} \sum_{i=1}^{+\infty} \alpha^{i-1} y^2(n-i), \tag{2}$$

where α is a weighting parameter, which can take values $0 < \alpha < 1$. Parameter α defines a learning period, that is, a time required for variance estimation [4]:

$$L = \frac{1+\alpha}{1-\alpha}. \tag{3}$$

Equation (2) can be written in the following recursive form:

$$\widehat{\sigma}_y^2(n) = \frac{1-\alpha}{1+\alpha} y^2(n-1) + \alpha \widehat{\sigma}_y^2(n-1), \tag{4}$$

which is straightforwardly used in the simplest mathematical model of Jayant's backward quantizer with one codeword memory. One of the realizations of backward adaptive PCM with one codeword memory that incorporates a widely used companding quantization model is shown in Figure 7, where $M(n-1)$ denotes a step size multiplier, used for adaptation, and $c(x)$ and $c^{-1}(x)$ are a compressor function and an expandor function, respectively.

Figure 6: Forward adaptive PCM: (a) encoder; (b) decoder.

Figure 7: One of the realizations of backward adaptive PCM with one codeword memory: (a) encoder; (b) decoder.

4.2. Dual-Mode Quantization. Dual-mode and adaptive dual-mode quantizers belong to a relatively new class of quantizers whose design is based on multiparameter adaptation, such as variance and maximum amplitude [8, 9, 11]. Depending on their purpose and application, they can perform quantizer adaptation according to the frame variance and to the frame maximum amplitude x_{max} and also according to the subframe maximum amplitude. By utilizing two quantizers, which compose the dual-mode system, and by applying switched technique, it is possible to achieve a better quality of the quantized signal, or a higher compression, compared to the common single-mode quantizers. In Figure 8, a dual-mode quantization scheme is shown, where Encoder 1 and Decoder 1 are related to the quantizer applied for processing signals having restricted amplitude range, whereas Encoder 2 and Decoder 2 are related to the quantizer applied for processing the signals having unrestricted amplitude range [8, 9, 11]. The switched process is frame-based, and it is performed so that the restricted quantizer is used in the case if all samples within a frame belong to the restricted quantizer's support region, while the unrestricted quantizer is used otherwise [8, 9, 11]. The main idea behind such quantization model is to enable a more preferable selection of the restricted quantizer, with a narrower support region, than the unrestricted one, since, in such a manner, an increase of the signal to quantization noise ratio can be provided.

Considering that speech signal can be described using Gaussian probability density function (PDF) or Laplacian PDF, which is heavy-tailed, it is expected that only small percentage of speech frames will have some samples of large values. However, this also depends on the size of a frame. Consequently, the support region threshold values for both quantizers should be chosen so that the restricted quantizer usage should be dominant, but taking also into account the frame size and the whole input signal dynamics in order to achieve a minimum of the total distortion introduced in the quantization process [8, 9, 11].

4.3. Differential Pulse Code Modulation. Differential pulse code modulation (DPCM) represents a simple but high-quality speech signal coding scheme for middle bit rates. It initially exploited uniform quantization and the first-order prediction [150, 151]. As it was already discussed, speech signal has highly predictable characteristics within a frame, which is exploited to reduce the dynamic range of amplitudes

Figure 8: Dual mode quantization scheme: (a) encoder; (b) decoder.

Figure 9: DPCM: (a) encoder; (b) decoder.

for quantization in the DPCM scheme [4, 12, 13, 152, 153]. In particular, DPCM predicts the next sample amplitude value and encodes the difference between the predicted value and the value of the current input signal amplitude. Due to the high correlation, these differences have much smaller values compared to ones of the input signal samples, so that the dynamic range of amplitudes is significantly reduced before quantization. Accordingly, with a suitable design of a DPCM system, a certain distortion may be provided at lower bit rates compared to the PCM system. In other words, a worthy compression may be achieved with the DPCM system compared to the PCM.

More sophisticated solutions may incorporate prediction of a higher order or other kinds of quantization models, such as a gain-adaptive quantization model [154]. Figure 9 shows a DPCM scheme with incorporated simple first-order predictor and forward gain-adaptive quantizer based on optimal companding model [151].

In the DPCM system given in Figure 9, the reconstructed speech signal $\hat{x}$ is determined by

$$\hat{x}[n] = a \cdot \hat{x}[n-1] + y^a[n], \quad (5)$$

where y^a denotes the output of the adaptive quantizer, whereas n denotes the n-th sample of the signal. It can be noted that the value of parameter a depends on the nature of the input signal. If an input signal is highly correlated, it is preferred to use values close to 1 (e.g., $a = 0.8$), whereas values close to zero are preferred for lowly correlated signals (e.g., $a = 0.3$). However, the choice of parameter a is not an easy task even if adaptation is applied. The determination of linear predictor coefficients can be done using methods that are based on statistical learning such as least mean squares (LMS) estimation method [155]. LMS search algorithm reduces distortion by adapting coefficients for each input sample, and its main features, which attract researches, are low computational complexity, proof of convergence in stationary environment, unbiased convergence in the mean to the Wiener solution, and stable behaviour when implemented with finite-precision arithmetic [156]. Moreover, coefficients of linear predictor as well as determination of other important parameters for quantizer design may be determined by exploiting artificial neural networks or regression methods.

5. Conclusions

This review article has provided an overview in the recent development of speech technologies and other scientific areas related to them, mostly due to the development of the new machine learning paradigm, which has had a tremendous impact in this domain. Apart from natural speech production and speech perception, understanding of cognitive aspects of speech communication is very important for future HCI systems including both spoken language understanding and generation as language technologies. The machine learning paradigm has had a great impact on automatic speech recognition (ASR) and text-to-speech synthesis (TTS) as basic speech technologies. It is expected that ASR systems based on deep learning and adaptive algorithms in the near future will be able to recognize spontaneous speech in complex acoustic environments, with the accuracy that will surpass the corresponding ability of humans. Synthetic speech has already reached such quality that is hard or impossible to differentiate from human speech. With flexibility of changing speaker and style, HCI is becoming as pleasant and natural as human-human interaction. Unsupervised and reinforcement-based machine learning algorithms will also develop further, which will, in turn, bring about progress in areas where large data sets are not available, as is the case in speech analysis for speech recognition and synthesis for under-resourced languages. A short overview of speech coding techniques and of current progress in adaptive scalar quantization has been presented as the quality of digitized and compressed speech signal is important for accurate automatic speech signal detection and synthesis. Although these techniques can be designed to be robust in a wide dynamic range of speech signal variations, or to be frame-adaptive, one can anticipate that machine learning tools of increasing popularity will lead to novel solutions, which will improve performances of various systems by adapting predictive coefficients. To conclude, we are witnessing an increasingly fast progress in the field of speech signal processing due to machine learning paradigms, and it appears very hard to predict what they will bring about next and how soon that can be expected.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work described in this paper was supported in part by the Ministry of Education, Science and Technological Development of the Republic of Serbia, through the project "Development of Dialogue Systems for Serbian and Other South Slavic Languages" (TR32035).

References

[1] T. S. Kuhn, *The Structure of Scientific Revolutions-50th Anniversary Edition*, vol. 3, 4th edition, 2012.

[2] R. K. Moore, "Cognitive informatics: the future of spoken language processing?," in *Proceedings of the 10th International Conference on Speech and Computer (SPECOM)*, Patras, Greece, October 2005.

[3] J. D. Paul, "Re-creating the sigsaly quantizer: this 1943 analog-to-digital converter gave the allies an unbreakable scrambler-(resources)," *IEEE Spectrum*, vol. 56, no. 2, pp. 16-17, 2019.

[4] N. S. Jayant and P. Noll, "Digital coding of waveforms. Principles and applications to speech and video," *Signal Processing*, vol. 9, no. 2, pp. 139-140, 1985.

[5] W. C. Chu, *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*, John Wiley & Sons, Hoboken, NJ, USA, 2003.

[6] L. Hanzo, F. C. A. Somerville, and J. P. Woodard, *Voice and Audio Compression for Wireless Communications*, John Wiley & Sons Ltd., Chichester, England, 2nd edition, 2007.

[7] A. M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communication Systems*, John Wiley & Sons Ltd., Chichester, UK, 2nd edition, 2004.

[8] Z. H. Peric and J. Nikolic, "An adaptive waveform coding algorithm and its application in speech coding," *Digital Signal Processing*, vol. 22, no. 1, pp. 199–209, 2012.

[9] J. Nikolic, Z. H. Peric, and A. Z. Jovanovic, "Two forward adaptive dual-mode companding scalar quantizers for Gaussian source," *Signal Processing*, vol. 120, pp. 129–140, 2016.

[10] J. Nikolic and Z. H. Peric, "Lloyd-Max's algorithm implementation in speech coding algorithm based on forward adaptive technique," *Informatica (Lithuanian Academy of Sciences)*, vol. 19, no. 2, pp. 255–270, 2008.

[11] Z. Peric and J. Nikolic, "High-quality Laplacian source quantisation using a combination of restricted and unrestricted logarithmic quantisers," *IET Signal Processing*, vol. 6, no. 7, pp. 633–640, 2012.

[12] O. Ordentlich and U. Erez, "Performance analysis and optimal filter design for sigma-delta modulation via duality

with DPCM," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 1153–1164, 2019.

[13] R. C. Farias and J. M. Brossier, "Adaptive quantizers for estimation," *Signal Processing*, vol. 93, no. 11, pp. 3076–3087, 2013.

[14] I. V. McLoughlin, *Speech and Audio Processing: a MATLAB-Based Approach*, Cambridge University Press, Cambridge, UK, 2016.

[15] M. Cernak, A. Asaei, and A. Hyafil, "Cognitive speech coding: examining the impact of cognitive speech processing on speech compression," *IEEE Signal Processing Magazine*, vol. 35, no. 3, pp. 97–109, 2018.

[16] Neurophysiology of Speech Communication, http://naplab.ee.columbia.edu/speech-neurophysiology.html.

[17] F. A. Everest and K. C. Pohlmann, *Master Handbook of Acoustics*, McGraw-Hill, New York, NY, USA, 5th edition, 2009, http://www.roletech.net/books/HandbookAcoustics.pdf.

[18] D. O'Shaughnessy, "Linear predictive coding," *IEEE Potentials*, vol. 7, no. 1, pp. 29–32, 1988.

[19] L. R. Rabiner and R. W. Schafer, "Introduction to digital speech processing," *Foundations and Trends in Signal Processing*, vol. 1, no. 1-2, pp. 1–194, 2007.

[20] X. Huang, A. Acero, H. W. Hon, and R. Reddy, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[21] S. Jovičić, *Speech Communication—Physiology of Psychoacoustics and Perception*, Izdavačko Preduzeće Nauka, Belgrade, Serbia, 1999, in Serbian.

[22] T. D. Rossing, *Springer Handbook of Acoustics*, Springer Nature, New York, NY, USA, 2007.

[23] J. Blauert, *Communication Acoustics*, Springer, Berlin, Germany, 2005.

[24] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: a tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[25] C. Brokish and M. Lewis, *A-Law and Mu-Law Companding Implementations Using the TMS320C54x. Application Note SPRA163A*, Texas Instruments, Dallas, TX, USA, 1997, http://www.ti.com/lit/an/spra163a/spra163a.pdf.

[26] D. M. W. Powers, "Applications and explanations of Zipf's law," in *Proceedings of the Joint Conference on New Methods in Language Processing and Computational Natural Language Learning, NeMLaP/CoNLL*, pp. 151–160, Sydney, Australia, January 1998.

[27] X. D. Huang, M. Jack, and M. Ariki, *Hidden Markov Models for Speech Recognition*, Columbia University Press, New York, NY, USA, 1990.

[28] D. M. Howard and J. A. S. Angus, *Acoustics and Psychoacoustics*, Elsevier Ltd., Oxford, UK, 4th edition, 2009, https://www.sciencedirect.com/book/9780240521756/acoustics-and-psychoacoustics.

[29] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 52–59, 1986.

[30] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*, MIT Press, Cambridge, MA, USA, 1994.

[31] M. Vorländer, *Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality*, Springer, Berlin, Germany, 2008.

[32] Y. Ando, *Auditory and Visual Sensations*, Springer, New York, NY, USA, 2010, https://link.springer.com/book/10.1007/b13253.

[33] T. Nagamine and N. Mesgarani, "Understanding the representation and computation of multilayer perceptrons: a case study in speech recognition," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 2564–2573, Sydney, Australia, August 2017.

[34] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 787–796, 2018.

[35] M. Gnjatović and V. Delić, "Cognitively-inspired representational approach to meaning in machine dialogue," *Knowledge Based Systems*, vol. 71, no. 1, pp. 25–33, 2014.

[36] D. Mišković, M. Gnjatović, P. Štrbac, B. Trenkić, N. Jakovljević, and V. Delić, "Hybrid methodological approach to context-dependent speech recognition," *International Journal of Advanced Robotic Systems*, vol. 14, no. 1, 2017.

[37] S. McAdams and C. Drake, "Auditory perception and cognition," in *Stevens' Handbook of Experimental Psychology*, H. Pashler, Ed., pp. 397–452, John Wiley & Sons, New York, NY, USA, 2002.

[38] T. Nagamine, M. L. Seltzer, and N. Mesgarani, "On the role of nonlinear transformations in deep neural network acoustic models," in *Proceedings of Interspeech 2016 17th Annual Conference of the International Speech Communication Association*, pp. 803–807, San Francisco, CA, USA, September 2016.

[39] W. Zhang, H. Li, M. Yang, and N. Mesgarani, "Synaptic depression in deep neural networks for speech processing," in *Proceedings 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5865–5869, Shanghai, China, March 2016.

[40] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[41] M. Sundermeyer, H. Ney, and R. Schluter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517–529, 2015.

[42] K. Irie, Z. Tüske, T. Alkhouli, R. Schlüter, and H. Ney, "LSTM, GRU, highway and a bit of attention: an empirical overview for language modeling in speech recognition," in *Proceedings of Interspeech 2016, 17th Annual Conference of the International Speech Communication Association*, pp. 3519–3523, San Francisco, CA, USA, September 2016.

[43] W. Xiong, J. Droppo, X. Huang et al., "Achieving human parity in conversational speech recognition," https://arxiv.org/pdf/1610.05256.pdf.

[44] A. Hannun, C. Case, J. Casper et al., "Deep Speech: scaling up end-to-end speech recognition," http://arxiv.org/abs/1412.5567.

[45] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: a neural network for large vocabulary conversational speech recognition," in *Proceedings 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, Shanghai, China, March 2016.

[46] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, 2015.

[47] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7962–7966, Vancouver, BC, Canada, May 2013.

[48] H. Zen and A. W. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3844–3848, Florence, Italy, May 2014.

[49] Y. Fan, Y. Qian, F. L. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proceedings of Interspeech 2014, 15th Annual Conference of the International Speech Communication Association*, pp. 1964–1968, Singapore, September 2014.

[50] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "Prosody contour prediction with long short-term memory, bi-directional deep recurrent neural networks," in *Proceedings of Interspeech 2014, 15th Annual Conference of the International Speech Communication Association*, pp. 2268–2272, Singapore, September 2014.

[51] A. van den Oord, S. Dieleman, H. Zen et al., "WaveNet: a generative model for raw audio," https://arxiv.org/abs/1609.03499.

[52] Y. Wang, R. J. Skerry-Ryan, D. Stanton et al., "Tacotron: towards end-to-end speech synthesis," in *Proceedings of the Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, pp. 4006–4010, Stockholm, Sweden, August 2017.

[53] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4475–4479, South Brisbane, Australia, April 2015.

[54] N. Hojo, Y. Ijima, and H. Mizuno, "An investigation of DNN-based speech synthesis using speaker codes," in *Proceedings of the Interspeech 2016, 17th Annual Conference of the International Speech Communication Association*, pp. 2278–2282, San Francisco, CA, USA, September 2016.

[55] H. T. Luong, S. Takaki, G. E. Henter, and J. Yamagishi, "Adapting and controlling DNN-based speech synthesis using input codes," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4905–4909, New Orleans, LA, USA, March 2017.

[56] Y. Wang, D. Stanton, Y. Zhang et al., "Style tokens: unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 5167–5176, Stockholm, Sweden, June 2018.

[57] E. Eide, A. Aaron, R. Bakis, W. Hamza, M. Picheny, and J. Pitrelli, "A corpus-based approach to expressive speech synthesis," in *Proceedings of the Fifth ISCA ITRW on Speech Synthesis*, pp. 79–84, Pittsburgh, PA, USA, June 2004.

[58] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Learning to compose neural networks for question answering," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1545–1554, San Diego, CA, USA, June 2016.

[59] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, October 2014.

[60] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proceedings of the Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems*, pp. 649–657, Montreal, Canada, December 2015.

[61] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pp. 3104–3112, Montreal, Canada, December 2014.

[62] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA, May 2015.

[63] C. N. D. Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 69–78, Dublin, Ireland, August 2014.

[64] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 655–665, Baltimore, MD, USA, June 2014.

[65] T. Mikolov, W. t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 746–751, Atlanta, GA, USA, June 2013.

[66] B. Gold, N. Morgan, and D. Ellis, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, John Wiley & Sons, Hoboken, NJ, USA, 2nd edition, 2011.

[67] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," in *Pattern Recognition and Artificial Intelligence*, C. H. Chen, Ed., vol. 116, pp. 374–388, Academic Press, New York, NY, USA, 1976.

[68] F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[69] A. R. Mohamed, G. Dahl, and G. E. Hinton, "Deep belief networks for phone recognition," in *Proceedings of the NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, Whistler, Canada, December 2009.

[70] L. Deng, G. E. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: an overview," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8599–8603, Vancouver, Canada, May 2013.

[71] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[72] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning*, pp. 1764–1772, Beijing, China, June 2014.

[73] B. M. Popovic, E. Pakoci, and D. Pekar, "End-to-end large vocabulary speech recognition for the Serbian language," in *Proceedings of the 19th International Conference Speech and Computer (SPECOM)*, pp. 343–352, Hatfield, UK, September 2017.

[74] D. Povey, V. Peddinti, D. Galvez et al., "Purely sequence-trained neural networks for ASR based on lattice-free MMI,"

in *Proceedings of the Interspeech 2016 17th Annual Conference of the International Speech Communication Association*, pp. 2751–2755, San Francisco, CA, USA, September 2016.

[75] E. Pakoci, B. Popović, and D. J. Pekar, "Improvements in Serbian speech recognition using sequence-trained deep neural networks," *SPIIRAS Proceedings*, vol. 3, no. 58, pp. 53–76, 2018.

[76] P. Zhang and N. Li, "The importance of affective quality," *Communications of the ACM*, vol. 48, no. 9, pp. 105–108, 2005.

[77] R. Bezooijen, *Characteristics and Recognizability of Vocal Expressions of Emotion*, Foris Publications Holland, Dordrecht, Netherlands, 1984.

[78] D. A. Cairns and J. H. L. Hansen, "Nonlinear analysis and classification of speech under stressed conditions," *Journal of Acoustical Society of America*, vol. 96, no. 6, pp. 3392–3400, 1994.

[79] J. Ang, R. Dhillon, A. Krupski, E. Shriberg, and A. Stolcke, "Prosody-based automatic detection of annoyance and frustration in human-computer dialog," in *Proceedings of the ICSLP 2002 7th International Conference on Spoken Language Processing*, pp. 2037–2040, Denver, CO, USA, September 2002.

[80] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis et al., "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[81] Y. Wang, S. Du, and Y. Zhan, "Adaptive and optimal classification of speech emotion recognition," in *Proceedings of the Fourth International Conference on Natural Computation ICNC*, vol. 5, pp. 407–411, Jinan, China, October 2008.

[82] C. Gobl and A. N. Chasaide, "The role of voice quality in communicating emotion, mood and attitude," *Speech Communication*, vol. 40, no. 1-2, pp. 189–212, 2003.

[83] G. Zhou, J. H. L. Hansen, and J. F. Kaiser, "Nonlinear feature based classification of speech under stress," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 201–216, 2001.

[84] T. L. Nwe, S. W. Foo, and L. C. De Silva, "Speech emotion recognition using hidden markov models," *Speech Communication*, vol. 41, no. 4, pp. 603–623, 2003.

[85] V. Delic, M. Bojanic, M. Gnjatovic, M. Secujski, and S. T. Jovicic, "Discrimination capability of prosodic and spectral features for emotional speech recognition," *Electronics and Electrical Engineering*, vol. 18, no. 9, pp. 51–54, 2012.

[86] R. B. Lanjewar, S. Mathurkar, and N. Patel, "Implementation and comparison of speech emotion recognition system using Gaussian mixture model (GMM) and K-nearest neighbor (K-NN) techniques," *Procedia Computer Science*, vol. 49, pp. 50–57, 2015.

[87] P. Shen, Z. Changjun, and X. Chen, "Automatic speech emotion recognition using support vector machine," in *Proceedings of the 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, vol. 2, pp. 621–625, Harbin, China, August 2011.

[88] K. Han, D. Yu, and I. Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," in *Proceedings of the Interspeech 2014 15th Annual Conference of the International Speech Communication Association*, pp. 223–227, Singapore, September 2014.

[89] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," in *Proceedings of the Interspeech 2015, 16th Annual Conference of the International Speech Communication Association*, pp. 1537–1540, Dresden, Germany, September 2015.

[90] C. M. Lee, S. S. Narayanan, and R. Pieraccini, "Combining acoustic and language information for emotion recognition," in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP2002)*, pp. 873–876, Denver, CO, USA, September 2002.

[91] H. M. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for speech emotion recognition," *Neural Networks*, vol. 92, pp. 60–68, 2017.

[92] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomedical Signal Processing and Control*, vol. 47, no. 1, pp. 312–323, 2019.

[93] J. Deng, X. Xu, Z. Zhang, S. Fruhholz, and B. Schuller, "Semisupervised autoencoders for speech emotion recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 31–43, 2018.

[94] D. Ververidis and C. Kotropoulos, "Emotional speech recognition: resources, features, and methods," *Speech Communication*, vol. 48, no. 9, pp. 1162–1181, 2006.

[95] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[96] M. Swain, A. Routray, and P. Kabisatpathy, "Databases, features and classifiers for speech emotion recognition: a review," *International Journal of Speech Technology*, vol. 21, no. 1, pp. 93–120, 2018.

[97] L. Chen, X. Mao, Y. Xue, and L. L. Cheng, "Speech emotion recognition: features and classification models," *Digital Signal Processing*, vol. 22, no. 6, pp. 1154–1160, 2012.

[98] M. R. Schroeder, "A brief history of synthetic speech," *Speech Communication*, vol. 13, no. 1-2, pp. 231–237, 1993.

[99] H. Dudley, "The carrier nature of speech," *Bell System Technical Journal*, vol. 19, no. 4, pp. 495–515, 1940.

[100] R. Teranishi and N. Umeda, "Use of pronouncing dictionary in speech synthesis experiments," *Reports of the Sixth International Congress on Acoustics*, vol. 2, pp. 155–158, 1968.

[101] D. H. Klatt, "Review of text-to-speech conversion for English," *Journal of Acoustical Society of America*, vol. 82, no. 3, pp. 737–793, 1987.

[102] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 373–376, Atlanta, GA, USA, May 1996.

[103] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proceedings of the Eurospeech 1999 6th European Conference on Speech Communication and Technology*, pp. 2347–2350, Budapest, Hungary, September 1999.

[104] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 66–83, 2009.

[105] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi, "Modeling of various speaking styles and emotions for HMM-based speech synthesis," in *Proceedings of the Eurospeech 2003 8th European Conference on Speech Communication and Technology*, pp. 2461–2464, Geneva, Switzerland, September 2003.

[106] Y. Qian, H. Liang, and F. K. Soong, "A cross-language state sharing and mapping approach to bilingual (Mandarin-

English) TTS," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1231–1239, 2009.

[107] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[108] T. Weijters and J. Thole, "Speech synthesis with artificial neural networks," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1764–1769, San Francisco, CA, USA, April 1993.

[109] Z. Wu, O. Watts, and S. King, "Merlin: an open source neural network speech synthesis system," in *Proceedings of the 9th ISCA Speech Synthesis Workshop*, pp. 202–207, Sunnyvale, CA, USA, July 2016.

[110] H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak, "Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices," in *Proceedings of the Interspeech 2016 17th Annual Conference of the International Speech Communication Association*, pp. 2273–2277, San Francisco, CA, USA, September 2016.

[111] Y. Saito, S. Takamichi, and H. Saruwatari, "Statistical parametric speech synthesis incorporating generative adversarial networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 84–96, 2018.

[112] Z. Wu and S. King, "Improving trajectory modelling for DNN-based speech synthesis by using stacked bottleneck features and minimum generation error training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1255–1265, 2016.

[113] Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King, "A study of speaker adaptation for DNN-based speech synthesis," in *Proceedings of the Interspeech 2015 16th Annual Conference of the International Speech Communication Association*, pp. 879–883, Dresden, Germany, September 2015.

[114] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Speaker and language factorization in DNN-based TTS synthesis," in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5540–5544, Shanghai, China, April 2016.

[115] T. Delic, S. Suzic, M. Secujski, and D. Pekar, "Rapid development of new TTS voices by neural network adaptation," in *Proceedings of the 2018 17th International Symposium Infoteh-Jahorina (Infoteh)*, pp. 1–6, Istočno Sarajevo, Bosnia and Herzegovina, March 2018.

[116] M. Abe, "Speaking styles: statistical analysis and synthesis by a text-to-speech system," in *Progress in Speech Synthesis*, J. P. H. Van Santen et al., Ed., pp. 495–510, Springer, New York, NY, USA, 1997, https://link.springer.com/chapter/10.1007/978-1-4612-1894-4_39.

[117] S. Brave and N. Clifford, "Emotion in human-computer interaction," in *Human-Computer Interaction Fundamentals*, A. Sears and J. A. Jacko, Eds., pp. 54–66, CRC Press, Boca Raton, FL, USA, 2009.

[118] S. Suzić, T. Delić, V. Jovanović et al., "A comparison of multi-style DNN-based TTS approaches using small datasets," in *Proceedings of the 13th International Scientific-Technical Conference on Electromechanics and Robotics Zavalishin's Readings*, vol. 161, pp. 1–6, St. Petersburg, Russia, May 2018.

[119] J. Lorenzo-Trueba, G. Eje Henter, S. Takaki, J. Yamagishi, Y. Morino, and Y. Ochiai, "Investigating different representations for modeling and controlling multiple emotions in DNN-based speech synthesis," *Speech Communication*, vol. 99, pp. 135–143, 2018.

[120] S. An, Z. Ling, and L. Dai, "Emotional statistical parametric speech synthesis using LSTM-RNNs," in *Proceedings of the 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1613–1616, Kuala Lumpur, Malaysia, December 2017.

[121] S. Suzić, T. Delić, D. Pekar, V. Delić, and M. Sečujski, "Style transplantation in neural network based speech synthesis," *Acta Polytechnica Hungarica*, 2019.

[122] K. Inoue, S. Hara, M. Abe, N. Hojo, and Y. Ijima, "An investigation to transplant emotional expressions in DNN-based TTS synthesis," in *Proceedings of the 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1253–1258, Kuala Lumpur, Malaysia, December 2017.

[123] A. van den Oord, Y. Li, I. Babuschkin et al., "Parallel WaveNet: fast high-fidelity speech synthesis," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 3915–3923, Stockholm, Sweden, June 2018.

[124] S. O. Arik, M. Chrzanowski, A. Coates et al., "Deep voice: real-time neural text-to-speech," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 195–204, Sydney, Australia, August 2017.

[125] J. Shen, R. Pang, R. J. Weiss et al., "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4779–4783, Calgary, Canada, September 2018.

[126] W. Ping, K. Peng, A. Gibiansky et al., "Deep voice 3: scaling text-to-speech with convolutional sequence learning," https://arxiv.org/abs/1710.07654.

[127] S. Ö. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018*, pp. 10040–10050, Montreal, Canada, December 2018.

[128] E. Nachmani, A. Polyak, Y. Taigman, and L. Wolf, "Fitting new speakers based on a short untranscribed sample," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 3680–3688, Stockholm, Sweden, June 2018.

[129] R. Skerry-Ryan, E. Battenberg, Y. Xiao et al., "Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron," in *Proceedings of the 35th International Conference on Machine Learning*, pp. 4693–4702, Stockholm, Sweden, July 2018.

[130] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Expressive speech synthesis via modeling expressions with variational autoencoder," in *Proceedings of the Interspeech 2018 19th Annual Conference of the International Speech Communication Association*, pp. 3067–3071, Hyderabad, India, September 2018.

[131] H. Chen, X. Liu, D. Yin, and J. Tang, "A survey on dialogue systems," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 2, pp. 25–35, 2017.

[132] A. Sordoni, M. Galley, M. Auli et al., "A neural network approach to context-sensitive generation of conversational responses," in *Proceedings of the 2015 Conference of the North American Chapter of the of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, pp. 196–205, Denver, CO, USA, May 2015.

[133] T. Mikolov, M. Karafiát, L. Burget, J. H. Černocký, and S. Khudanpur, "Recurrent neural network based language

model," in *Proceedings of the Interspeech 2011 12th Annual Conference of the International Speech Communication Association*, pp. 1045–1048, Florence, Italy, August 2011.

[134] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 1137–1155, 2003.

[135] J. Gao, X. He, W.-T. Yih, and L. Deng, "Learning continuous phrase representations for translation modeling," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 699–709, Baltimore, MD, USA, June 2014.

[136] I. V. Serban, R. Lowe, P. Henderson, L. Charlin, and J. Pineau, "A survey of available corpora for building data-driven dialogue systems," https://arxiv.org/pdf/1512.05742.pdf.

[137] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 3776–3783, Phoenix, AR, USA, February 2016.

[138] C. Xing, Y. Wu, W. Wu, Y. Huang, and M. Zhou, "Hierarchical recurrent attention network for response generation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 5610–5617, New Orleans, LA, USA, February 2018.

[139] P. Sinha, *Speech Processing in Embedded Systems*, Springer, Berlin, Germany, 2009.

[140] M. Hasegawa-Johnson and A. Alwan, "Speech coding: fundamentals and applications," in *Wiley Encyclopedia of Telecommunications*, J. G. Proakis, Ed., John Wiley & Sons, Hoboken, NJ, USA, 2003.

[141] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Springer Nature, New York, NY, USA, 1991.

[142] Z. H. Perić, J. R. Nikolić, A. V. Mosić, and M. D. Petković, "Design of fixed and adaptive companding quantizer with variable-length codeword for memoryless Gaussian source," *Informatica (Lithuanian Academy of Sciences)*, vol. 24, no. 1, pp. 71–86, 2013.

[143] J. Nikolic, Z. Peric, A. Jovanovic, and D. Antic, "Design of forward adaptive piecewise uniform scalar quantizer with optimized reproduction level distribution per segments," *Electronics and Electrical Engineering*, vol. 119, no. 3, pp. 19–22, 2012.

[144] J. Nikolić, Z. Perić, D. Antić, A. Jovanović, and D. Denić, "Low complex forward adaptive loss compression algorithm and its application in speech coding," *Journal of Electrical Engineering-Elektrotechnicky Casopis*, vol. 62, no. 1, pp. 19–24, 2011.

[145] Z. Perić, A. Jocić, J. Nikolić, L. Velimirović, and D. Denić, "Analysis of differential pulse code modulation with forward adaptive Lloyd-Max's quantizer for low bit-rate speech coding," *Revue Roumaine des Sciences Techniques. Ser. Electrotechnique et Energetique*, vol. 58, no. 4, pp. 424–434, 2013.

[146] Z. H. Perić, J. R. Nikolić, and A. V. Mosić, "Design of forward adaptive hybrid quantiser with golomb-rice code for compression of Gaussian source," *IET Communications*, vol. 8, no. 3, pp. 372–377, 2014.

[147] S. Tomic, Z. Peric, M. Tancic, and J. Nikolic, "Backward adaptive and quasi-logarithmic quantizer for sub-band coding of audio," *Information Technology and Control*, vol. 47, no. 1, pp. 131–139, 2018.

[148] Telkom Institute of Technology, *Recommendation G.711, Pulse Code Modulation (PCM) of Voice Frequencies*, Telkom Institute of Technology, Bandung, Indonesia, 1972, https://www.itu.int/rec/T-REC-G.711.

[149] A. Ortega and M. Vetterli, "Adaptive scalar quantization without side information," *IEEE Transactions on Image Processing*, vol. 6, no. 5, pp. 665–676, 1997.

[150] P. Cummiskey, N. S. Jayant, and J. L. Flanagan, "Adaptive quantization in differential PCM coding of speech," *Bell System Technical Journal*, vol. 52, no. 7, pp. 1105–1118, 1973.

[151] V. M. Despotovic, Z. H. Peric, L. Velimirovic, and V. D. Delic, "DPCM with forward gain-adaptive quantizer and simple switched predictor for high quality speech signals," *Advances in Electrical and Computer Engineering*, vol. 10, no. 4, pp. 95–98, 2010.

[152] S. S. Tomic, Z. H. Peric, and J. R. Nikolic, "An algorithm for simple differential speech coding based on backward adaptation technique," *Informatica*, vol. 29, no. 3, pp. 539–553, 2018.

[153] R. Pratheek and M. N. Suma, "Performance analysis of DPCM and ADPCM," in *Proceedings of the International Conference on Electronic Design and Signal Processing*, pp. 19–23, Manipal, India, December 2013.

[154] J. H. Chen and A. Gersho, "Gain-adaptive vector quantization with application to speech coding," *IEEE Transactions on Communications*, vol. 35, no. 9, pp. 918–930, 1987.

[155] S. O. Haykin, *Neural Networks and Learning Machines*, Prentice-Hall, New Jersey, NJ, USA, 3rd edition, 2009.

[156] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, Springer, Boston, MA, USA, 4th edition, 2013, https://www.springer.com/gp/book/9781461441052.

Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 9439407, 14 pages
https://doi.org/10.1155/2019/9439407

Research Article

Steady-State Motion Visual Evoked Potential (SSMVEP) Enhancement Method Based on Time-Frequency Image Fusion

Wenqiang Yan,[1,2] **Guanghua Xu**,[1,2] **Longting Chen**,[1,2] **and Xiaowei Zheng**[1,2]

[1]*School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China*
[2]*State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an, China*

Correspondence should be addressed to Guanghua Xu; ghxu@xjtu.edu.cn

Received 14 February 2019; Revised 1 April 2019; Accepted 7 April 2019; Published 21 May 2019

Guest Editor: Zoran Peric

Copyright © 2019 Wenqiang Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The steady-state motion visual evoked potential (SSMVEP) collected from the scalp suffers from strong noise and is contaminated by artifacts such as the electrooculogram (EOG) and the electromyogram (EMG). Spatial filtering methods can fuse the information of different brain regions, which is beneficial for the enhancement of the active components of the SSMVEP. Traditional spatial filtering methods fuse electroencephalogram (EEG) in the time domain. Based on the idea of image fusion, this study proposed an SSMVEP enhancement method based on time-frequency (T-F) image fusion. The purpose is to fuse SSMVEP in the T-F domain and improve the enhancement effect of the traditional spatial filtering method on SSMVEP active components. Firstly, two electrode signals were transformed from the time domain to the T-F domain via short-time Fourier transform (STFT). The transformed T-F signals can be regarded as T-F images. Then, two T-F images were decomposed via two-dimensional multiscale wavelet decomposition, and both the high-frequency coefficients and low-frequency coefficients of the wavelet were fused by specific fusion rules. The two images were fused into one image via two-dimensional wavelet reconstruction. The fused image was subjected to mean filtering, and finally, the fused time-domain signal was obtained by inverse STFT (ISTFT). The experimental results show that the proposed method has better enhancement effect on SSMVEP active components than the traditional spatial filtering methods. This study indicates that it is feasible to fuse SSMVEP in the T-F domain, which provides a new idea for SSMVEP analysis.

1. Introduction

To improve the comfort of light-flashing stimulation, we proposed a steady-state motion visual evoked potential (SSMVEP) method to replace light-flashing stimulation with motion stimulation in the previous study [1]. In this study, the SSMVEP signal was selected as a research object. The SSMVEP collected from the scalp will be contaminated by a variety of artifacts, such as the electrooculogram (EOG) and electromyogram (EMG). Consequently, the requirements for subsequent signal processing are high. In electroencephalogram (EEG) signal processing, using multichannel EEG is beneficial and effective [2, 3]. In the EEG literature, the method for linearly fusing the multilead signals into single-channel or multichannel signals is called spatial filtering. Spatial filtering combines the EEG information of different brain regions, which enhances the active components of the EEG [4, 5].

At present, the main methods used in EEG spatial filtering are average fusion, native fusion, bipolar fusion, Laplacian fusion, common average reference (CAR) fusion, canonical correlation analysis (CCA) fusion, minimum energy fusion, and maximum contrast fusion [4–6]. Average fusion is obtained by averaging all electrode signals, and a bipolar fusion signal is obtained by subtracting the two electrode signals [7, 8]. It is called native fusion if only one of the electrode signals is analyzed. Laplacian fusion is a centered electrode signal minus the sum of four electrodes at the same distance from their surroundings. CAR fusion is another commonly used EEG spatial filtering method that enhances the signal-to-noise ratio (SNR) of the selected electrode signal by subtracting the mean of all electrode

signals from the selected electrode signal. Dennis et al. studied the enhancement effects of standard ear reference, CAR fusion, and Laplacian fusion on the active components of EEG. The results show that the Laplacian fusion achieved the best fusion effects on EEG active components [9]. Friman et al. proposed the minimum energy fusion and maximum contrast fusion methods [5]. The minimum energy fusion was achieved by minimizing the noise energy, and the maximum contrast fusion was achieved by maximizing the EEG SNR. In the study, Friman et al. compared the effects of average fusion, native fusion, bipolar fusion, Laplacian fusion, minimum energy fusion, and maximum contrast fusion on steady-state visual evoked potential (SSVEP) signals. Among these, the enhancement effect of average fusion was worst, and the enhancement effect of minimum energy fusion was best. CCA fusion is used to study the linear relationship between two groups of multidimensional variables and is also a commonly used spatial filtering method [6].

Since Friman et al. demonstrated that minimum energy fusion and maximum contrast fusion had better fusion effects and did not compare minimum energy fusion and maximum contrast fusion with CCA fusion, this study only compared the enhancement effects of minimum energy fusion, maximum contrast fusion, CCA fusion, and the proposed method on the active components of the SSMVEP. Minimum energy fusion, maximum contrast fusion, and CCA fusion are used to analyze multidimensional EEG signals in the time domain. This study aims to investigate whether electrode signals can be fused in the time-frequency (T-F) domain. The method of image fusion can fuse multiple images into one, thus improving the image quality [10–12]. This study transformed EEG from the time domain to the T-F domain and analyzed multidimensional EEG with the idea of image fusion, to improve the enhancement effect of existing spatial filtering methods on SSMVEP active components. Firstly, two electrode signals were transformed from the time domain to the T-F domain via short-time Fourier transform (STFT). Then, two T-F images were decomposed by two-dimensional multiscale wavelet decomposition, and both the high-frequency coefficients and low-frequency coefficients of the wavelet were fused by specific fusion rules. After two-dimensional wavelet reconstruction, two images could be fused into one image. The fused image was filtered via mean filter, and the fused time-domain signal could be obtained by inverse STFT (ISTFT). The experimental results show that the proposed method based on T-F image fusion can enhance the SSMVEP better than the traditional time-domain analysis method.

2. Methods

2.1. Subjects. Six males and four females (20–28 years old) were recruited as subjects for this study. The participants were healthy and had normal colour and visual perception.

2.2. Experimental Equipment. g.USBamp (g.tec, Austria) was used to collect the EEG. The sampling frequency of the equipment is 1200 Hz. A g.USBamp EEG amplifier and g.GAMMAbox active electrode system were combined to form the experimental platform. Prior to the experiment, the reference electrode was placed at the left ear of the subjects, and the ground electrode Fpz was placed at the forehead. EEGs were collected from the following six channels: PO7, Oz, PO8, PO3, POz, and PO4.

2.3. Experimental Step. A checkerboard of radial contraction-expansion motion was used as a stimulation paradigm, and the details of the paradigm can be found in Reference [1]. The stimulus paradigms were presented on a screen at a refresh rate of 144 Hz, and the subjects were positioned 0.6–0.8 m from the screen. The experiment included six blocks, each containing 20 trials corresponding to all 20 targets. The stimuli were arranged in a 5×4 matrix, and the horizontal and vertical intervals between two neighboring stimuli were 4 cm and 3 cm, respectively. The stimulus frequencies were 7–10.8 Hz with a frequency interval of 0.2 Hz, and the radius of the stimuli was 60 pixels. Each trial lasted 5 s, separated by an interval of 2 s. Between two blocks, the subjects were allowed to rest properly. Twenty targets were simultaneously presented on the screen and numbered 1 to 20. Before the stimulus began, one of the 20 serial numbers appeared below the corresponding target, indicating the focus target.

2.4. Preprocessing of EEG Data. The corresponding EEG data segments were extracted in accordance with the trial start and end times. The MATLAB library function *detrend* was used to remove the linear trends for each channel. Chebyshev bandpass filtering of 0.5–50 Hz was used to remove low-frequency drifts and high-frequency interferences.

2.5. Significance Test. The data were expressed as mean values. A paired-sample t-test was used to determine the significance. Statistical significance was defined as $p < 0.05$.

2.6. Spatial Filtering Methods: Minimum Energy Fusion, Maximum Contrast Fusion, and CCA Fusion. For EEG data Y recorded from multiple electrode leads, the spatial filtering method obtains new data by linearly summing each of the electrode signals with spatial filter coefficients W. A spatially filtered signal can be expressed as

$$Z = W^T Y. \tag{1}$$

The spatial filter coefficients W for minimum energy fusion, maximum contrast fusion, and CCA fusion are described below.

2.6.1. Minimum Energy Fusion. For a visual stimulus frequency f, the SSMVEP signal recorded by the ith electrode can be expressed as follows:

$$y_i = \sum_{j=1}^{n} a_{i,j} \sin\left(2j\pi f t + \phi_{i,j}\right) + e_i(t), \tag{2}$$

where n represents the number of harmonics and $a_{i,j}$ and $\phi_{i,j}$ represent the amplitude and phase of the jth harmonic component, respectively. The model decomposes the signal into the sum of the SSMVEP induced by the visual stimulus and the noise $e_i(t)$ generated by the electromyogram (EMG), electrooculogram (EOG), and other components. Thus, equation (2) can be expressed as

$$y_i = S_i + e_i, \tag{3}$$

where

$$\begin{aligned} S_i &= a_i X, \\ X &= [X_1, X_2, \ldots, X_n]. \end{aligned} \tag{4}$$

The submatrix X_n is composed of $\sin(2n\pi ft)$ and $\cos(2n\pi ft)$, and a_i represents the amplitude of SSMVEP at its stimulus frequency and harmonics. We assume that the signal matrix recorded by N electrodes is $Y = [Y_1\ Y_2 \ldots Y_N]$, where each column corresponds to an electrode channel. Y can be projected to the SSMVEP space through a projection matrix Q, namely,

$$S = QY. \tag{5}$$

Reference [13] gives the solution of Q as

$$Q = X\left(X^T X\right)^{-1} X^T. \tag{6}$$

Thus, the noise signal can be expressed as

$$Y' = Y - QY. \tag{7}$$

The minimum energy fusion obtains the spatial filter coefficients W by minimizing the noise energy, namely,

$$\min_W \left\|Y'W\right\|^2 = \min_W W^T Y' Y' W. \tag{8}$$

The above minimization problem can be obtained by decomposing the eigenvalues of the positive definite matrix $Y'^T Y'$. After decomposition, N eigenvalues and corresponding eigenvectors can be obtained. The spatial filter coefficient W is the eigenvector corresponding to the smallest eigenvalue (MATLAB code for minimum energy fusion and test data is provided in Supplementary Materials (available here)).

2.6.2. Maximum Contrast Fusion. The goal of maximum contrast fusion is to maximize SSMVEP energy while minimizing noise energy. The SSMVEP energy can be approximated as $W^T Y^T Y W$. Therefore, the maximum contrast fusion can be achieved by the following maximizing equation:

$$\max_W \frac{\|YW\|^2}{\left\|Y'W\right\|^2} = \max_W \frac{W^T Y^T Y W}{W^T Y'^T Y' W}. \tag{9}$$

Here, Y and Y' are the same as in Section 2.6.1. Equation (9) can be solved by generalized eigenvalue decomposition of $Y^T Y$ and $Y'^T Y'$. The spatial filter coefficient W is the eigenvector corresponding to the largest eigenvalue (MATLAB code for maximum contrast fusion and test data is provided in Supplementary Materials (available here)).

2.6.3. Canonical Correlation Analysis Fusion. CCA fusion is used to study the linear relationship between two groups of multidimensional variables. Using two sets of signals Y and X, the goal is to find two linear projection vectors w_Y and w_X so that the two groups of linear combination signals $w_Y^T Y$ and $w_X^T X$ have the largest correlation coefficients:

$$\max_{w_Y, w_X} p = \frac{E\left(w_Y^T Y X^T w_X\right)}{\sqrt{E\left(w_Y^T Y Y^T w_Y\right) E\left(w_X^T X X^T w_X\right)}}. \tag{10}$$

The reference signals were constructed at the stimulation frequency f_d:

$$X = \begin{Bmatrix} \cos\left(2\pi f_d t\right) \\ \sin\left(2\pi f_d t\right) \\ \vdots \\ \cos\left(2k\pi f_d t\right) \\ \sin\left(2k\pi f_d t\right) \end{Bmatrix}, \quad t = \frac{1}{f_s}, \ldots, \frac{l}{f_s}, \tag{11}$$

where k is the number of harmonics, which is dependent on the number of frequency harmonics existing in SSMVEP; f_s is the sampling rate; and l represents the number of sample points. The optimization problem of equation (10) can be transformed into the eigenvalue decomposition problem, and the spatial filter coefficient W is the eigenvector corresponding to the largest eigenvalue (MATLAB code for CCA fusion and test data is provided in Supplementary Materials (available here)).

2.7. Two-Dimensional T-F Image Representation and Reconstruction of EEG Signals. The T-F analysis can transform the signal from the time domain to the T-F domain, and then, the T-F domain signal can be regarded as an image for analysis. In this study, T-F transform was performed on the time-domain EEG signal using STFT. Let $h(t)$ be a time window function with center at τ, a height of 1, and a finite width. The portion of the signal $x(t)$ observed by $h(t-\tau)$ is $x(t)h(t-\tau)$. The STFT is generated by the fast Fourier transform (FFT) of the windowed signal $x(t)h(t-\tau)$:

$$\mathrm{STFT}_x(\tau, f) = \int_{-\infty}^{+\infty} x(t) h(t-\tau) e^{-j2\pi f t} dt. \tag{12}$$

Equation (12) can map the signal $x(t)$ onto the two-dimensional T-F plane (τ, f), and f can be regarded as the frequency in the STFT. After fusing multiple EEG T-F images, the fused two-dimensional image signals need to be transformed into one-dimensional time-domain signals for the subsequent analysis. The STFT has an inverse transform that transforms the T-F domain signal into a time-domain signal:

$$x(t) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mathrm{STFT}(\tau, f)h(t-\tau)e^{j2\pi ft}df\,d\tau. \quad (13)$$

In this study, the T-F transform of the SSMVEP signal was conducted by MATLAB's *tfrstft* function. After fusing two T-F images, the T-F signal was inversely transformed into a one-dimensional time-domain signal by MATLAB's *tfristft* function.

2.8. T-F Image Fusion Based on Two-Dimensional Multiscale Wavelet Transform. Among the many image fusion technologies, two-dimensional multiscale wavelet transform-based image fusion methods have become a research hotspot [14–16]. Let $f(x, y)$ denote a two-dimensional signal, and x and y are its abscissa and ordinate, respectively. Let $\psi(x, y)$ denote a two-dimensional basic wavelet and $\psi_{a;b_1,b_2}(x, y)$ denote the scale expansion and two-dimensional displacement of $\psi(x, y)$:

$$\psi_{a;b_1,b_2}(x, y) = \frac{1}{a}\psi\left(\frac{x-b_1}{a}, \frac{y-b_2}{a}\right). \quad (14)$$

Then, the two-dimensional continuous wavelet transform is

$$\mathrm{WT}_f(a; b_1, b_2) = \frac{1}{a}\int\int f(x, y)\psi\left(\frac{x-b_1}{a}, \frac{y-b_2}{a}\right)dx\,dy. \quad (15)$$

The factor $1/a$ in equation (15) is a normalization factor that has been introduced to ensure that the energy remains unchanged before and after wavelet expansion. The two-dimensional multiscale wavelet decomposition and reconstruction process is shown in Figure 1. The decomposition process can be described as follows: First, one-dimensional discrete wavelet decomposition is performed on each line of the image, which obtains the low-frequency component L and the high-frequency component H of the original image in the horizontal direction. The low-frequency component and high-frequency component are the low-frequency wavelet coefficients and high-frequency wavelet coefficients obtained after wavelet decomposition, respectively. Then, one-dimensional discrete wavelet decomposition is performed on each column of the transformed data to obtain the low-frequency components LL in the horizontal and vertical directions, the high-frequency components LH in the horizontal and vertical directions, the low-frequency components HL in the horizontal and vertical directions, and the high-frequency component HH in the vertical direction. The reconstruction process can be described as follows: Firstly, one-dimensional discrete wavelet reconstruction is performed on each column of the transform result. Then, one-dimensional discrete wavelet reconstruction is performed on each row of the transformed data to obtain the reconstructed image.

The T-F image obtained by the STFT is decomposed into N layers by the two-dimensional multiscale wavelet decomposition, as shown in Figure 1, and $3N+1$ frequency components are obtained. In this study, the LL_N decomposed from the highest level is defined as the low-frequency component, and the HL_M, LH_M, and HH_M ($M = 1, 2, \ldots, N$) decomposed from each level are defined as high-frequency components. The active components of the EEG are mainly contained in the low-frequency component. The fusion process of two T-F images based on two-dimensional multiscale wavelet transform is shown in Figure 2. Firstly, the T-F images 1 and 2 are decomposed by two-dimensional wavelet decomposition, and then, the corresponding low-frequency components and high-frequency components are fused according to the corresponding fusion rules. Finally, two-dimensional wavelet reconstruction is performed to obtain the fused image by fusing low-frequency components and high-frequency components. In this study, the two-dimensional wavelet decomposition was performed using the MATLAB's *wavedec2* function and the two-dimensional wavelet reconstruction was performed using the MATLAB's *waverec2* function.

2.9. Image Mean Filtering. For the current pixel to be processed, a template is selected, which is composed of several pixels adjacent to the current pixel. The method of replacing the value of the original pixel with the mean of the template is called mean filtering. Defining $f(x, y)$ as the pixel value at coordinates (x, y), the current pixel $g(x, y)$ can be calculated by

$$g(x, y) = \frac{1}{D}\sum_{i=1}^{D} f(x_i, y_i), \quad (16)$$

where D represents the square of the template size.

2.10. Summary of the Proposed Method for EEG Enhancement Based on T-F Image Fusion. Here, the T-F image fusion method of the electrode signals $x_1(t)$ and $x_2(t)$ is introduced, and the fusion process is shown in Figure 3.

(1) STFT is performed on the two electrode signals $x_1(t)$ and $x_2(t)$ to obtain T-F images 1 and 2.

(2) The low-frequency components $\mathrm{LL}_{N,1}$ and $\mathrm{LL}_{N,2}$ and the high-frequency components ($\mathrm{HL}_{M,1}$, $\mathrm{LH}_{M,1}$, and $\mathrm{HH}_{M,1}$) and ($\mathrm{HL}_{M,2}$, $\mathrm{LH}_{M,2}$, and $\mathrm{HH}_{M,2}$) ($M = 1, 2, \ldots, N$) of two T-F images 1 and 2 are obtained by N-layer two-dimensional multiscale wavelet decomposition.

(3) The low-frequency components $\mathrm{LL}_{N,1}$ and $\mathrm{LL}_{N,2}$ and the high-frequency components ($\mathrm{HL}_{M,1}$, $\mathrm{LH}_{M,1}$, and $\mathrm{HH}_{M,1}$) and ($\mathrm{HL}_{M,2}$, $\mathrm{LH}_{M,2}$, and $\mathrm{HH}_{M,2}$) ($M = 1, 2, \ldots, N$) of each decomposition layer are fused according to the following fusion rules:

(a) The low-frequency components of the T-F images 1 and 2 are subtracted to obtain the low-frequency components of the fused image F:

$$\mathrm{LL}_{N,F} = \mathrm{LL}_{N,1} - \mathrm{LL}_{N,2}. \quad (17)$$

The low-frequency components represent the active components of the SSMVEP, and the subtraction of the

Figure 1: The process of two-dimensional discrete wavelet decomposition and reconstruction.

Figure 2: Image fusion process based on two-dimensional wavelet transform.

Figure 3: The fusion process of two-electrode signal T-F images.

low-frequency components can remove the common noise in the electrode signals.

(b) The magnitudes of the high-frequency components ($HL_{M,1}$, $LH_{M,1}$, and $HH_{M,1}$) and ($HL_{M,2}$, $LH_{M,2}$, and $HH_{M,2}$) ($M = 1, 2, ..., N$) of each decomposition layer are obtained for the T-F images 1 and 2. The high-frequency component with a large value is used as the high-frequency component of the fused image F:

$$\begin{cases} HL^i_{M,F} = \begin{cases} HL^i_{M,1}, \text{real}\left(HL^i_{M,1}\right) > \text{real}\left(HL^i_{M,2}\right), \\ HL^i_{M,2}, \text{real}\left(HL^i_{M,1}\right) < \text{real}\left(HL^i_{M,2}\right), \end{cases} \\ LH^i_{M,F} = \begin{cases} LH^i_{M,1}, \text{real}\left(LH^i_{M,1}\right) > \text{real}\left(LH^i_{M,2}\right), \\ LH^i_{M,2}, \text{real}\left(LH^i_{M,1}\right) < \text{real}\left(LH^i_{M,2}\right), \end{cases} \\ HH^i_{M,F} = \begin{cases} HH^i_{M,1}, \text{real}\left(HH^i_{M,1}\right) > \text{real}\left(HH^i_{M,2}\right), \\ HH^i_{M,2}, \text{real}\left(HH^i_{M,1}\right) < \text{real}\left(HH^i_{M,2}\right), \end{cases} \end{cases} \quad M = 1, 2, \ldots, N. \tag{18}$$

The two-dimensional signal after the STFT is a complex signal; the two-dimensional signal after two-dimensional wavelet decomposition is still a complex signal. The value used for the comparison is the real part of the complex signal. Here, i represents the number of frequency components of the M decomposition layer (i.e., the number of wavelet coefficients).

(4) According to both low-frequency components $LL_{N,F}$ and high-frequency components ($HL_{M,F}$, $LH_{M,F}$, and $HH_{M,F}$) ($M = 1, 2, ..., N$) of the fused image F, the fused image F is generated by two-dimensional wavelet reconstruction.

(5) Mean filtering is performed on the fused image F.

(6) An ISTFT is performed on the mean filtered image to generate a fused time-domain signal (MATLAB code for T-F image fusion and test data is provided in Supplementary Materials (available here)).

3. Results

3.1. Parameter Selection

3.1.1. STFT Parameters. In this study, the EEG time-domain signals were subjected to STFT and ISTFT, using the MATLAB library functions *tfrstft* and *tfristft*. The parameters that need to be set here are the number of frequency bins and the frequency smoothing window. If the number of frequency bins is too large, the calculation speed of the entire process will be low. In this study, the value of this parameter can be set to [32 54 76 98 120]. Since the Fourier transform of the Gaussian function is also a Gaussian function, the window function of the optimal time localization of the STFT is a Gaussian function. The EEG is a nonstationary signal [17] and requires a high time resolution of the window function; that is, the window length should not be too long. In this paper, a Gaussian window was selected, and the value of the window length can be set to [37 47 57 67 77 87 97].

3.1.2. Two-Dimensional Multiscale Wavelet Transform Parameters. In this study, the MATLAB library functions *wavedec2* and *waverec2* were used to perform two-dimensional multiscale wavelet decomposition and reconstruction on T-F images. Here, the number of layers of wavelet decomposition and the wavelet basis functions need to be set. The sampling frequency of the EEG acquisition device used in this paper was 1200 Hz, and the stimulation frequency of the SSMVEP was 7–10.8 Hz. Considering that the SSMVEP may contain the second harmonic of the stimulation frequency, this study sets the number of layers of wavelet decomposition to 4. The db_N wavelet has good regularity and was chosen as the wavelet basis function. The vanishing moment N of the wavelet function can be set to [2 3 5 7 10 13].

3.1.3. Mean Filtering Parameters. Here, the template size of the mean filtering needs to be set. In this study, the value of the template size can be set to [2 4 6 8 10 12 14 16 18 20 22 24 26].

3.1.4. Low-Frequency Component of the Fused Image F. The values of $LL_{N,F}$ taking $LL_{N,1}$–$LL_{N,2}$ or $LL_{N,2}$–$LL_{N,1}$ will affect the experimental results, and the values of $LL_{N,F}$ need to be determined to optimize the final fusion effect (see details in Section 2.10 (3)).

3.1.5. Parameter Selection. The T-F image fusion method proposed in this study requires the setting of the following five parameters: the number of frequency bins, frequency smoothing window, vanishing moments of db_N wavelet functions, mean filter template size, and the value of $LL_{N,F}$. In this study, the grid search method was used to traverse the combination of all parameters to find the best combination of parameters that can enhance the SSMVEP active components. In this study, six rounds of experiments were conducted. The first three rounds of experimental data were used to select the optimal combination of parameters for T-F

image fusion. According to the selected combination of parameters, the last three rounds of experimental data were used to compare the enhancement effect of SSMVEP active components by the method of minimum energy fusion, maximum contrast fusion, CCA fusion, and T-F image fusion. The disadvantage of grid search is that the amount of calculation is large; however, considering the difficulty of superparameter selection, grid search is still a more reliable method.

3.2. Electrode Signal Selection for T-F Image Fusion. The T-F image fusion method analyzes two electrode signals. Thus, two suitable electrodes need to be selected from multiple electrodes. The bipolar fusion signal is obtained by subtracting the two original EEG signals. In equation (17), we subtracted the wavelet low-frequency components of the two T-F images to obtain the wavelet low-frequency components of the fused image. The subtraction of the low-frequency components removes the common noise in the electrode signals, which is similar to the principle of bipolar fusion. Therefore, the two electrode signals with the best bipolar fusion effect were used in the T-F image fusion. For SSVEP or SSMVEP stimulation, the Oz electrode usually has the strongest response [18] and is most commonly used to analyze SSVEP or SSMVEP [19]. The Oz electrode was used as one of the analysis electrodes for bipolar fusion. In this study, the Oz electrode was fused with the other five electrode signals so that the best combination of electrodes can be selected. When bipolar fusion is used, the Oz electrode as a reducing electrode or a reduced electrode will not affect the experimental results. For example, Oz-POz and POz-Oz have the same fusion effect. In this study, the Oz electrode was used as the reduced electrode. Twenty trials per subject were used for electrode selection. The Welch power spectrum analysis [20] was performed on the obtained signals of bipolar fusion for each subject. The Gaussian window was used, and the length of the window was 5 s. The overlapping was 256, and the number of discrete Fourier transform (DFT) points was 6000 for the Welch periodogram. 20 focused targets correspond to 20 stimuli frequencies, and the frequencies corresponding to the highest power spectrum amplitudes at the 20 frequencies were determined as the focused target frequency. The high recognition accuracy indicates that the amplitude of the power spectrum at the stimulation frequency is prominent, and it indicates the effectiveness of the fusion method for the enhancement of the SSMVEP active component. Table 1 shows the recognition accuracy of all 10 subjects under the bipolar fusion. The Oz electrode was fused with different electrodes and obtained different fusion effects. Moreover, due to the differences among individuals, the electrode combination with the highest recognition accuracy per subject was also different. The asterisk (*) marks the bipolar fusion electrode group with the highest recognition accuracy of each subject, and the corresponding electrode combination was used in T-F image fusion.

3.3. Comparison of Enhancement Effects of Minimum Energy Fusion, Maximum Contrast Fusion, CCA Fusion, and T-F Image Fusion on SSMVEP Active Components. T-F image fusion used two channel signals, and CCA fusion, maximum contrast fusion, and minimum energy fusion used all six channel signals. In this study, the accuracy was used as the evaluation standard of the fusion effect. The high online accuracy indicates that the amplitude at stimulus frequency is prominent, which shows the effectiveness of the fusion method. For T-F image fusion, the Welch power spectrum analysis (the parameters are the same as in Section 3.2) was performed on the obtained signal after the T-F image fusion, and the frequency with the highest amplitude at twenty stimulus frequencies was identified as the focused target frequency. The minimum energy fusion, maximum contrast fusion, and CCA fusion require prior knowledge of the frequency of the signal to be fused. However, the frequency of the signal to be fused cannot be determined because the user's focused target cannot be determined beforehand. For the current tested signal, the minimum energy fusion, maximum contrast fusion, and CCA fusion need to first perform the fusion analysis at all stimulus frequencies, and then, the frequency with the highest amplitude was identified as the focused target frequency. Take CCA fusion as an example, and assume that the current tested signal is Y. First, 20 template signals are set according to equation (11), and then, 20 spatial filter coefficients are obtained according to equation (10). The tested signal is fused with 20 spatial filter coefficients, respectively. The obtained 20 sets of vectors are separately analyzed by the Welch power spectrum (the parameters are the same as in Section 3.2) to obtain the amplitude at the corresponding frequency. Finally, the frequency with the highest amplitude is identified as the focused target frequency. The first three rounds of experimental data were used to select the parameters of the T-F image fusion (see details in Section 3.1), and the last three rounds of the experimental data were used to test the online fusion effects under the four methods. Figures 4(a)–4(d) show the analysis results plotted using the data of subject 4 in the fourth round of the experiment. Figure 4(a) shows the power spectrum results of the 20 targets under T-F image fusion, Figure 4(b) shows the power spectrum results of 20 targets under CCA fusion, Figure 4(c) shows the power spectrum results of 20 targets under minimum energy

Table 1: Accuracies of 10 subjects under bipolar fusion.

Subjects	Bipolar fusion				
	Oz-PO7	Oz-PO8	Oz-PO3	Oz-POz	Oz-PO4
S1	0.2	0.7	0.45	0.15	0.95*
S2	0.25	0	0.3*	0.1	0.2
S3	0.95*	0.85	0.85	0.4	0.8
S4	0.8*	0.1	0.7	0.35	0.35
S5	0.2*	0.15	0.15	0.15	0.1
S6	0.95*	0.3	0.95*	0.85	0.7
S7	0.2	0.15	0.25	0.6*	0.55
S8	0.1	0.05	0.25	0.3	0.9*
S9	0.3	0.15	0	0.35*	0.2
S10	0.35	0.35	0.5	0.8*	0.4

*The bipolar fusion electrode group with the highest recognition accuracy of each subject.

(a)

(b)

Figure 4: Continued.

(c)

(d)

Figure 4: The power spectrum results of 20 targets under (a) T-F image fusion, (b) CCA fusion, (c) minimum energy fusion, and (d) maximum contrast fusion (the discrete sequence frequency values from left to right are {7 8 9 10 7.2 8.2 9.2 10.2 7.4 8.4 9.4 10.4 7.6 8.6 9.6 10.6 7.8 8.8 9.8 10.8} Hz, respectively).

fusion, and Figure 4(d) shows the power spectrum results of 20 targets under maximum contrast fusion. In the stalk plot, the discrete sequence frequency values from left to right are {7 8 9 10 7.2 8.2 9.2 10.2 7.4 8.4 9.4 10.4 7.6 8.6 9.6 10.6 7.8 8.8 9.8 10.8}, respectively, where f represents the stimulation frequency of the corresponding target and the green dot marks the amplitude of the corresponding target stimulus frequency. The red flag f represents the target was misidentified, and the blue flag f represents the target was correctly identified. Figures 4(a)–4(d) show that the T-F image fusion method has the best online performance and the minimum energy fusion method has the worst online performance. Maximum contrast fusion and CCA fusion have similar fusion effects.

In online analysis, the power spectrum amplitudes of the test signal at all stimulus frequencies were calculated, and then, the frequency corresponding to the maximum amplitude was identified as the focused target frequency. Error recognition occurs when the amplitude at the focused target frequency is lower than the amplitudes at the nonfocused target frequencies. Therefore, the power spectrum amplitude at the focused target frequency can be regarded as the active component of the signal, and the power spectrum amplitudes at the remaining nonfocused target frequencies can be regarded as noise. The SNR in this study was defined as the ratio of the power spectrum amplitude at the focused target frequency to the mean of the power spectrum amplitudes at the remaining nonfocused target frequencies. The SNRs of ten subjects at all stimulus frequencies were superimposed and averaged, and the experimental results are shown in Figure 5. It can be seen from Figure 5 that the T-F image fusion method obtained the highest SNR, which indicates that the T-F image fusion method proposed in this study can effectively enhance the SSMVEP SNR.

The online accuracies of each subject in the three rounds of experiments were superimposed and averaged. The experimental results are shown in Figure 6. It can be seen that most subjects obtained highest online accuracy under the T-F image fusion and some subjects obtained highest online accuracy under CCA fusion and maximum contrast fusion. Only subject 9 obtained the highest online accuracy under the minimum energy fusion. The online accuracies of all the subjects were superimposed and averaged. The experimental results are shown in Figure 7. The experimental results show that the online performance of the T-F image fusion method is better. The online accuracy of the T-F image fusion method is 6.17%, 6.06%, and 30.50% higher than that of the CCA fusion, maximum contrast fusion, and the minimum energy fusion, respectively. The paired t-test shows significant differences between the accuracies of T-F image fusion and minimum energy fusion ($p = 0.0174$). Online accuracy analysis results show that the proposed method is better than the traditional time-domain fusion method.

4. Discussion

The SSMVEP collected from the scalp contains a lot of noise and requires an effective signal processing method to enhance the active components of SSMVEP. Spatial filtering methods utilize EEG information of multiple channels and exert positive significance to enhance the active components of the SSMVEP. At present, the commonly used spatial filtering methods include average fusion, native fusion, bipolar fusion, Laplacian fusion, CAR fusion, CCA fusion, minimum energy fusion, and maximum contrast fusion. The first five spatial filtering methods can only process the preselected electrode signals. Minimum energy fusion, maximum contrast fusion, and CCA fusion improve the above problem, and they can be used to fuse any number of electrode signals with better fusion effects. Minimum energy fusion, maximum contrast fusion, and CCA fusion fuse SSMVEP in the time domain. This study first put forward the idea of fusing EEG in the T-F domain and proposed a T-F image fusion method. We compared the enhancement effects of minimum energy fusion, maximum contrast fusion, CCA fusion, and T-F image fusion on SSMVEP. It is verified by the test data that the proposed method is better than the traditional time-domain fusion method.

Figure 5: The average SNR of ten subjects at all stimulus frequencies under four methods.

T-F analysis is a good choice for the transformation of the EEG data from one dimension to two dimension. EEG transformed into the T-F domain can be used as an image for analysis. The STFT has an inverse transform, which can inverse transform the fused T-F image into a time-domain signal. Therefore, the STFT was used to transform the EEG from the time domain to the T-F domain. The two-dimensional wavelet transform can fuse two images into one to achieve the purpose of multichannel EEG fusion. After two-dimensional wavelet decomposition, the low-frequency components of the two subimages were subtracted to obtain the low-frequency components of the fused image, and the maximum value of the high-frequency components of the two subimages was taken as the high-frequency component of the fused image. The low-frequency components after wavelet decomposition represent the active components of the SSMVEP. The subtraction of the low-

Figure 6: The accuracy of each subject under four fusion methods.

Figure 7: The average accuracy of ten subjects under four fusion methods.

frequency components can remove the common noise in the electrode signals. This is similar to bipolar fusion, which subtracts two electrode signals in the time domain. We also compared the enhancement effects of bipolar fusion and the proposed method on SSMVEP active components. The electrodes used for bipolar fusion are the same as those used in T-F image fusion. The results show that the proposed method is better than the bipolar fusion (the online accuracy of bipolar fusion is 71.52%). In this study, the fused image was filtered by mean filter to further achieve low-pass filtering. We also tried to remove the mean filtering step after completing the T-F image fusion. The results show that removing the mean filtering step reduced the fusion effect. In this study, the mean filtering step was performed after T-F image fusion. We tested the fusion effect when the mean filtering step was performed before T-F image fusion (performing mean filtering step on the subimage after STFT). For some subjects, it was better to perform the mean filtering step after T-F image fusion. Therefore, we recommend that researchers follow the analysis steps in this study.

The wavelet low-frequency coefficients (see details in equation (17)) of the fused image have an important influence on the fusion effect. If we perform T-F image fusion on six electrode signals at the same time, we will get six sets of wavelet low-frequency coefficients. Here, each set of low-frequency coefficient is a two-dimensional signal. Referring to equation (1), we can assign a coefficient to each two-dimensional signal and obtain the fused two-dimensional signal (i.e., the wavelet low-frequency coefficients of the fused image) after linear summation. This study explored the feasibility of applying the spatial filter coefficients of CCA fusion and maximum contrast fusion to T-F image fusion when fusing six electrode signals by using the T-F image fusion method at the same time. Since the fusion effect of minimum energy fusion was poor, the spatial filter coefficients of the minimum energy fusion were not used here. The spatial filter coefficients of the maximum contrast fusion and CCA fusion at the frequency f were obtained by equations (9) and (10) and are set to $V_{\max}$ and V_{cca}, respectively. $V_{\max}$ and V_{cca} are two vectors with dimensions of 6×1, where $V_{\max}(1, 1)$ and $V_{\text{cca}}(1, 1)$ represent the first elements of the vectors $V_{\max}$ and V_{cca}. Since the T-F image fusion method was performed on the six electrode signals, equation (17) is transformed into equation (19), where V corresponds to the spatial filter coefficient $V_{\max}$ or V_{cca}. The rest of the analysis process is the same as that in Section 2.10. The same analysis steps were performed for all 20 targets (7–10.8 Hz). The Welch spectrum analysis (the parameters are the same as in Section 3.2) was performed on the obtained signal after T-F image fusion. Figures 8(a) and 8(b) show the Welch power spectra plotted using the data of subject 4 in the fourth round of the experiment. Figure 8(a) shows the power spectrum plotted using spatial filter coefficients of CCA fusion and Figure 8(b) shows the Welch power spectrum plotted using spatial filter coefficients of maximum contrast fusion, where f represents the stimulus frequency and the red circle indicates the amplitude at the stimulus frequency. Figures 8(a) and 8(b) show that the amplitudes at the stimulation frequencies are prominent. The spatial filter coefficients of CCA fusion and maximum contrast fusion are effective for T-F image fusion. Thus, the T-F image fusion method proposed in this study focuses on the fusion of wavelet low-frequency coefficients of multiple

Figure 8: The Welch power spectrum plotted using the (a) CCA fusion spatial filter coefficients and (b) maximum contrast fusion spatial filter coefficients.

images (by assigning a coefficient to each two-dimensional signal and obtaining the fused two-dimensional signal after linear summation). The premise of the above test results is that we know the frequency of the signal to be fused. If the online test method is used (see details in Section 3.3), the same online fusion results as CCA fusion and maximum

contrast fusion were obtained. T-F image fusion used only two electrode signals, and the better online fusion effect than that of the CCA fusion and the maximum contrast fusion was obtained, which shows the effectiveness of the proposed method. Next, we will explore the fusion method of multiple T-F images (more than two images), that is, to find suitable spatial filter coefficients for the wavelet low-frequency components of multiple T-F images:

$$\begin{aligned} \mathrm{LL}_{N,F} = {} & \mathrm{LL}_{N,1} * V(1,1) + \mathrm{LL}_{N,2} * V(2,1) + \mathrm{LL}_{N,3} * V(3,1) \\ & + \mathrm{LL}_{N,4} * V(4,1) + \mathrm{LL}_{N,5} * V(5,1) \\ & + LL_{N,6} * V(6,1). \end{aligned} \tag{19}$$

The parameters affect the fusion effect of the proposed method. We listed the parameters that need to be selected and the possible values of the parameters in Section 3.1. In this study, the grid search method was used to traverse the combination of all parameters, and the best combination of parameters was found. In the experiment, we found that the number of frequency bins and the Gaussian window length can be fixed to 54 and 57. We recommend that the researchers determine the parameters according to the parameter selection principles and ranges given in Section 3.1. In this study, the SSMVEP active component was enhanced by fusing multichannel signals into a single-channel signal, and then, the focused target frequency was identified by performing spectrum analysis on the fused signals. This is beneficial for SSMVEP studies based on spectrum analysis. For example, Reference [21] proposed a frequency and phase mixed coding method in the SSVEP-based brain-computer interface (BCI), which increases the number of BCI coding targets by making one frequency correspond to multiple different phases. In the study, the FFT analysis of the test signal is required to find the possible focused target frequency and then calculate the phase value at that frequency. The proposed method in this study has a positive significance for accurately finding the focused target frequency in the spectrum. Moreover, some EEG feature extraction algorithms for the BCI also require spectrum analysis of the test signals [22, 23]. Therefore, the method proposed in this study has a potential application value.

5. Conclusion

To explore whether T-F domain analysis can achieve better fusion effects than time-domain analysis, this study proposed an SSMVEP enhancement method based on T-F image fusion. The parameters of the T-F image fusion algorithm were determined by the grid search method, and the electrode signals used for T-F image fusion were selected by bipolar fusion. The analysis results show that the key of the T-F image fusion algorithm is the fusion of the wavelet low-frequency components. This study compared the enhancement effects of minimum energy fusion, maximum contrast fusion, CCA fusion, and T-F image fusion on SSMVEP. The experimental results show that the online performance of the T-F image fusion method is better than that of the traditional spatial filtering methods, which indicates that the proposed method is feasible to fuse SSMVEP in the T-F domain.

Data Availability

The analytical data used to support the findings of this study are included within the article, and the raw data are available from the corresponding author upon request.

Ethical Approval

The subjects provided informed written consent, in accordance with the protocol approved by the Institutional Review Board of Xi'an Jiaotong University.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Wenqiang Yan worked on the algorithm and wrote the paper. Guanghua Xu involved in discussions and gave suggestions throughout this project. Longting Chen and Xiaowei Zheng worked on the experimental design. All authors read and approved the final manuscript.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (NSFC) (no. 51775415). We want to thank the subjects for participating in these experiments.

Supplementary Materials

MATLAB codes for T-F image fusion, minimum energy fusion, maximum contrast fusion, and CCA fusion. (*Supplementary Materials*)

References

[1] W. Yan, G. Xu, J. Xie et al., "Four novel motion paradigms based on steady-state motion visual evoked potential," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 8, pp. 1696–1704, 2018.

[2] J. R. Wolpaw and D. J. Mcfarland, "Multichannel EEG-based brain-computer communication," *Electroencephalography and Clinical Neurophysiology*, vol. 90, no. 6, pp. 444–449, 1994.

[3] B. O. Peters, G. Pfurtscheller, and H. Flyvbjerg, "Mining multi-channel EEG for its information content: an ANN-based method for a brain-computer interface," *Neural Networks*, vol. 11, no. 7-8, pp. 1429–1433, 1998.

[4] S. Parini, L. Maggi, A. C. Turconi, and G. Andreoni, "A robust and self-paced BCI system based on a four class SSVEP paradigm: algorithms and protocols for a high-transfer-rate direct brain communication," *Computational Intelligence and Neuroscience*, vol. 2009, pp. 1–11, 2009.

[5] O. Friman, I. Volosyak, and A. Graser, "Multiple channel detection of steady-state visual evoked potentials for brain-

computer interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 4, pp. 742–750, 2007.

[6] Z. Lin, C. Zhang, W. Wu, and X. Gao, "Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs," *IEEE Transactions on Biomedical Engineering*, vol. 53-, no. 12, pp. 2610–2614, 2006.

[7] M. Middendorf, G. Mcmillan, G. Calhoun, and K. S. Jones, "Brain-computer interfaces based on the steady-state visual-evoked response," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 211–214, 2000.

[8] G. R. Müller-Putz, R. Scherer, C. Brauneis, and G. Pfurtscheller, "Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components," *Journal of Neural Engineering*, vol. 2, no. 4, pp. 123–130, 2005.

[9] D. J. Mcfarland, L. M. Mccane, S. V. David, and J. R. Wolpaw, "Spatial filter selection for EEG-based communication," *Electroencephalography and Clinical Neurophysiology*, vol. 103, no. 3, pp. 386–394, 1997.

[10] H. Li, B. S. Manjunath, and S. K. Mitra, "Multisensor image fusion using the wavelet transform," *Graphical Models and Image Processing*, vol. 57, no. 3, pp. 235–245, 1995.

[11] S. Li and B. Yang, "Multifocus image fusion by combining curvelet and wavelet transform," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1295–1301, 2008.

[12] B. Zhang, "Study on image fusion based on different fusion rules of wavelet transform," *Acta Photonica Sinica*, vol. 33, pp. 173–176, 2004.

[13] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra Book and Solutions Manual*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[14] K. J. A. Sundar, M. Jahnavi, and K. Lakshmisaritha, "Multisensor image fusion based on empirical wavelet transform," in *Proceedings of the International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques*, pp. 93–97, Mysuru, India, December 2018.

[15] L. Tian, J. Ahmed, Q. Du, B. Shankar, and S. Adnan, "Multi focus image fusion using combined median and average filter based hybrid stationary wavelet transform and principal component analysis," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, 2018.

[16] G. He, S. Xing, D. Dong, and X. Zhao, "Panchromatic and multi-spectral image fusion method based on two-step sparse representation and wavelet transform," in *Asia-Pacific Signal and Information Processing Association Summit and Conference*, pp. 259–262, Honolulu, HI, USA, January 2018.

[17] Y. Shin, S. Lee, M. Ahn, H. Cho, S. C. Jun, and H.-N. Lee, "Noise robustness analysis of sparse representation based classification method for non-stationary EEG signal classification," *Biomedical Signal Processing and Control*, vol. 21, pp. 8–18, 2015.

[18] M. Xu, L. Chen, L. Zhang et al., "A visual parallel-BCI speller based on the time-frequency coding strategy," *Journal of Neural Engineering*, vol. 11, no. 2, article 026014, 2014.

[19] J. Xie, G. Xu, A. Luo et al., "The role of visual noise in influencing mental load and fatigue in a steady-state motion visual evoked potential-based brain-computer interface," *Sensors*, vol. 17, no. 8, p. 1873, 2017.

[20] K. Barbe, R. Pintelon, and J. Schoukens, "Welch method revisited: nonparametric power spectrum estimation via circular overlap," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 553–565, 2010.

[21] C. Jia, X. Gao, B. Hong, and S. Gao, "Frequency and phase mixed coding in SSVEP-based brain-computer interface," *IEEE transactions on bio-medical engineering*, vol. 58, no. 1, pp. 200–206, 2011.

[22] C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C.-Y. Su, "Mind control of a robotic arm with visual fusion technology," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3822–3830, 2017.

[23] J. Castillo, S. Muller, E. Caicedo, and T. Bastos, "Feature extraction techniques based on power spectrum for a SSVEP-BCI," in *Proceedings of the IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, Istanbul, Turkey, June 2014.

Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 9323482, 12 pages
https://doi.org/10.1155/2019/9323482

Research Article

Concurrent, Performance-Based Methodology for Increasing the Accuracy and Certainty of Short-Term Neural Prediction Systems

Miljana Milić,[1] **Jelena Milojković**,[2] **Ivan Marković**,[3] **and Petar Nikolić**[4]

[1]*Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia*
[2]*Innovation Centre of Advanced Technologies, Bulevar Nikole Tesle 61, Loc. 5, 18000 Niš, Serbia*
[3]*Faculty of Economics, University of Niš, Trg Kralja Aleksandra Ujedinitelja 11, 18000 Niš, Serbia*
[4]*Tigar Tyres, Nikole Pašića 213, 18300 Pirot, Serbia*

Correspondence should be addressed to Miljana Milić; miljana.milic@elfak.ni.ac.rs

Received 28 December 2018; Revised 25 February 2019; Accepted 7 March 2019; Published 1 April 2019

Guest Editor: Vlado Delic

Copyright © 2019 Miljana Milić et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate prediction of the short time series with highly irregular behavior is a challenging task found in many areas of modern science. Such data fluctuations are not systematic and hardly predictable. In recent years, artificial neural networks have widely been exploited for those purposes. Although it is possible to model nonlinear behavior of short time series by using ANNs, very often they are not able to handle all events equally well. Therefore, alternative approaches have to be applied. In this study, a new, concurrent, performance-based methodology that combines best ANN topologies in order to decrease the forecasting errors and increase the forecasting certainty is proposed. The proposed approach is verified on three different data sets: the Serbian Gross National Income time series, the municipal traffic flow for a particular observation point, and the daily electric load consumption time series. It is shown that the method can significantly increase the forecasting accuracy of the individual networks, regardless of their topologies, which makes the methodology more applicable. For quantitative comparison of the accuracy of the proposed methodology with that of similar methodologies, a series of additional forecasting experiments that include a state-of-the-art ARIMA modelling and a combination of ANN and linear regression forecasting have been conducted.

1. Introduction

Prediction is a process that uses data from the present and the past in order to estimate future. The result of this process is the information about probable events in the future and their effects and outcomes. Making good forecasts is essential for making good decisions and planning in all areas of life. Although it does not have to reduce uncertainties and difficulties of the future, it can increase the certainty and the level of the preparedness for challenges and environmental changes that future events bring.

The need for development of prediction methods occurs in almost every area of life—technology, engineering, industry, science, politics, economy, business, sport, medicine, etc. Good forecasts can ensure lower cost of the services and products, increased customer/client satisfaction, and significant competitive advantage [1].

Every daily activity begins with planning. The planning begins with a prediction [2]. Prediction errors may have crucial implications on decision-making, profits and investment justification, risk assessment, alerting events, hard real-time systems' actions, timely handling of emergency health and medical conditions, etc. [3]. Because of that, decreasing the error of the prediction is an essential task for every forecasting expert, regardless of the applied prediction methods.

Prediction methods described in the literature can be roughly categorized into two large groups: traditional and modern. Each of them has advantages and disadvantages. None of them is superior to all others if we consider all possible criteria of evaluation [4]. Traditional prediction methods try to extrapolate time series data using different modelling: exponential smoothing [5], linear or nonlinear regression [6–8], simple (AR) or more complex autoregressive

models (ARMA, ARIMA, and double seasonal ARIMA) [6, 9]. On the other hand, modern prediction methods exploiting the artificial intelligence (AI) behavior can model both nonlinear and linear structures of time series [10] and can produce good accuracy of the forecasting. Such techniques use different topologies of artificial neural network, fuzzy modelling, vector machines, and genetic-simulated annealing algorithms to predict time series data [4, 11–14]. Different authors have shown that AI-based models frequently express better predictive characteristics compared to models using standard multilinear regression [14]. Finally, both theoretical and empirical findings in the literature show that combination of two or more different methods can be an effective and efficient way to improve forecasts and decrease the error [10]. Such hybrid methods are studied in [2, 3, 10, 15–17].

Despite numerous ways to predict the future mathematically, there are many cases of variables that could not reliably be predicted. Causes for this limitation could be found in the randomness of the events and the lack of significant relationship in data. When factors considered during forecasting of a certain variable are not well known or understood, prediction becomes imprecise or mistaken. Sometimes, there is simply not enough data about everything that affects the forecasted variable. The prediction process relies on some specific hypothesis. If they are set wrong, due to bad judgment, i.e., human error, the prediction will be mistaken. Although the forecasting is based on past events, no one can guarantee that the history will repeat every time in the same way. Therefore, forecasts are subject to human error.

A time series can be defined as a sequence of numerical data occurring in regular intervals over a period of time, collected in a successive order. Short time series are characterized by a lack of trend information, randomness and periodicity, and demands for such forecasting represent a challenging problem [18]. Usually, time series cases where the sample length N is very small are not applicable for generating statistically reliable variants of forecasting. In this paper, we will focus on such time series and their forecasting. We will propose a new methodology that can be applied to irregular series. The methodology is applicable to all types and topologies of neural networks, or similar AI based forecasting methods, in order to improve their accuracy.

The usual step in development of forecasting ANN is to train many networks, while changing the number of neurons in some particular layer. The ANN with the most accurate forecasting wins. Nevertheless, if we observe the forecasts of all obtained networks, we can conclude that sometimes different networks predict different directions of the trend change in the next forecasting step. In this point, one cannot determine which one is the correct. This is particularly noticeable when dealing with volatile data series. Therefore, incorporating more than one network in the forecasting decision could make better predictions of the future events. The methodology that is suggested in this paper improves the forecasting accuracy of the ANN in the sense that it concurrently exploits several most accurate networks instead of the winner one. In this way, the forecasting accuracy can be significantly improved, as well as the confidence of the prediction. The performance of the proposed method is verified on an example of Serbian Gross National Income (GNI) data series, using Feed Forward Accommodated for Prediction (FFAP) neural networks' topology. The results demonstrate higher forecasting accuracy compared to individual FFAP networks.

The rest of the paper is organized in the following manner. In Section 2, the structure of the FFAP neural network topology is presented. The section that follows describes in detail the concurrent best-performance-based methodology for increasing the accuracy of the short time series FFAP forecasting. Three case studies are performed and analyzed: Serbian Gross National Income time series, the municipal traffic flow for a particular observation point, and the daily based electric load consumption time series; the forecasting results of the proposed methodology and other state-of-the-art forecasting techniques and their combinations are given in Section 4. The obtained results are discussed in Section 5, while conclusions are summarized in the last section.

2. FFAP Neural Networks

In general, neural network-based computational and forecasting methods developed from the desire to reveal, realize, and emulate the capability of the brain to process information [14]. The entire brain is composed of many neural networks that receive information from the surroundings, extract and recombine their relevant parts, and make the decisions about the needs of the organism. Artificial neural networks (ANN) emulate such abilities of the brain in order to realize complex nonlinear input-output transformations.

Consider a time series denoted by y_i, $i = 1, 2, \ldots, m$. It is a set of observables of an undefined function $\widehat{y} = \widehat{f}(t)$, that are taken at regular time intervals Δt, where $t_{i+1} = t_i + \Delta t$. In the forecasting process, the historical data are used to determine the direction of future trends, while one-step-ahead forecasting implies the mathematical search for such a function f that can accurately perform the following mapping:

$$y_{m+1} = f\left(t_{m+1}\right) = \widehat{y}_{m+1} + \varepsilon, \tag{1}$$

where the term $\widehat{y}_{m+1}$ represents the desired response, while ε is the acceptable error.

In the past decades, ANNs have been developed as a tool that has great capabilities for recognizing and modelling data patterns that are not easily identifiable by traditional methods. However, one may notice a common feature in all existing ANN applications in forecasting. It is the necessity for a relatively long time series in order to achieve high accuracy. Usually, there should be at least 50 data points to consider [19]. Because of this and due to previous research in short-term forecasting [20–22], we have chosen the FFAP neural network topology, as a base to be used throughout this study. This structure will briefly be explained next.

General structure of a feed forward neural network is illustrated in Figure 1. It has just one hidden layer, since it is confirmed to be sufficient enough to solve univariate forecasting problems [23]. In this figure, indices "in," "h," and "o" denote input, hidden, and output layers of the ANN, respectively. Weights are labeled with $w(k, l)$, where connections of the input and the hidden layer are designated with $k = 1, 2, \ldots, m_{\text{in}}$, $l = 1, 2, \ldots, m_{\text{h}}$, while connections of the hidden and output layer are designated with $k = 1, 2, \ldots, m_{\text{h}}$, $l = 1, 2, \ldots, m_{\text{o}}$. The thresholds are denoted with $\theta_{\text{x,r}}$, $r = 1, 2, \ldots, m_{\text{h}}$ or m_{o}, depending on the layer. The neurons in the input layer distribute the input signals, while neurons in the hidden layer are activated by a sigmoid function. Finally, linear function activates neurons in the output layer. A modified version of the steepest-descent minimization algorithm is applied as a learning method [24]. The problem of initialization was solved using the method described in [25].

In the case of short time series prediction problem, a set of observables (samples) is given (per time instant) meaning that only one input signal is available, the discretized time [18]. To enable mathematical solution for the forecasting problem, in most cases, both values for time variable and the response need to be transformed, as shown in the following equation:

$$t = t^* - t_0. \tag{2}$$

Having in mind that t^* stands for the time instant, this reduction gives the value of 1 to the time of the first sample (t_0). Samples are normalized in the following way:

$$y = y^* - M, \tag{3}$$

where y^* stands for the current value of the target function and M is a constant, selected so as to reduce the relative difference between output values, if necessary. When implementing the architecture in Figure 1, the following series would have to be learned: $(t_i, f(t_i))$, $i = 1, \ldots, m$.

Exploiting the basic topology shown in Figure 1, in [26, 27], better forecasting solutions were suggested for the problem of short prediction base period. This architecture is referred to as feed forward accommodated for prediction (FFAP) and depicted in Figure 2. The main idea during the FFAP architecture development was to force the neural network to learn the same mapping several times simultaneously but shifted in time. In that way, it is supposed that previous responses of the function will have larger influence on the $f(t)$ mapping. This also forces the network to identify complex intertwined deterministic relations existing between phenomena that influence the observed variable.

The FFAP architecture is depicted in Figure 2. The input set (t_i) is brought to the input terminal. The future terminal at the $Output_3$, approximates y_{i+1}. $Output_3$ may also be seen as a vector when a multiple-step prediction is required. The present value y_i is obtained at terminal $Output_2$. Finally, $Output_1$ should learn the past value, i.e., $y_{i\text{-}1}$. Although it is not explicitly stressed out, this may also be seen as a vector of past values of the response (since multiple samples from the past responses were used).

Figure 1: Basic fully connected feed forward neural network (one hidden layer and multiple outputs in the output layer).

Figure 2: FFAP—feed forward accommodated for prediction ANN structure.

We can express the functionality of the network as

$$(y_{i+1}, y_i, y_{i-1}, y_{i-2}, y_{i-3}) = f(t_i), \quad i = 4, \ldots, m, \tag{4}$$

where $\text{Output}_1 = \{y_{i-1}, y_{i-2}, y_{i-3}\}$, meaning that one future, one present, and three previous responses are to be learned.

Our task here was to do one-step-ahead prediction. Using the already predicted value as input data for multistep-ahead prediction leads to accumulation of the prediction error what we demonstrated in [21].

In this way, an efficient network topology is created, that uses in the test phase, only matrix w_{in} in combination with single column/row of matrix w_{o}, and the outputs corresponding to different moments, for evaluation of weighs in w_{in} and thresholds in hidden layer.

3. Concurrent Best-Performance-Based Prediction Methodology

The methodology for increasing the accuracy and certainty of short time series ANN forecasting that we are suggesting is depicted in Figure 3. The aim of this procedure is to establish tools and procedure that will increase the accuracy of the existing individual forecasting FFAP networks, exploiting the best of them, in a concurrent manner.

Figure 3: Algorithm for obtaining high-accuracy predictions.

We start with a turbulent short time series, expressing a certain variable for a given period of time. In order to facilitate their learning, these data sets have to be properly adjusted (shifted and/or normalized). After that, we evaluate the effect of changing the number of neurons in the hidden layer of the FFAP on the forecast accuracy. We begin with generating and training the FFAP network that has 3 neurons in the hidden layer. The experiment repeats with increasing number of hidden neurons. The construction of networks ends when FFAP with 10 neurons in the hidden layer is obtained and trained. This gives 8 different neural networks.

When a single network is trained, it requires a minimal learning set of first 13 samples (in our case, this is shown to be enough) in order to create the first forecast at the output, that is the extrapolated value of the trend curve. Since this is still training, one can also calculate the first forecasting error for the particular network. By entering every further sample, the network can better learn and predict the trend and its change for the input variable. As already mentioned, this process repeats for the entire group of networks. At the end of this process, all networks are trained with the entire time series, representing the past and the present. The result of this process can be systematized in a form of a forecasting matrix, with columns representing the number of neurons in a particular network, while rows represent time instances for which the forecast were made. The number of columns goes from three to ten, while number of rows is equal to the number of input samples.

In the next step, we analyze the forecasting matrix and search for three topologies that have the best performance measures. The reason for choosing three topologies lies in the fact that the time series can be irregular. Two best networks can predict totally opposite trend change for the next predicting step.

Because no one can tell which one is more correct, we introduce the forecast of the third best network to perform a kind of arbitration. In this way, the confidence of the prediction as well as its accuracy can be improved. In this analysis, we calculate mean average error (MAE), mean square error (MSE), root-mean-squared error (RMSE), and mean absolute percentage error (MAPE), for each column, using equations (5)–(8). The least three values of a particular error correspond to three most accurate ANN forecasting topologies:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} \left| y_i - \hat{y}_i \right|}{N}, \tag{5}$$

$$\mathrm{MSE} = \frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)^2}{N}, \tag{6}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)^2}{N}}, \tag{7}$$

$$\mathrm{MAPE} = 100 \cdot \frac{\sum_{i=1}^{N} \left| \left(y_i - \hat{y}_i \right) / y_i \right|}{N}, \tag{8}$$

where N denotes the number of sample cases, y_i represents the individual predicted value, and $\hat{y}_i$ is the target value for the sample i.

The process ends by adopting the final prediction on a test set data calculated using three best topologies simultaneously. Four cases of concurrent forecasting were investigated. First, we simply calculate an average of the forecasts for three most accurate networks. Namely, if the two of three predictions are supporting each other in forecasting the future trend, they make the largest impact to the final prediction, decreasing the importance of the third that predicts the opposite change of the trend. In that way, none may be qualified as the better one. So, the average is the best representative.

Nevertheless, from our experience, creating a linear combination of three best forecasts, where weights or a multiplying factor for each network, corresponds to a reciprocal of its forecasting error, can further improve the forecasting accuracy. The one that was the closest to the correct value gets the largest weight in the equation for the final forecast, as shown in the following equations:

$$\begin{aligned} y_{\text{final}} &= w_1 \cdot y_{\text{f1}} + w_2 \cdot y_{\text{f2}} + w_3 \cdot y_{\text{f3}}, \\ w_1 : w_2 : w_3 &= \varepsilon_3 : \varepsilon_2 : \varepsilon_1, \\ w_1 + w_2 + w_3 &= 100\%, \end{aligned} \tag{9}$$

where ε can be of any error types that are calculated (MAE, MSE, or RMSE). In these equations y_{final} denotes the final forecast value, while y_{f1}, ε_1; y_{f2}, ε_2; and y_{f3}, ε_3 denote forecasts of the first, second, and third most accurate networks and their corresponding errors, respectively.

4. Case Studies

4.1. Prediction of the Serbian Gross National Income. Gross National Income (GNI) is defined as an estimate of the income from goods and services produced by an economy and received by a country both domestically and from abroad. This measure of the size of an economy is highly important and has large economic, political, and social implications. For politically and economically turbulent developing countries, it is very difficult to predict it due to mutual influence of many known or unknown factors. The methodology that can be considered for such predictions exploits artificial neural networks. This kind of time series appears superb for verifying the methodology that we propose. In this study, we will try enhancing the neural network approaches, described in Section 2, for short -term forecasting of the Serbian GNI, by decreasing the forecasting error and increasing the confidents of the predictions in the next term, using its historical data. The same data set will be used for ARIMA forecasting in order to create a feeling about the order of accuracy of the proposed methodology.

For the analyzed series, time is discretized at year long periods and reduced by 1989, as given by (2), while the value of the GNI is given in billions of Dollars, like in original data. In this case, there was no need for normalization of data. However, in our experience, these transformations can make the training process numerically better conditioned.

Although this time series covers a long period of time (1990–2017), the GNI is the economic variable that is obtained annually. It was first introduced by economic scientists in 1990s. Because of that, this time series can be considered as a very short. The time series data are obtained from the World Bank National Accounts data and OECD National Accounts data files.

After the initial training of 8 networks using the test set that contains GNI data for a period 1990–2012, we have calculated their performance measures, i.e., MAE, RMSE, and MSE. Three most accurate FFAP networks have 7, 9, and 10 neurons in the hidden layer. Corresponding weights were calculated based on the values of the networks' errors in order to be used in combined predictions of the test set (2013–2017). Values are listed in Table 1, while Figure 4 gives their graphical representation.

Table 2 contains prediction results obtained for 5 successive time instants of the test set, starting from 2013, as well as the true value of the GNI. Figure 5 visualizes three best GNI predictions from Table 2, while Figure 6 illustrates different GNI predictions, i.e., true, averaged, MAE-weighted, RMSE-weighted, and MSE-weighted values.

As a final measure of performances, for the proposed methodology, we have calculated MAE, MSE, RMSE, and MAPE for all GNI predictions of the test set for the period 2013–2017, and these are shown in Table 3. All types of errors obtained after applying our forecasting algorithm are better than any particular FFAP. For example, the best FFAP that has 7 neurons in the hidden layer had a RMSE for the entire predicted period of 6.85, while the best improvement in forecasting is achieved when MSE-weighted concurrent linear combination of three best FFAP is applied. RMSE in this case is 2.60. The error reduction in this case is almost 65%.

An economic variable that is very similar to the GNI is the GNI per capita. It is an economy measure that is obtained when the value of the GNI is divided by the number of residents in a country. It should be emphasized that values for the GNI and the GNI per capita show very similar trends for particular time interval in the case of the Republic of Serbia. This could be explained by the fact that the natality for the Republic of Serbia is varying very slow over time. The authors have performed the similar forecasting procedure on a GNI per capita time series for the same time period, and it has shown very similar forecasting performance measures.

To the best of our knowledge, analyzing GNI time series in the case of Serbian economy is the task that has not been studied or published yet, and no comparative analysis of the forecasting accuracies with some alternative forecasting methods can be conducted. To overcome this problem alternative state-of-the-art, i.e., autoregressive integrated moving average forecasting methodology (ARIMA) was performed on the same data set. The theoretical background on this topic as well as its implementation strategies can be found in the literature [28, 29].

The model fitting process for GNI training set will be very briefly explained here, since this task is out of the scope of this study. To achieve the adequate ARIMA(p, d, q) model, GNI series was tested for stationarity by applying the unit root tests: Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test and augmented Dickey–Fuller test (ADF). They indicated that the GNI series is nonstationary. Since the null hypothesis of nonstationarity is not rejected, the series needs to be transformed. After the first difference, the autocorrelation coefficients led to a conclusion that this new series is now stationary. Second differencing over the data led to over-differenced series and was discarded from further analysis. In this way, the order of I term, d, in ARIMA(p, d ,q), has been identified (here, 1). The analysis of autocorrelation

Table 1: Forecasting weight coefficients based on the networks' training set accuracy in GNI forecasting.

Weight coefficients	7 neurons, 1st most accurate	9 neurons, 2nd most accurate	10 neurons, 3rd most accurate
Averaged	33.33	33.33	33.33
MAE-based	40.63	31.03	28.34
RMSE-based	51.26	28.83	19.91
MSE-based	42.14	31.60	26.26

Figure 4: Combined weights of three best networks predicting GNI, corresponding to particular error types.

Table 2: GNI ANN forecasting for three most accurate FFAPs.

Year	True value	7 neurons	9 neurons	10 neurons
2013	43.64	56.45	30.17	15.42
2014	42.42	49.20	50.12	28.56
2015	35.33	38.33	42.30	34.45
2016	36.06	36.97	37.33	30.28
2017	38.55	34.73	55.25	35.57

Figure 5: GNI forecasting of three most accurate FFAP ANNs.

function ACF and partial autocorrelation function PACF confirmed the stationarity of differenced data set and helped in selecting the candidates for the best ARIMA model and determining whether the pattern of autocorrelation can be better explained by autoregressive AR terms, moving average MA terms or a combination of both. Selection of the best model, among few suitable, is achieved based on

Figure 6: Combined predictions of the GNI: true, averaged, MAE-weighted, RMSE-weighted, and MSE-weighted values.

Akaike's Information Criterion (AIC) and Schwarz Bayesian Information Criterion (BIC). The ARIMA(0, 1, 1) model with minimal value of BIC was selected as the best, expressing the smallest error variance.

In forecasting the Serbian GNI over the period of five years (2013–2017), using the obtained ARIMA(0, 1, 1) model, values for MAE, RMSE, MSE, and MAPE were 5.98, 6.52, 42.51, and 14.92, respectively.

We believed that it would be interesting to extend existing experiments with additional predictions that combine ANN weights with those obtained if an appropriate weight for linear regression of the observed time series is added into a final linear combination. We have constructed a new forecasting system that exploits the performance measures of both most accurate individual ANN topologies with most accurate linear regression model. According to their performances, new weights have been calculated in order to obtain modified forecasting linear combinations of four terms. The newly obtained forecasting results are shown in Figure 7.

Corresponding performance measures in GNI forecasting in the case of linearly combined best ANN topologies and their extensions with linear regression models, outperformed the traditional ARIMA modelling. This is systematized in Table 4.

4.2. Prediction of the Municipal Traffic Flow. We define the traffic flow as the number of vehicles that pass a particular observation point per unit of time (usually 15 minutes). There can be various ways to "measure" the volume of the traffic and various sources of data such as simulations, sensors, taxi GPS, floating cars, and similar. In short-term prediction, which is our goal, the traffic is predicted in the next moments (usually 15 minutes) on the basis of real-time historical data.

The traffic flow time series consists of about 70 measurements, collected in collaboration with the Faculty of Transport and Traffic Engineering of the University of Belgrade, for one of the Belgrade's busiest roundabouts. This time series can also be considered as short. Here, we assume that the traffic from the immediate past has the greatest impact on the future value which, in turn, is produced for a

Table 3: GNI forecasting errors of different ANN approaches.

Error type	7 neurons FFAP	9 neurons FFAP	10 neurons FFAP	Aver. FFAP	MAE weighted	RMSE weighted	MSE weighted
MAE	5.47	9.22	10.34	3.47	3.00	2.95	2.36
RMSE	6.85	10.68	14.36	4.78	3.70	3.20	2.60
MSE	46.92	114.02	206.23	22.84	13.68	12.27	6.77
MAPE	13.26	23.12	24.71	8.60	7.47	7.39	5.99

Figure 7: Combined ANN and linear regression models for predictions of the GNI: true, MAE-weighted, RMSE-weighted, and MSE-weighted values.

Table 4: GNI forecasting performances of different approaches.

Method	Error type			
	MAE	RMSE	MSE	MAPE
Single ANN	5.47	6.85	46.92	13.26
Combined ANN				
Averaged	3.47	4.78	22.84	8.60
MAE weighted	3.00	3.70	13.68	7.47
RMSE weighted	2.95	3.20	12.27	7.39
MSE weighted	2.36	2.60	6.77	5.99
Combined ANN-LIN.REG.				
MAE weighted	2.71	3.64	13.30	6.78
RMSE weighted	2.62	3.40	11.54	6.60
MSE weighted	2.24	2.51	6.29	5.73
ARIMA	13.26	23.12	24.71	8.60

moment in the near future. In that sense, a large series of consecutive values, we believe, can obscure the information needed for prediction. That stands especially for specific days such as state holidays (very low urban traffic) or football matches (very high urban traffic) for which the information older than several hours (at most 24 or so) has no significance. This is the reason for shortening the initial time series to 70 samples of interest.

Traffic data are accommodated and constant M (here, 140) selected so as to reduce the relative difference between output values. Although this transformation creates negative values in the training set, it is also the way to make the training process numerically better conditioned.

The initial training of 8 networks using the test set was performed. We have calculated their performance measures, i.e., MAE, RMSE, and MSE. Three most accurate FFAP networks have 5, 3, and 4 neurons in the hidden layer. Based on the values of the networks' errors, corresponding weights were calculated for the combined predictions of the test set. Combined weights of three best networks, corresponding to particular error types, are shown in Table 5, while Figure 8 gives their graphical representation.

Table 6 contains the prediction results obtained for 11 successive time instants. Previous instants were used for ANN trainings and its forecasting of the first sample in the table. This matrix also required 8 FFAP-ANNs to be trained with rising number of hidden neurons. The best performance measures are obtained for ANNs with 3, 4, and 5 neurons in the hidden layer. Figure 9 visualizes three best traffic flow predictions from Table 6, while Figure 10 illustrates different predictions of the traffic density, i.e., true, averaged, MAE-weighted, RMSE-weighted, and MSE-weighted values.

We have again calculated errors (MAE, MSE, RMSE, and MAPE), i.e., performance measures of most accurate networks and of the concurrent, performance-based predictions for the next 11 time instances, and these are shown in Table 7. All types of errors obtained after applying our forecasting algorithm are again better than any particular FFAP. For example, the best FFAP that has 5 neurons in the hidden layer had a MAE for the entire predicted period of 20.46, while the best improvement in forecasting is achieved when RMSE-weighted concurrent linear combination of three best FFAP is applied. MAE in this case is 16.44, which is the improvement of almost 20%.

We have again performed an ARIMA fitting process in order to get the best forecasting model. The ARIMA(8,1,2) model appeared to have minimal values of AIC and BIC.

In forecasting the municipal traffic flow for 11 samples from the test set with 95% confidence limits, using ARIMA(8,1,2) model, values obtained for MAE, RMSE, MSE, and MAPE were 47.20, 24.00, 575.83, and 389.60, respectively. Corresponding performance measures in the case of linearly combined best ANN topologies and their extensions with linear regression model performed better than the traditional ARIMA modelling, considering different types of forecasting errors. This is systematized in Table 8. Introducing the linear regression into the suggested ANN based forecasting model additionally improved the accuracy of the prediction. These predictions are shown in Figure 11.

4.3. Prediction of the Daily Electric Load Consumption. We define values of the electric load consumption as an average power consumed (in kWh) for a period of one day, at a particular suburban measuring point. Data for the implementation of the method are acquired from the EUNITE 2001 competition file [30]. The electric load consumption time series consists of about a hundred measurements and is considered short.

Table 5: Forecasting traffic flow weight coefficients based on the networks' training set accuracy.

Weights coefficients	5 neurons, 1st most accurate	3 neurons, 2nd most accurate	4 neurons, 3rd most accurate
Averaged	33.33	33.33	33.33
MAE based	35.34	33.73	30.93
RMSE based	39.07	34.33	26.60
MSE based	44.73	34.54	20.73

Figure 8: Combined weights of three best networks predicting traffic flow, corresponding to particular error types.

Table 6: Traffic ANN forecasting for three most accurate FFAPs.

Sample number	True value	3 neurons	4 neurons	5 neurons
1	12.00	13.20	16.27	−23.84
2	3.67	−26.15	9.22	1.16
3	2.67	26.61	−34.38	−25.58
4	−5.33	−92.84	−13.90	−34.12
5	−7.00	−2.21	26.67	−40.55
6	−12.67	−14.42	−7.51	−37.49
7	−11.67	−18.03	−25.44	−17.15
8	−19.67	−17.49	−22.77	3.46
9	−30.00	−59.27	−72.45	−55.44
10	−40.33	−94.37	−94.53	−49.01
11	−42.33	−37.72	−91.69	−33.72

Figure 9: Traffic flow forecasting of three most accurate ANNs.

Figure 10: Predictions of the traffic density: true, averaged, MAE-weighted, RMSE-weighted, and MSE-weighted values.

Using the training set data, 8 FFAP ANNs were trained with rising number of hidden neurons. The best performance measures are obtained for ANNs with 3, 5, and 7 neurons in the hidden layer. Combined weights of three best networks, corresponding to particular error types, are listed in Table 9, while their graphical distributions are shown in Figure 12.

Table 10 contains the prediction results obtained for 11 successive time instants of the test set. Figure 13 visualizes three best traffic flow predictions from Table 8 while Figure 14 illustrates different predictions of the consumption, i.e., true, averaged, MAE-weighted, RMSE-weighted, and MSE-weighted values.

For this forecasting process, we have introduced additional accuracy metric, i.e., maximal error of estimate-M [30, 31]. It can be determined using the following equation:

$$M = \max\left(\left|y_i - \hat{y}_i\right|\right), \tag{10}$$

where N again denotes the number of samples, y_i is the individual predicted value, and $\hat{y}_i$ is the true value for the sample i.

Calculated errors (MAE, MSE, RMSE, MAPE, and maximal error of estimate) for prediction of the test set with 11 time instances are shown in Table 11. Errors obtained after applying our forecasting algorithm are again better than particular ANNs. For example, the best ANN that has 3 neurons in the hidden layer had a MAE for the entire predicted period of 34.42, while the best improvement in forecasting is achieved when MSE-weighted concurrent linear combination of three best ANN is applied. MSE in this case is 29.72, which is the improvement of about 13%.

We have now performed a seasonal ARIMA fitting process in order to get the best forecasting model. The SARIMA(2, 0, 2) $(1, 1, 1)_7$ model had minimal BIC value.

In forecasting, the municipal traffic flow for 11 samples from the test set with 95% confidence limits, using SARIMA(2, 0, 2) $(1, 1, 1)_7$ model, values were obtained for MAE, RMSE, MSE, and MAPE, and maximal error of estimate were 28.27, 35.13, 1234.13, 42.38, and 85.18, respectively. Corresponding performance measures in the case of linearly combined best ANN topologies and their extensions with linear regression model did not outperform the traditional

Table 7: Traffic flow forecasting errors of different ANN approaches.

Error type	5 neurons FFAP	4 neurons FFAP	3 neurons FFAP	Aver. FFAP	MAE weighted	RMSE weighted	MSE weighted
MAE	20.46	23.38	22.32	16.60	16.53	16.44	18.17
RMSE	23.39	30.20	34.37	21.32	21.02	20.54	22.31
MSE	547.34	911.99	1181.01	454.45	441.86	421.84	497.88
MAPE	266.59	253.19	341.13	186.23	186.41	204.37	187.39

Table 8: Traffic flow forecasting performances of different approaches.

Method	Error type			
	MAE	RMSE	MSE	MAPE
Single ANN	20.46	23.39	547.34	266.59
Combined ANN				
Averaged	16.60	21.32	454.45	186.23
MAE weighted	16.53	21.02	441.86	186.41
RMSE weighted	16.44	20.54	421.84	204.37
MSE weighted	18.17	22.31	497.88	187.39
Combined ANN-LIN.REG.				
MAE weighted	10.21	13.00	168.67	102.84
RMSE weighted	10.22	12.39	153.41	98.40
MSE weighted	9.67	11.35	128.84	105.95
ARIMA	47.20	24.00	575.83	389.60

Figure 11: Combined ANN and linear regression models for predictions of traffic flow: true, MAE-weighted, RMSE-weighted, and MSE-weighted values.

Table 9: Electric load consumption forecasting weight coefficients based on the networks' training set accuracy.

Weights coefficients	3 neurons, 1st most accurate	5 neurons, 2nd most accurate	7 neurons, 3rd most accurate
Averaged	33.33	33.33	33.33
MAE based	34.12	33.99	31.89
RMSE based	35.13	33.91	30.95
MSE based	36.93	34.41	28.66

SARIMA modelling, considering different types of forecasting errors. This is systematized in Table 12. Introducing the linear regression into the suggested ANN-based forecasting model did not improve the accuracy of the prediction. These predictions are graphically represented in Figure 15.

5. Discussion

Results obtained using neural networks in predicting the GNI for the Republic of Serbia have multiple qualities.

Figure 12: Combined weights of three best networks predicting electric load consumption, corresponding to particular error types.

Table 10: Electric load consumption ANN forecasting for three most accurate FFAPs.

Sample number	True value	3 neurons	5 neurons	7 neurons
1	131.54	149.29	138.41	146.25
2	107.41	136.26	145.27	143.92
3	65.44	100.88	97.84	104.24
4	115.08	63.43	78.41	80.12
5	118.29	24.84	29.32	82.51
6	122.52	69.39	54.85	206.74
7	92.98	109.1	92.36	91.51
8	51.89	88.85	141.45	145.32
9	50.12	51.37	40.14	9.38
10	54.35	91.64	57.9	78.68
11	88.39	81.68	59.03	88.35

Specifically, the use of mathematical methods and models in the prediction of future economic trends in the Balkan region at the Southeast Euro zone, encompassing the Republic of Serbia, is very ungrateful. GNI is particularly vulnerable to turbulent changes and numerous of non-economic factors to a great extent. Observing different types of forecasting performance measures in suggested ANN methodology just confirm the quality of the achieved results.

The results in predicting the movement of the GNI are of extreme importance for candidates for full membership in the European Union, such as Serbia. The size of the GNI determines the obligations of membership in this integration as well as the benefits that can be granted from numerous funds. The EU budget is financed by its own system of resources whose amount is limited to 1.23% of EU GNI (for the period 2014–2020). These funds are to be

Figure 13: Electric load consumption forecasting of three most accurate ANNs.

Figure 14: Predictions of the electric load consumption: true, averaged, MAE-weighted, RMSE-weighted, and MSE-weighted values.

Table 11: Electric load consumption forecasting errors of different ANN approaches.

Error type	3 neurons FFAP	7 neurons FFAP	5 neurons FFAP	Aver. FFAP	MAE weighted	RMSE weighted	MSE weighted
MAE	34.42	36.82	36.68	30.67	30.55	30.53	29.72
RMSE	42.31	46.35	48.02	38.18	38.05	38.00	37.28
MSE	1789.95	2148.44	2305.90	1457.48	1448.13	1443.75	1389.59
MAPE	39.00	49.23	44.11	39.66	39.57	38.43	39.52

Table 12: Electric load consumption forecasting performances of different approaches.

Method	Error type				
	MAE	RMSE	MSE	MAPE	Maximal error
Single ANN	34.42	42.31	1789.95	39.00	93.45
Combined ANN					
Averaged	30.67	38.18	1457.48	39.66	73.15
MAE weighted	30.55	38.05	1448.13	39.57	72.93
RMSE weighted	30.53	38.00	1443.75	38.43	72.85
MSE weighted	29.72	37.28	1389.59	39.52	72.49
Combined ANN-LIN.REG.					
MAE weighted	35.49	41.59	1729.46	50.78	87.10
RMSE weighted	35.83	41.94	1759.12	51.42	87.45
MSE weighted	37.58	43.49	1897.06	54.26	87.63
SARIMA	28.27	35.13	1234.13	42.38	85.18

filled from the budged of each member country with the amount of 0.73% from its own GNI value. With the amount of about 80 billion Euros per year, these resources represent the largest source of budget revenue, reaching 69% of all revenues. On the other hand, each member can expect from the EU budget the maximal amount of 4% of its national GNI. In some situations, a small difference in the amount of the GNI can significantly change the ratio of

Figure 15: Combined ANN and linear regression models for predictions of daily electric load consumption: true, MAE-weighted, RMSE-weighted, and MSE-weighted values.

liabilities and benefits of new members. Future GNI values are also important for creating a budget revenues and expenditures, due to restrictions in monetary and fiscal policy. Finally, the design of GNI forecasting system is crucial for making decisions on large investment projects because one gets a realistic picture of the capacity of the national economy. This reduces risks of illiquidity and even insolvency.

On the other hand, with the intention to preserve sustainable future, the importance of prediction of local traffic in large cities comes in for many reasons such as environmental and pollution monitoring; fuel usage reduction; journey planning; traffic control; urban planning; real-time route guidance; and ITS (intelligent transport system). In this analysis, ANN-based forecasting systems were developed enabling prediction of travel times, travel speeds, and traffic volumes on transportation networks using historic and real-time data.

At the end, we can also conclude that electric power load forecasting is the foundation of planning, development, and the assurance of operation efficiency and reliability of electric power systems. Because of the inherent characteristics of uncertainty, randomness, and nonlinearity, the load forecast has always been a forefront and hot issue. In the case of this forecasting task, we have achieved results using different types of forecasting performance measures in treated ANN models in order to confirm their quality.

6. Summary and Conclusion

In this paper, a novel methodology for increasing the predictions accuracy of different ANN-based systems has been suggested. Throughout analysis of three different time series of important everyday parameters, we have introduced some efficient improvements for prediction of short time series. The proposed method has been verified on GNI forecasting at national economy level, municipal traffic volume forecasting, and suburban daily electric load consumption forecasting. ANN-based models have been trained, and the performance of the models has been analyzed by applying various performance evaluation criteria and statistical tests that included MAE, RMSE, MSE, MAPE, and maximal error of estimate. Based on their accuracy, best performing ANN topologies, considering number of the neurons in the hidden layer, have been selected and implemented into a new forecasting system that linearly combines the forecasts of most accurate individual networks. More accurate one has been assigned a greater weight value in the linear combination. The same forecasting accuracy tests have been repeated for a test set data in order to conclude which model is superior. We have concluded that results of these three case studies reveal that linear combination of three most accurate ANN forecasts could predict trend of the future changes more accurately and with more confidence and that in most cases outperforms individual ANN forecasts, ARIMA forecasts, and hybrid ANN-linear regression forecasts. Moreover, we have determined the that accuracy improvement in these three particular cases ranges from 13% in the case of the electric load prediction and up to 65% in the case of Serbian GNI prediction. Based on the analysis presented in the paper, we can anticipate that the applicability of the method can be extended to other AI forecasting and modelling methods, as well as different types of ANN topologies. Our future research would also be oriented toward further error reductions and the discussion on the smallest number of the neurons in ANN layers in order to achieve this goal.

Data Availability

Data representing the Serbian GNI time series used to support this study are obtained from World Bank National Accounts data and OECD National Accounts data files. Data sets representing daily electric load consumption are obtained from the EUNITE 2001 competition file and are cited at relevant places within the text as references [30]. The rest of the time series data representing the municipal traffic flow are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was partly funded by the Ministry of Education and Science of Republic of Serbia under contract no. TR32004.

References

[1] K. Lawrence, R. Klimberg, and S. Lawrence, *Fundamentals of Forecasting Using Excel*, Industrial Press, Inc., New York, NY, USA, 2008.

[2] A. Abdullah, G. Suranegara, and D. Hakim, "Hybrid PSO-ANN application for improved accuracy of short-term load forecasting," *WSEAS Transactions on Power Systems*, vol. 9, pp. 446–451, 2014.

[3] S. Ilić, S. Vukmirović, A. Erdeljan, and F. Kulić, "Hybrid artificial neural network system for short-term load forecasting," *Thermal Science*, vol. 16, no. 1, pp. S215–S224, 2012.

[4] J. Wang, B. Zhou, and Z. Sheng, "Forecasting nonlinear chaotic time series with function expression method based on an improved genetic-simulated annealing algorithm,"

Computational Intelligence and Neuroscience, vol. 2015, Article ID 341031, 10 pages, 2015.

[5] W. Christiansse, "Short term Load forecasting using general exponential smoothing," *IEEE Transactions on Power Apparatus and Systems*, vol. 90, no. 2, pp. 900–910, 1971.

[6] N. Mohamed, M. H. Ahmed, S. Ismail, and Z. Ismail, "Improving short-term load forecasting using double seasonal arima model," *World Applied Sciences Journal*, vol. 15, no. 2, pp. 223–234, 2011.

[7] M. Ghiassi, D. K. Zimbra, and H. Saidane, "Medium term system load forecasting with a dynamic artificial neural network model," *Electric Power Systems Research*, vol. 76, no. 5, pp. 302–316, 2006.

[8] G. J. Tsekouras, E. N. Dialynas, N. D. Hatziargyriou, and S. Kavatza, "A non-linear multivariable regression model for midterm energy forecasting of power systems," *Electric Power Systems Research*, vol. 77, no. 12, pp. 1560–1568, 2007.

[9] T. Al-Saba and I. El-Amin, "Artificial neural networks as applied to long-term demand forecasting," *Artificial Intelligence in Engineering*, vol. 13, no. 2, pp. 189–197, 1999.

[10] C. H. Aladag, E. Egriolu, and C. Kadilar, "Improvement in forecasting accuracy using the hybrid model of ARFIMA and feed forward neural network," *American Journal of Intelligent Systems*, vol. 2, no. 2, pp. 12–17, 2012.

[11] B. Li, "Research on WNN modeling for gold price forecasting based on improved artificial bee colony algorithm," *Computational Intelligence and Neuroscience*, vol. 2014, Article ID 270658, 10 pages, 2014.

[12] D. Kown, M. Kim, C. Hong, and S. Cho, "Short term load forecasting based on BPL neural network with weather factors," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 1, pp. 415–424, 2014.

[13] P. Valtonen, S. Honkapuro, and J. Partanen, "Improving short-term load forecast accuracy by utilizing smart metering," in *Proceedings of the CIRED Workshop*, Lyon, France, June 2010.

[14] M. Karelson, D. A. Dobchev, O. V. Kulshyn, and A. R. Katritzky, "Neural networks convergence using physicochemical data," *Journal of Chemical Information and Modeling*, vol. 46, no. 5, pp. 1891–1897, 2006.

[15] D. Niu and Y. Wei, "An Improved short-term power load combined forecasting with ARMA-GRACH-ANN-SVM based on FHNN similar-day clustering," *Journal of Software*, vol. 8, no. 3, pp. 716–723, 2013.

[16] Y. Yang, J. Wu, Y. Chen, and C. Li, "A new strategy for short-term load forecasting," *Computational Intelligence and Neuroscience*, vol. 2013, Article ID 208964, 9 pages, 2013.

[17] H. Omar, V. H. Hoang, and D.-R. Liu, "A hybrid neural network model for sales forecasting based on ARIMA and search popularity of article titles," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 9656453, 9 pages, 2016.

[18] A. S. Mandel, "Method of analogs in prediction of short time series: an expert-statistical approach," *Automation and Remote Control*, vol. 65, no. 4, pp. 634–641, 2004.

[19] G. Zhang, B. Eddy Patuwo, and Y. Hu M., "Forecasting with artificial neural networks:," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[20] J. Milojković and V. Litovski, "Dynamic short-term forecasting of electricity load using feed-forward ANNs," *Engineering Intelligent Systems for Electrical Engineering and Communication*, vol. 17, no. 1, pp. 39–48, 2009.

[21] J. Milojković and V. Litovski, "Short-term forecasting in electronics," *International Journal of Electronics*, vol. 98, no. 2, pp. 161–172, 2011.

[22] M. Dimitrijević, J. Milojković, S. Bojanić, O. Nieto-Taladriz, and V. Litovski, "ICT and power: new challenges and solutions," *International Journal of Reasoning-Based Intelligent Systems*, vol. 5, no. 1, pp. 32–41, 2013.

[23] T. Masters, *Practical Neural Network Recipes in C++*, Academic Press, San Diego, CA, USA, 1993.

[24] Z. Zografski, "A novel machine learning algorithm and its use in modeling and simulation of dynamical systems," in *Proceedings of the 5th Annual European Computer Conference COMPEURO'91*, pp. 860–864, Bologna, Italy, May 1991.

[25] T. Denoeux and R. Lengellé, "Initializing back propagation networks with prototypes," *Neural Networks*, vol. 6, no. 3, pp. 351–363, 1993.

[26] J. Milojković and V. Litovski, "Comparison of some ANN based forecasting methods implemented on short time series," in *Proceedings of the 9th Symposium NEUREL-2008*, pp. 175–178, Belgrade, Serbia, September 2008.

[27] M. Dimitrijevic, M. Andrejevic-Stosovic, J. Milojkovic, and V. Litovski, "Implementation of artificial neural networks based AI concepts to the smart grid," *Facta Universitatis Series: Electronics and Energetics*, vol. 27, no. 3, pp. 411–424, 2014.

[28] T. H. Naylor, T. G. Seaks, and D. W. Wichern, "Box-jenkins methods: an alternative to econometric models," *International Statistical Review/Revue Internationale De Statistique*, vol. 40, no. 2, pp. 123–137, 1972.

[29] H. A. N. Hejase and A. H. Assi, "MATLAB-assisted regression modeling of mean daily global solar radiation in Al-Ain, UAE," in *Engineering Education and Research Using MATLAB*, IntechOpen, London, UK, 2011.

[30] C. C. Hsu, C. H. Wu, S. C. Chen, and K. L. Peng, "Dynamically optimizing parameters in support vector regression: an application of electricity load forecasting," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, vol. 2, p. 30, Kauai, HI, USA, January 2006.

[31] B.-J. Chen, M.-W. Chang, and C.-J. Lin, "Load forecasting using support vector machines: a study on EUNITE competition 2001," *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1821–1830, 2004.

Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 2186916, 8 pages
https://doi.org/10.1155/2019/2186916

Research Article

Product Module Network Modeling and Evolution Analysis

Hu Qiao,[1] **Zhaohui Xu**,[1] **Jiang He**,[1] **and Ying Xiang**[2]

[1]*School of Mechatronic Engineering, Xi'an Technological University, Xi'an 710021, Shaanxi, China*
[2]*College of Mechanical and Electrical Engineering, Shaanxi University of Science and Technology, Xi'an 710021, Shaanxi, China*

Correspondence should be addressed to Ying Xiang; yingcara@hotmail.com

Received 19 October 2018; Revised 21 January 2019; Accepted 7 February 2019; Published 6 March 2019

Guest Editor: Zoran Stamenkovic

Copyright © 2019 Hu Qiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Modular technology for product design and manufacturing is an effective way to solve mass customization problems. One difficulty in the application of modular technology is that the characteristics of mass customization, such as multi batch and small batch, easily increase the complexity of the module structure of the enterprise products. To address this problem, based on complex network theory, the enterprise products module is mapped as the vertex of the network, the number of modules used is mapped as the node weight, the dependency between the modules is mapped to the edge, and the product module network is established. The brittleness risk entropy of the product module network is put forward by considering the internal and external factors that influence the application of the enterprise module to determine the rationality of the required modules' organizational structures. Then, the stability uncertainty of the product module network can be determined by calculating the brittleness risk entropy, in which the subsystem that is the most brittle risk entropy can be identified. And the evolution of the product module network can be promoted by changing factors of the entropy maximum subsystem. To analyze the change in the product module network caused by module evolution, a BBV (Barrat–Barthelemy–Vespignani) model of the product module network is established to dynamically determine the brittle risk of the product module network. Finally, the modularity structure of a series of special vehicles is used as an example to verify the presented method, and the results confirm the rationality and effectiveness of the method.

1. Introduction

Modular design allows enterprises to offer diversified products to respond quickly to changing market demands by improving product diversity and reduce the complexity of the engineering system. Modular technology, as the most effective and widely used technology, has become the most important feature of product structure design [1].

Under mass customization, product modules are organized and managed in accordance with certain rules, and different types of modules are collected to form the module organization structure. The modular structure of complex products has many characteristics, such as various types of modules, decentralized module sources and unbalanced use of modules. It is a complicated system that affects and interacts with many kinds of factors. A variety of factors lead to instability of the module organization structure, such as random factors, including the outsourcing module arrival failure rate and low self-made module completion rate, and objective factors, including large module demand and high module usage rates. These factors will lead to a lack of timely module supply, delay product delivery, etc., and ultimately affect the stability of production. Therefore, considering various factors, the stability of the module organization structure is improved by changing the module type, improving the module structure, developing new modules, and so on. Consequently, the study of module evolution has important significance for engineering applications.

As the complexity and type of mechanical products increase, the type and quantity of product modules increase quickly. This is an important issue in guiding module evolution. The module organization structure within an enterprise is not a simple statistic of the number of different modules. To optimize the module organization structure, it is necessary to consider a series of problems, such as the basic relationship between modules, the number of modules, and the changes in modules. The two important considerations in realizing the evolution and optimization of module organization structure are as follows:

(1) How to establish the module organization structure model

Firstly, the module structure model must be established when researching module evolution. At present, design structure matrix [2, 3], graph theory [4], and complex network [5–7] are the main module organization structure modeling methods. Research methods describe the relationship between components and modules by establishing an undirected, directed, or weighted network model. However, most modeling objects are dependent on the product and do not consider the module organization structure of a variety of products. Using the same parts in the product family as the coincidence point and establishing multiproduct network models through product tree superposition have been proposed [8] to place products and components in the same network for research, but multilevel relationships among products, modules, components, and others coexist, which is not conducive to the evolution of research modules.

(2) How to ensure the stability of the module organization structure when researching the module evolution

Module evolution promotes changes in the modules' organization structure, and the direction of module evolution directly affects the optimization of the module organization structure. As a complex system, the stability of module organization is critical. In the field of system stability research, Fouad [9] first proposed the concept of power system vulnerability and established a weak analysis method based on a transient energy function and artificial neural network. In 2000, Albert researched brittle sources based on a complicated theory that brought system brittleness into a new age [10]. Wang et al. [11, 12] proposed a coupled map lattice model of cascading failure that provides a good mathematical means for cascading failure models of complex networks. Jin et al. [13] proposed system brittleness with entropy theory and system mutation theory. In addition, there have recently been new advances in the theory of complex network brittleness [14, 15] and applications [16] of complex network brittleness. However, compared with other complex systems, the propagation mechanism of brittle behavior, the brittle sources, and other factors differ due to the unique characteristics of module organization structure, and the relevant mechanism needs to be analyzed in depth. In addition, the dynamics of module evolution determine the characteristics of the dynamic changes in the system. System stability, as an important indicator of system optimization, needs to follow the dynamics of updating.

In this paper, a module evolution method is proposed to improve the stability of production to solve the problem of optimizing module organization structure for modular products. Complex network theory is used to map the module organization structure as a product module network with the characteristics of a scale-free network and small-world network by analyzing the network topology characteristics of the product module network model. By analyzing factors such as the arrival rate, completion rate, and utilization rate of the product module, the brittle risk entropy of the product module network model, that is, the uncertainty of the stability of the product module network, is established. The dynamic evolution model of the product module network is established to guide changes in the product module network caused by module evolution, and thus the goal of determining the brittleness risk of the product module network dynamically can be achieved. In addition, a series of special vehicles are presented to explain the effectiveness and reasonableness of the proposed method.

2. Product Module Network Model

2.1. Product Module Network Modeling. The modular product takes the module as the core and combines different modules to form a product. The combination relationship between two modules can be mapped as an edge, and thus any modular product can be represented using a network. As a part of the product, the module needs to be assembled to form a qualified product. The module combination has a direction order based on the order of assembly. One of the main features of modular products is reusability; that is, the same module can be used for a variety of products. It is possible to research the relationship between modules of multiple products in the same network because many reusable modules are used in a series of products.

Based on the module's usage order and reuse frequency in combination, the topology of the module organization structure is formed and defined as the product module network G (V, E, W):

V: nodes set, consisting of modules

E: directed edges set

W: weight, indicating the reuse number of the module

A series of special vehicles designed and produced by a special vehicle manufacturer are used as an example. The product module network G (V, E, W), which contain 132 nodes and 816 edges, is constituted by 6 types and 27 product modules. As shown in Figure 1, the network comprises the modules of multiple products in accordance with the combination relation, and the direction of connection edges between nodes is determined by the assembly order of the modules.

2.2. Topology Analysis of the Product Module Network. As one of the most basic static geometric vectors in a complex network, the degree distribution of nodes can reflect the macroscopic statistical characteristics of the network. Among the statistics of the product module network established in Section 2.1, the out-degree distribution function $P(k_{\text{out}})$ with the change in the out-degree node k_{out} is drawn in a double logarithmic coordinate system, the in-degree distribution function $P(k_{\text{in}})$ with the change in the in-

Figure 1: The process of product module network modeling.

degree node k_{in} is drawn, and the degree distribution function $P(k)$ with the change in the degree k is drawn (shown in Figure 2).

Figure 2 shows that the tail of the out-degree distribution function $P(k_{out})$ has a significant heavy-tailed phenomenon and is approximately a straight line. The whole function is consistent with the power-law distribution. The degree distribution $P(k_{in})$ is closer to a straight line and has a significant heavy-tailed phenomenon. The degree distribution $P(k)$ is closer to a straight line except for the initial point shifting, and the tail shows heavy tail traces consistent with the power-law distribution. Thus, the product module network conforms to the characteristics of a scale-free network, which means that the network distribution obeys an exponential distribution [17]:

$$P(k) \sim k^{-r}. \tag{1}$$

The average shortest distance of a random network with the same number of nodes and node averages as the product module network, which is expressed as L_r, is calculated by formula (2) [17], which also shows the result:

$$L_r = \frac{\ln n}{\ln K} = 2.6913. \tag{2}$$

The average cluster coefficient of a random network with the same number of nodes and node averages as the product module network, which is expressed as C_r, is calculated by formula (3) [17], which also shows the result:

$$C_r = \frac{K}{n} = 0.0465, \tag{3}$$

where K is the average degree of a network node.

Compared with the product module network, the average shortest path l (calculated by formula (2)) of the product module network is

$$l = 2.9316 \approx L_r. \tag{4}$$

The average shortest path c (calculated by formula (3)) of the product module network is

$$c = 0.3157 \geq C_r. \tag{5}$$

From formulas (4) and (5), there are a short average distance and a large average clustering coefficient in the product module network, which means that the product module network has the characteristics of a small-world network [17].

In a product module network, the out-degree reflects the module application range, the in-degree is the acceptance range of the module, the average shortest distance reflects the separation degree of the module, and the clustering factor reflects the criticality of the module. Thus, the product module network not only has the characteristics of a scale-

Figure 2: The node parameter statistical distribution of the product module network.

free network and small-world network but also has some of its own characteristics:

(1) In Figure 2, there are some nodes whose degree of out-degree or in-degree is 1, which indicates that these nodes can only be assembled to a specific module or only interface with a module. These modules are special modules of a single custom product, in accordance with the actual production.

(2) The degree distribution of the product module network is not uniform, and there are some hub nodes. A hub node indicates a few modules that are reused by multiple products in a product module network. The reliance on these modules is excessive, which indicates that the module is less diversified or has higher market popularity. Excessive reliance can easily lead to an inability of the product module network to bear the impact of node collapse.

(3) A scale-free network exhibits a high robustness to random faults and a high vulnerability to deliberate attacks. A similar situation occurs in the product module network, and therefore it is necessary to analyze the vulnerability of the product module network.

3. Product Module Network Brittle Analysis

For a complex system S, there is a subsystem or a part S_i that has a strong sensitivity to the environment. When S_i collapses because S_i suffers from a disturbance or attack by internal or external factors, other subsystems or parts might also collapse, potentially leading to collapse of the entire complex system. This behavioral characteristic of complex systems is called brittleness [18]. Brittleness is the basic property of complex systems that always exists and does not disappear with system evolution or environmental change.

If a module is a subsystem in a product module network, then a subsystem also has brittleness due to a low arrival rate, insufficient completion rate, and so on within the specified time. When the number of modules corresponding to the subsystem fails to support the production tasks corresponding to the product module network, the subsystem is abnormal, which is defined as subsystem collapse. The lower the effective supply rate, the greater the brittleness of the subsystem, if the module supply fault caused by subsystem brittleness is used as the characterization of the subsystem brittleness.

Assume that brittle event I_x in brittle event space $\mathbf{I} = \{I_1, I_2, \ldots, I_X\}$ that affects the arrival rate of subsystem S_i is a random event with probability p_x. The module arrival rate is defined as A_i:

$$A_i = 1 - \prod_{x=1}^{X} (1 - p_x), \tag{6}$$

where X denotes the total number of brittle events in brittle event space $\mathbf{I}$, $x = 1, 2, \ldots, X$, $0 \le p_x \le 1$.

Assume that brittle event J_y in brittle event space $\mathbf{J} = \{J_1, J_2, \ldots, J_Y\}$ that affects the completion rate of subsystem S_i is a random event with probability p_y. The module completion rate is defined as R_i:

$$R_i = 1 - \prod_{y=1}^{Y} \left(1 - p_y\right), \tag{7}$$

where Y denotes the total number of brittle events in brittle event space $\mathbf{J}$, $y = 1, 2, \ldots, Y$, $0 \le p_y \le 1$.

The number of modules can be ensured to meet the product module network requirements only if the outsourced module of the subsystem arrives on time and the self-made module completes on time. Therefore, the probability, (P_{si}), that the number of modules corresponding to subsystem S_i satisfies the requirements of the product module network is obtained:

$$P_{si} = 1 - (1 - A_i)(1 - R_i), \tag{8}$$

where $i = 1, 2, \ldots, n$; $A_i = 0$, the subsystem S_i contains modules that are self-made; and $R_i = 0$, the subsystem S_i contains modules that are outsourced.

If the number of modules of system S_i meets the requirements of the product module network, then the internal factor, that is, the module usage rate, which leads to product module network collapse, will be considered. The higher the module usage rate U_i, the greater the possibility of subsystem collapse due to scheduling failure, department collaboration, and other reasons.

Considering the arrival rate, completion rate, and usage rate, the brittleness measure function of subsystem S_i collapse caused by brittleness is defined as

$$f(S_i) = \frac{U_i}{P_{si}}, \tag{9}$$

where $i = 1, 2, \ldots, n$.

The probability of system S collapse is p_i, $0 \le p_i \le 1$, under the influence of subsystem S_i collapse. p_i is defined as the influence coefficient that the subsystem S_i collapse causes system $S = \{S_1, S_2, \ldots, S_n\}$ to collapse. The normalized value of the influence coefficient and collapse probability distribution of the subsystem are calculated. The utility coefficient q_i of subsystem S_i collapse is formulated as follows:

$$q_i = \frac{f(S_i)p_i}{\sum_{i=1}^{n} f(S_i)p_i}, \tag{10}$$

where $i = 1, 2, \ldots, n$, $0 \le q_i \le 1$, $\sum_{i=1}^{n} q_i = 1$.

According to Shannon theory [19], the brittle risk entropy of subsystem S_i can be defined as

$$G(S_i) = -q_i \log f(S_i), \tag{11}$$

where $i = 1, 2, \ldots, n$.

The mean value of the measure function of brittle events in the utility coefficient space is defined as the brittle risk entropy of the system, which is expressed as $H(S)$:

$$H(S) = -\sum_{i=1}^{n} q_i \log f(S_i) = -\sum_{i=1}^{n}\left(\frac{f(S_i)p_i}{\sum f(S_i)p_i}\right)\log f(S_i), \tag{12}$$

where $i = 1, 2, \ldots, n$.

According to the structure of the product module network, the usage rate of the module, which is expressed as U_i, can be mapped to the ratio of the output intensity to the total output intensity of a node in the product module network:

$$U_i = \frac{s_i^{\text{out}}}{\sum s_i^{\text{out}}}, \tag{13}$$

where s_i^{out} = output intensity of the ith node, $i = 1, 2, \ldots, n$.

The influence coefficient p_i of subsystem S_i collapse to system collapse can be mapped to the normalized value of the subsystem cluster coefficient:

$$p_i = \frac{c_i}{\sum c_i}, \tag{14}$$

where c_i = the cluster coefficient of the ith node, $i = 1, 2, \ldots, n$.

Thus, the brittle risk entropy of the product module network can be expressed by the following:

$$\begin{aligned} H(S) &= -\sum_{i=1}^{n} q_i \log f(S_i) \\ &= -\sum_{i=1}^{n}\left(\frac{(s_i^{\text{out}}/\sum s_i^{\text{out}})(c_i/\sum c_i)(1/P_{si})}{\sum(s_i^{\text{out}}/\sum s_i^{\text{out}}{}_i)(c_i/\sum c_i)(1/P_{si})}\right)\cdot \log\frac{s_i^{\text{out}}}{\sum s_i^{\text{out}}}\frac{1}{P_{si}}, \end{aligned} \tag{15}$$

where $i = 1, 2, \ldots, n$.

The brittle risk entropy reflects the brittle risk of the system at a moment and is the uncertainty measure of the possibility of system collapse [20]. From formula (11), the brittle risk entropy of the system is closely related to the output intensity of the network node and the node cluster coefficient. Therefore, the subsystem can be adjusted to improve the system performance and reduce brittleness risks.

Because of the different module types and sources of subsystems, there are also differences in brittle events. The uncertainty of the subsystem's brittle risk can be measured. If the risk entropy of subsystem S_k in system S is the largest, then,

$$G(S_k) = \max G(S_i), \tag{16}$$

where $i = 1, 2, \ldots, n$.

Multiple calls of formula (16) obtain several subsystems with large entropy in the system. The higher the brittle risk entropy, the poorer the grasp of the uncertainty of subsystem brittle risk. The brittle events in the subsystem with larger brittle risk entropy are analyzed to determine the main brittle factors that may lead to subsystem collapse. The subsystem brittle risk can be effectively reduced, and the system brittle risk will be reduced by controlling the main brittle factors.

4. Product Module Network Evolution

From formula (15), the brittleness of a subsystem comes from three aspects: arrival rate, completion rate, and usage rate. The arrival rate, completion rate, and usage rate of modules may be improved by adding a new module and deleting the old module. The brittle risk of a product module network subsystem is reduced by enhancing the control of the arrival rate, completion rate, and usage rate, gradually improving the arrival rate and completion rate and reducing the module usage rate.

In the product module network, the original network edge weight will change due to the addition of a new module. Module evolution is the evolution of the interaction between

network nodes, which is consistent with the common characteristics of the actual network.

From formulas (13) and (14), the output intensity and cluster coefficients of the product module network nodes have key functions for assessing the brittle risk entropy of the system.

In the product module network, the output intensity s_i^{out} of node N_i can be defined as

$$s_i^{\text{out}} = \sum_{j\in h} w_{i,j}, \tag{17}$$

where $w_{i,j}$ = the weight between two adjacent points and h = the number set of adjacent nodes that N_i points to.

Considering node N_i and the weight between two adjacent points that N_i points to, the output intensity s_i^{out} can reflect the influence of N_i. The output intensity of nodes indicates the usage frequency of N_i in the product module network.

The cluster coefficient c_i of the ith node in the product module network is

$$c_i = \frac{1}{s_i(k_i-1)} \cdot \sum_{j,k} \frac{(w_{i,j}+w_{j,k})}{2}(a_{ij}a_{jk}a_{ki}), \tag{18}$$

where s_i = the point intensity of N_i and k_i = the degree of N_i.

The cluster coefficient c_i is the weight relationship between the node and its surrounding nodes and can comprehensively reflect the importance of the node. In a product module network, a node's cluster coefficient expresses the importance of modules contained in the nodes.

According to formulas (17) and (18), the output intensity and cluster coefficient of the network node are calculated based on the edge weight. To adapt to the dynamic evolution process of the module, it is necessary to analyze the product module network dynamically.

The BBV (Barrat–Barthelemy–Vespignani) model is a network evolution model based on the point intensity driving and weight strengthening mechanisms. It can imitate changes in interaction intensity in a real system [21]. Here, the BBV model is used to describe and analyze the evolution of the product module network in a dynamic environment.

In a product module network that contains n nodes, when new node N_j is added to the network in a point-driven manner at each time interval, the weights of old node N_i and new node N_j will be allocated as

$$\begin{aligned} P_{j\longrightarrow i} &= \frac{s_i}{\sum s_j}, \\ w_{i,j}\big|_{\text{new}} &= w_{i,j} + \delta\frac{w_{i,j}}{s_i}, \end{aligned} \tag{19}$$

where s_i = the point intensity of N_i and δ = weight increment.

When N_j increases, the weight of N_j will have an increment δ. δ will be divided into all nodes connected to N_i based on weight.

The output intensity of N_i in the product module network can be modified as

$$s_i^{\text{out}} = \sum_{j\in h} w_{i,j}\big|_{\text{new}}. \tag{20}$$

The cluster coefficient of the ith node can be modified as

$$C_i = C_i\big|_{\text{new}}, \tag{21}$$

where $C_i|_{\text{new}}$ = the new cluster coefficient after weight change.

The objective function of solving the subsystem with the maximum entropy can be modified to the following:

$$G(S_k) = \max G(S_i)\big|_{\text{new}}, \tag{22}$$

where $i = j, 1, 2, \ldots, n$ and $G(S_i)|_{\text{new}}$ = new subsystem risk entropy after weight change.

Then, the change in the brittle risk entropy of the system is shown as

$$H(S) = -\sum_{i=1}^{n} q_i \log f(S_i)\big|_{\text{new}}, \tag{23}$$

where $i = j, 1, 2, \ldots, n$.

The new node weight information after network evolution can be obtained by using the BBV model, and the BBV model can adapt to the dynamic environment of the network evolution, dynamically determine which subsystem has the maximum brittle risk entropy in the product module network, and improve the grasp of the brittle risk of the product module network. In addition, the BBV model provides support for determining the brittle events of the subsystem and reducing brittle risk during network evolution.

5. Case

A series of special vehicles designed and produced by a special vehicle's manufacturer were tracked for 6 months using the above methods. The first 6 months of production tasks without this method were counted and analyzed according to production records. The effectiveness of the method was verified by comparing the effects before and after the application.

The monthly output in a production year and the node number changes of the product module network were obtained as shown in Figure 3.

In a production year, the product output of this series maintains 30–38 units/month. The maximum number of used modules is 156, and the minimum number is 136. Module types vary with monthly production in every month.

The brittle risk entropy of the first 6 months of the product module network is calculated by formula (15). Starting in the 7th month, the impact of the arrival rate, completion rate, and usage rate on subsystems with large entropy values is analyzed to promote and realize the evolution of the product module network by the method shown in Section 4, and the brittle risk entropy of the product module network in the next month is calculated. Finally, the brittle risk entropy of the product module network in a production year can be obtained, as shown in Figure 4.

Figure 3: The output and module type in every month.

Figure 4: The brittle risk entropy of the product module network.

The trend of the brittle risk entropy of the product module network shows that the brittle risk of the product module network declines before using this method (from the 1st to 6th month), with a maximum of 0.8136 and a minimum of 0.7880. However, the rate of decline is relatively slow, and there is even a slight increase in the 4th month.

Starting from the seventh month, the brittle risk entropy function of the subsystem is used to locate a subsystem with larger brittle risk entropy, and module evolution is performed based on the brittle event type. The brittle risk entropy of the product module network in the 7th, 8th, and 9th months declines rapidly from 0.7880 to 0.6616 because the subsystem with higher risk entropy is located precisely. This downward trend continues in the 10th, 11th, and 12th months.

Compared with Figures 3 and 4, the brittle risk entropy trend of the product module network in the first six months is consistent with the monthly output trend and the module type usage trend, which shows that the original brittle risk control is low. In the 4th month, there is high monthly output and large brittle risk entropy of the product module network. According to the production record, the number of product module changes increases from 5 in the 3rd month to 8 in the 4th month, which indicates that it is difficult for the product module network to resist the risk from production task changes.

From the 4th month to the 9th month, the monthly production tasks include 8 models. From the 4th month to the 6th month, the number of module types increases from 140 to 143, and the brittle risk entropy of the product module network decreases from 0.8012 to 0.7880. However, after introducing complex network theory, the number of module types increases from 143 in the 6th month to 154 in the 9th month, and the brittle risk entropy of the product module network decreases from 0.7880 to 0.6616. In addition, the evolution speed of the module is obviously improved, and the brittle risk entropy of the product module network is reduced rapidly, which indicates that the control ability of the product module network against brittle risk is enhanced.

The brittle risk entropy of the product module network slows in the 10th month, but there is a downward trend. According to production records, the number of product types decreased from 8 to 7 in the 10th month, but a new model was introduced. The brittle risk uncertainty of a new module may increase the brittle risk uncertainty of the product module network, which leads to slowing of the brittle risk entropy of the product module network. A similar phenomenon is observed in the 11th to 12th month, indicating a significant increase in the brittle risk resistance of the product module network and that a small number of external modules have a limited influence on the robustness of the product module network after effectively reducing the brittle risk entropy of the product module network.

Therefore, the effectiveness of the module evolution method based on brittle risk entropy analysis of the product module network is validated.

6. Conclusions

It is important to improve enterprise risk resistance and increase product diversity by optimizing module organization structure in production enterprises using modular technology. As a basic part of the product, there is a necessary relationship between the modules. As the product continues to be produced, the module usage of the product continues to accumulate. According to the relationship between modules and the accumulation of modules, a product module network can be established to study the module organization structure of the enterprise. Accordingly, the development law of the organizational structure of the enterprise module can be obtained to realize the optimization of the enterprise module organization structure.

In this paper, a product module network is built based on complex network theory. The network has small-world character and scale-free character according to an analysis of the topology characteristics of the network and shows uneven use of specific modules. The subsystem in the network with the maximum brittle risk entropy is obtained by establishing the brittle risk entropy function. Considering the brittle event source that leads to subsystem brittle risk, module evolution is put forward to reduce the brittle risk

entropy of the product module network, which shows that the ability to grasp the brittle risk of the network is improved. The BBV model promotes product module network evolution, and the change in the brittle risk entropy of the product module network can be grasped dynamically.

The product module network is built based on a series of products that use some of the same modules. The product module network can be set up separately to analyze the module organization structure in enterprises with multiple product lines. If all products of the enterprise have some of the same modules, then a full product module network can be built and the module structure is analyzed uniformly. The product module network of the whole product series may be large, which will affect the calculation efficiency and optimization effect because the construction of the product module network is based on the types of modules and the number of reused modules.

Data Availability

All data used in this manuscript come from a Chinese tank-truck manufacturer company. But according to the cooperation agreement, real data to the third part cannot be provided. If anyone who needs data to confirm the methods in this manuscript, please contact the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (Grant no. 51705392) and the Xi'an Technological University President Foundation (Grant no. XAGDXJJ16004)

References

[1] C. T. Ulrich and S. D. Eppinger, *Product Design and Develop. Yang Delin, transl. Dalian*, Dongbei University of Finance & Economics Press, Liaoning, China, 2001.

[2] A. Yassine, "An introduction to modeling and analyzing complex product development processes using the design structure matrix (DSM) method," *Urbana*, vol. 51, no. 9, pp. 1–17, 2004.

[3] M. Danilovic and T. R. Browning, "Managing complex product development projects with design structure matrices and domain mapping matrices," *International Journal of Project Management*, vol. 25, no. 3, pp. 300–314, 2007.

[4] H. Yu, J. Yu, S. Bi et al., "Configuration synthesis of reconfigurable robot based on graph theory," *Chinese Journal of Mechanical Engineering*, vol. 41, no. 8, pp. 79–83, 2006.

[5] I. P. Mcarthy, C. Tsinopoulos, P. Allen et al., "New product development as a complex adaptive system of decisions," *Journal of Product Innovation Management*, vol. 23, no. 5, pp. 437–456, 2006.

[6] A. Barrat, M. Barthelemy, R. Pastor-satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proceedings of the National Academy of Sciences*, vol. 101, no. 11, pp. 3747–3752, 2004.

[7] Y. Qin, L. Zhao, Y. Yao, and D. Xu, "Multistage machining processes variation propagation analysis based on machining processes weighted network performance," *International Journal of Advanced Manufacturing Technology*, vol. 55, no. 5–8, pp. 487–499, 2011.

[8] B. Fan, G. Qi, and Y. Ji, "Generation process of product family modularization based on complex network," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 40, no. 7, pp. 187–191, 2009.

[9] A. A. Fouad, Q. Qin Zhou, and V. Vittal, "System vulnerability as a concept to assess power system dynamic security," *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 1009–1015, 1994.

[10] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[11] X. F. Wang and J. Xu, "Cascading failures in coupled map lattices," *Physical Review E*, vol. 70, no. 5, pp. 56–113, 2004.

[12] X. F. Wang and J. Xu, "Cascading failures in scale-free coupled map lattices," *Physica A*, vol. 349, no. 3, pp. 685–692, 2005.

[13] H. Jin, W. Qi, J. Guo et al., *Brittle Theory and Application of Complex Systems*, Northwestern Polytechnical University Press, Xi'an, China, 2010.

[14] X. Fang, Q. Yang, and W. Yan, "Modeling and analysis of cascading failure in directed complex networks," *Safety Science*, vol. 65, pp. 1–9, 2014.

[15] Y. Zeng and R. Xiao, "Modelling of cluster supply network with cascading failure spread and its vulnerability analysis," *International Journal of Production Research*, vol. 52, no. 23, pp. 1–16, 2014.

[16] U. Soni, V. Jain, and S. Kumar, "Measuring supply chain resilience using a deterministic modeling approach," *Computers & Industrial Engineering*, vol. 74, no. 8, pp. 11–25, 2014.

[17] X. Wang, L. Xiang, and G. Chen, *Complex Network Theory and Application*, Tsinghua University Press, Beijing, China, 2006.

[18] H. Wang and H. Jin, "Britteness theory analysis of complex system by calculus on manifolds," *Journal of Information and Decision Science*, vol. 2, no. 1, pp. 1–6, 2007.

[19] G. Xu, *Systems Engineering*, Education Press of Shanghai Science and Technology, Shanghai, China, 2002.

[20] Y. Qin, L. Zhao, and Y. Yao, "Dynamic quality characteristics modelling based on brittleness theory in complex manufacturing processes," *International Journal of Computer Integrated Manufacturing*, vol. 24, no. 10, pp. 915–926, 2011.

[21] A. Barrat, M. Barthelemy, and A. Vespignani, "Weighted evolving networks: coupling topology and weight dynamics," *Physical Review Letters*, vol. 92, no. 22, pp. 1–4, 2004.

Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 8613639, 11 pages
https://doi.org/10.1155/2019/8613639

Research Article

Deep Learning Network for Multiuser Detection in Satellite Mobile Communication System

Guan Qing yang,[1,2] **Wu Shuang**,[1] **and He Ya-Ru**[1]

[1]*College of Engineering, Xi'an International University, Xi'an 710077, China*
[2]*College of Electronic and Information Engineering, Shenyang Aerospace University, Shenyang 110136, China*

Correspondence should be addressed to Guan Qing yang; guan_qy@163.com

Received 16 November 2018; Accepted 24 December 2018; Published 4 March 2019

Guest Editor: Zoran Peric

Copyright © 2019 Guan Qing yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A multiuser detection (MUD) algorithm based on deep learning network is proposed for the satellite mobile communication system. Due to relative motion between the satellite and users, multiple access interference (MUI) introduced by multipath fading channel reduces system performance. The proposed MUD algorithm based on deep learning network firstly establishes the CINR optimal loss function according to the multiuser access mode and then obtains the best multiuser detection weight through the steepest gradient iteration. Multilayer nonlinear learning obtains interference cancellation sharing weights to achieve maximum signal-to-noise ratio through gradient iteration, which is superior than the traditional serial interference cancellation algorithm and parallel interference cancellation algorithm. Then, the weights with multiuser detection through multilayer network forward learning iteration are obtained with traditional multiuser detecting quality characteristics. The proposed multiuser access detection based on deep learning network algorithm improves the MUD accuracy and reduces the number of traditional multiusers. The performance of the satellite multifading uplink system shows that the proposed deep learning network can provide high precision and better iteration times.

1. Introduction

Due to high-speed relative motion between mobile users and satellites in the satellite mobile communication system, different users access with the satellite at different elevation angles and multipath channel between satellite and user links is fading. These factors are creating obstacles for multiuser detection. In the case of limited bandwidth system, multiuser access detection (MUD) is an important issue in satellite mobile communication systems.

In the early literature, Cao and Viswanathan [1] proposed a method based on transformation of training sequences for single-user detection; the performance based on transforming training sequences is poor because the algorithm needs to add redundant information to make the signal transmission efficient. In the literature [2, 3], a soft iterative method was proposed for multiuser signal detection, but soft iteration required too much user information and it was not easy to achieve convergence.

Since multiusers accessed the satellite system at different elevation angles, different access carrier frequency offsets (CFOs) introduced multiuser access interference (MUI), so it was difficult to implement single-tap FDE to achieve multiuser detection. Some related research studies had been designed for multiuser detection. Tang and Heath [4] proposed a joint MUD scheme for MIMO. On this basis, in [5], CP was used to perform multiuser detection by accurately estimating the frequency offset. Zhang and Gao [6] proposed a blind scheme for multiuser uplink with large antennas. However, literatures [4–6] were proposed based on CP, and their solutions relied on searching for sampling. In this case, it suffered significant performance degradation, especially in the case of relatively large delay spread.

Blind user detection did not require a priori information, which effectively improves transmission efficiency. Therefore, Zhang and Gao [6] proposed a blind detection algorithm. At present, based on this, in [7, 8], a combination of advanced technologies, such as space-time processing and

interference cancellation, which improved transmission performance of the system, is proposed. Karakaya et al. [9] proposed an improved Kalman filter (KF) for multiuser detection, which requires a longer training sequence. Cao et al. [10] proposed LS-based algorithm (least squares) and MMSE (least mean square error) for multiuser interference, but they are not accurate. Chang et al. [11] proposed a multiuser detecting method, which needs cancel interference caused by carrier frequency with a high degree of complexity. A new algorithm based on time domain carrier frequency offset compensation algorithm was proposed in [9] for multiuser detection, but its accuracy was low. Cui in [10] proposed the joint iterative detection algorithm, which required a lot of matrix transposition operations and high complexity.

In recent years, literatures researched on the compressed sensing reconstruction algorithm for multiuser detection. Abebe and Kang [12] proposed an iterative sorting least squares (IORLS) algorithm for detecting multiuser signals. Based on this, orthogonal signal tracking (OMP) was used in [13] to reconstruct the signal for fast multiuser signal detection. An iterative support detection (ISD) algorithm had been proposed in [14], and a structured iterative detection (SISD) algorithm was proposed in [15] to recover multiple sparse signals. In [16], an approximate message delivery mechanism was proposed to reconstruct the signal. On this basis, Wei et al. [17] had proposed the application of this message for multiuser detection. On this basis, Donoho et al. [18] introduced expectation-maximization (EM) into multiuser detection and was named "joint-EM-AMP algorithm." This algorithm could achieve good bit error rate (BER) performance by jointly utilizing structured sparsity of prior information. Wang et al. [19] proposed a dynamic compression-aware algorithm for more practical scenarios.

For nonlinear transformation of single-layer network, the related literature also discussed and optimized in detail. For blind channel estimation of MIMO communication systems, AsadUllah et al. [20] proposed fuzzy logic-empowered opposite learning algorithm, which adopted mutant particle swarm optimization to obtain MMSE and BER performance. However, the single-layer nonlinear transformation of the algorithm reduces system performance.

Especially for the NOMA system, in literature [12, 21–23], a multiuser detection (MUD) algorithm was proposed. Since UEs of the NOMA system randomly transmitted data, they must perform blind activity detection. Wang and Yin [14] proposed compressed sensing (CS) algorithms. Wang et al. [24] proposed a modified version of the original ISD algorithm. Wang et al. [24] summarized multiuser detection method for the satellite mobile communication system. Literature [25] presents a joint ML-based CFO estimation method, but the complexity of this method was relatively high and was not suitable for satellite systems. In [26–28], the PIC algorithm for multiuser detection was proposed. Durand et al. [29] proposed the SIC algorithm based on weighting to detect multiuser in LTE-A systems, which is to improve the signal-to-interference ratio SINR. Kiayani et al. [30] proposed an improved PIC algorithm; the complexity was large since interference matrix transposition operation was very large, and the number of subcarriers was proportional.

For multilayer networks analysis, the related literature also discussed about CNN architecture. Yinghao et al. [31] proposed a multilayer neural network based on CNN for target recognition analysis, in order to deal with waste classification and obtain better resolution. Simulation results showed that the classification accuracy is higher than 90% under two different testing scenarios. Similarly, Albawi et al. [32] proposed a method for touch recognition, which was also implemented by CNN network. The proposed system outperformed other classification algorithms in terms of classification ratio. Simultaneously, the same CNN in [33] was also used in the Biomimetic Pattern Recognition to obtain a higher recognition rate. However, the above literatures all used the CNN network for visual feature recognition.

For the feature extraction of multilayer neural networks, Chao et al. [34] adopted a multilayer deep neural network with DBN architecture to acquire emotions through EEG signal acquisition. Compared with the CNN architecture, the DBN architecture was more suitable for processing transform domain features and achieving higher analysis accuracy. However, the algorithm was currently only for EEG signal analysis. Similarly, for multilayer neural network fusion decision, Wei et al. [35] proposed a weight-based fuzzy decision algorithm to achieve emotion recognition. The analysis was performed by multisource data decision fusion, which was including electroencephalography (EEG), electrocardiogram (ECG), respiration amplitude (RA), and galvanic skin response (GSR). However, the algorithm was used for data fusion direction.

2. System Model and Problem Formulation

2.1. Satellite-to-Ground Channel Model. If the shadow fading follows the Nakagami distribution, the Abdi star fading model was formed [36]. So, it is common for satellite-to-ground link model to use probability density functions, such as Rician model, Loo model, and Rician-Lognormal model. As a supplement to satellite-to-ground communication system, the satellite service scenario is mainly about wilderness area and open ground area, the shadow is less, and Clarke [37] also proved the correctness of this condition.

Some related literatures also have carried out studies on the satellite ground link channel model fitting through the measured data. Loo et al. [38], in view of the shadow effect on the signal propagation induced by trees, established the shadow effect model, but the application of the model has some limitations. Abdi et al. [39] established the urban road shadow (ERS) model for the satellite ground link with carrier frequency field between 870 MHz and 1500 MHz, and the model is suitable for the 1.5 GHz band channel fading distribution. In the literature [40], an integrated (CEFM) model is obtained by the integrated ERS model and the EFM model, which can be applied to a larger elevation range of 20 degrees to 80 degrees. Hess has used ATS-6 satellite to establish a small-scale and large-scale satellite ground link channel model within the city of 1200 km [41]. According to

the ESA satellite ground link access elevation data, ERS model is put forward in literature [42], and the transmission channel model of terrestrial mobile satellites is given in [43], the establishment of L-band elevation model of this band access for multiple users.

In this paper, the satellite-to-ground channel model is established using the measured results of the German Aerospace Research Center [42], and the multiuser access elevation model is established using the test data of the ESA (European Space Agency) in [37]. In this paper, the theory of tapped delay line channel model is used to establish the satellite-to-ground channel model. Figure 1 shows frequency selective channel for satellite-to-ground link model based on tapped delay line.

The specific method is to simulate the signal amplitude fading through the tapped delay line filter. Firstly, it is assumed that the scattering body is divided into several clusters, and the bandwidth of the signal transmission bandwidth is not resolved within each cluster. Then, the multicluster is used to model the satellite-to-ground link.

In the tapped delay line model, each tap represents a set of a plurality of delay paths with the same sum, but also the time delay path changes due to different flat fading amplitudes.

The tapped delay line model for satellite-to-ground link, multipath channel impulse response is composed of different delay characteristics; the channel modeling method is established for satellite-to-ground channel model following the tapped delay line.

Define $h(t,\tau)$ multipath propagation delay and time channel impulse response function, for different τ's; $h(t,\tau)$ is not related to each other. For determining the time delay τ, $h(t,\tau)$ is a stochastic process with mean complex Gauss time variation, and the impulse response $h(t)$ is with the amplitude characteristics of flat fading. Therefore, the time-varying impulse of the multipath channel can be expressed as follows:

$$h(t,\tau) = \sum_{l=0}^{L-1} b_l v_l(t)\delta(\tau-\tau_l), \tag{1}$$

where τ_l is the path of the transmission delay; $v_l(t)$ is a complex Gauss process; l is expressed as the path delay component, and the fading of the path is induced by the Doppler power and power spectrum; b_l is expressed as the delay coefficient, the square root of the value for the average lth path delay power.

$v_l(t)$ can be expressed in a delay interval at different incident angles' weighted path. The measured results with the German Aerospace Research Center are proposed in [36], which are about the rural environment, urban environment, suburban environment, with a signal carrier frequency of 1.82 GHz.

2.2. Multiuser Access Model. The Doppler shift caused by satellite motion is regular. At the same time, for defined mobile user, the Doppler shift is determined by the velocity of the high-speed motion and the elevation angle of the user. The Doppler shift introduced by high-speed satellite motion can be approximately equal.

Defining $x(n)$ as the transmitted signal, ξ as the frequency offset factor, and $h(n,l)$ as the impact response channel, the received $y(n)$ can be expressed as

$$y(n) = \sum_{l=0}^{L-1} x(n)h(n,l). \tag{2}$$

After introducing frequency offset interference, the received signal can be expressed as

$$y(n) = \sum_{l=0}^{L-1} x(n)h(n,l)\exp\left(\frac{j2\pi n\xi}{N}\right). \tag{3}$$

The receiving end performs N-point DFT demodulation on the received band offset signal, and after serial conversion, we could obtain frequency domain signal, which can be expressed as

$$Y(k) = X(k)H(k)C(0) + \sum_{\substack{l=0\\ l\neq k}}^{N-1} X(l)H(l)C(l-k), \tag{4}$$

where $H(k)$ is the channel frequency response and $C(k)$ is the interference introduced in the frequency domain. The first term indicates the part without carrier interference. The frequency offset causes change for amplitude and the rotation caused by carrier k. The second term is the intercarrier interference caused by the remaining subcarriers of carrier k. As can be seen from the above formula, when $\xi = 0$, $C(0) = 1$. This means, when the carrier frequency offset is zero, the interference term coefficient is 1, which means no interference would occur. This also shows that the interference between carriers depends on the relative frequency deviation and the serial number distance between subcarriers as the relative frequency offset interference factor increases.

When the carrier frequency offset is zero, the interference coefficient is 1. This shows that the interference between carriers depends on the relative frequency deviation and the serial number distance between subcarriers. As the relative frequency offset interference factor increases, $C(0)$ reduced interference to the received signal, but $C(l-k)$ increased interference to the received signal; intercarrier interference plays a major role where

$$\begin{aligned} C(l-k) = {} & \frac{\sin(\pi(l+\xi-k))}{N\sin((\pi/N)(l+\xi-k))} \\ & \cdot \exp\left(j\pi\left(\frac{N-1}{N}\right)(l+\xi-k)\right). \end{aligned} \tag{5}$$

Equation (5) also shows the energy leakage of subcarrier k to subcarrier l due to the frequency offset effect. The magnitude of the energy interference depends on the sequence difference of the carrier spacing and the relative carrier frequency offset factor. If the relative carrier frequency offset factor is e, then the relative energy leakage of the kth carrier to the lst carrier can be expressed as

Figure 1: Channel model for satellite-to-ground link based on tapped delay line.

$$\left|C_{k,l}\right|^2 = \frac{\sin^2\left(\pi\left[(l-k)+\xi\right]\right)}{N^2 \cdot \sin^2\left(\pi\left[(l-k)+\xi\right]/N\right)}. \quad (6)$$

If the interference signal with frequency offset is DFT transformed, it can be written as follows:

$$Y(k) = X(k) \cdot C, \quad (7)$$

where C is the carrier frequency offset interference matrix, which can be expressed as

$$C = \begin{bmatrix} c(0) & \cdots & c(N-1) \\ c(N-1) & \ddots & c(N-2) \\ \vdots & c(0) & \vdots \\ c(1) & \cdots & c(0) \end{bmatrix}. \quad (8)$$

Equation (8) can obtain some characteristics of the interference matrix C. The interference matrix introduced by the frequency offset is a Toeplitz-type matrix, in which each element of the matrix satisfies the periodic cycle property which can be expressed as

$$c(k) = c(k+N), \quad (9)$$

where $(\cdot)^{-1}$ is the inverse matrix and $(\cdot)^T$ is the transposed matrix conjugate, which can be expressed as

$$C^{-1} = C^*. \quad (10)$$

Defining $\Phi = \exp(j2\pi\Delta fT)$, the elements of the matrix can be written in recursive form:

$$c(k) = \Phi \cdot c(k-1) = \Phi^2 \cdot c(k-2) = \cdots = \Phi^i \cdot c(0). \quad (11)$$

Through the elemental analysis of the matrix, in the case of smaller frequency offset interference, the energy is mainly concentrated on the diagonal. The larger the frequency offset value, the more dispersed the energy, the larger the interference term, and the more the interference of the introduced ICI. Its energy distribution diagram is shown in Figures 2 and 3.

We could obtain from equation (11) that $\exp(j2\pi\xi m'/M)$ is the linear transformation of the introduced phase.

Figure 2: Matrix energy distribution introduced by small frequency offset.

Figure 3: Matrix energy distribution introduced by large frequency offset.

The number of set carriers is defined as 512, the channel bandwidth is 20 MHz, the Doppler shift is 15 kHz, and the signal mapping mode is QPSK. Figure 4 shows that the demodulated signal phase changes linearly with the increasing of subcarrier number.

For the multiuser uplink access, the interference comes from frequency offsets. At the same time, the larger the frequency offset range for each user, the more serious the multiuser interference. For the satellite transmission system uplink system, the access interference cancellation of each user is the key for uplink user detection.

3. Proposed MUD Algorithm Based on Deep Learning Network

The process of multiuser detection is divided into three parts. First, the multiuser signal is completed to cancel the

Figure 4: Linear phase introduced by frequency offset.

access interference and the multiuser access interference is reduced by establishing the optimal weight of the multilayer network. Secondly, through the multilayer network, the weight is iterated to obtain the optimal point. Through the network weight sharing and iteration of the first two parts, the optimal identification weight is finally obtained. Multiuser detection and identification is accomplished by optimally identifying the weight network.

3.1. Proposed Shared IC Algorithm. The proposed algorithm is based on the goal for optimizing CINR, which is to find the optimal CINR corresponding to WIC interference cancellation algorithm weights. Thus, multiuser interference signal received by satellite can be expressed as

$$Y^i = D^i H^i C_{i,i} + D^i H^i C_{i,k} + D^j H^j C_{i,j} + N, \quad k, i = 0, 1, \ldots, N-1, \; j = 0, 1, \ldots, M-1, \tag{12}$$

where $D_1^{(i)} = [y_1^{(i)}, y_1^{(i)}, \ldots, y_M^{(i)}] \in C^{NxM}$ and $H^{(i)} = [h_1^{(i)}, h_1^{(i)}, \ldots, h_M^{(i)}]$. Here, M is the number of subcarrier cancellations and w is the weight. $C_{k,l}$ is defined as the interference induced between subcarrier i of user j and carrier k of user i, which can be expressed as

$$C_{j,k} = \sum_{j=1}^{N-1} \sum_{j \neq k'}^{N-1} \frac{\sin\pi\left(j - k' + \xi^j\right)}{N \sin(\pi/N)\left(j - k' + \xi^j\right)} \cdot \frac{\exp\left(j\pi\left(j - k' + \xi^j\right)\right)(N-1)}{N}. \tag{13}$$

According to multicarrier allocation, this process can be a multiuser signal separation. Firstly, each user can be according to the traditional WIC algorithm for cancellation. For user i, subcarrier k, with the WIC algorithm, it can be expressed as

$$Y_m^i(k) = Y_{(m-1)}^i(k) - w\Lambda_{k,l} Y_{m-1}^i(l) - w\Lambda_{k,l} Y_{m-1}^j(l), \tag{14}$$

where $wC_{k,l}Y_{m-1}^j(l)$ is considered as the previous interference term from subcarrier l.

$Y_m^i(k)$ is defined as the mth interference cancellation signals for user i and subcarrier k, and $Y_{(m-1)}^i(k)$ is the $(m-1)$th r interference cancellation signals for user j. Getting formula (14) into the WSIC judgment, it could be obtained as the mth interference cancellation signals:

$$Y_2^i = D^i(k)H(k)\left(1 - wC_{k,l}^2\right) + \sum_{\substack{l=0 \\ l \neq k}}^{N-1} \sum_{l \in j} D^i(l)H^i(l)\left(C_{k,l}(1-w) - wC_{k,l}^2\right). \tag{15}$$

Idealizing it, we could obtain

$$\mathrm{CINR}^i \propto \left(1 - wC_{k,l}^2\right)^2. \tag{16}$$

Secondly, according to the traditional WPIC algorithm, each user can be for parallel cancellation interference, and it can be expressed as

$$Y_m^i = Y_{(m-1)}^i - wC_{i,l}Y_{m-1}^j, \tag{17}$$

where m is the iteration for WPIC interference cancellation and w is defined as the weight, $\Theta_{k,l}$ is considered as the interference, and $wC_{i,l}Y_{m-1}^j$ is considered as the previous interference term.

Y_m^i is the mth interference cancellation signals for user i and $Y_{(m-1)}^i$ is the $(m-1)$th interference cancellation signals for user j. Getting formula (13) into the WSIC judgment, it could be obtained as the mth interference cancellation signals:

$$Y_2^i = D^i(k)H^i(k)\left(1 - wC_{k,l}^2\right) + I, \tag{18}$$

where I is the second interference judgments and Y_2^i is the received signal after cancellation, which can be obtained as

$$I = \sum_{\substack{l=0 \\ l \neq k}}^{N-1} \sum_{l \in j} D^i(l)H^i(l)\left(C_{k,l}(1-w) - wC_{k,l}^2\right). \tag{19}$$

Due to different user access elevation angles, as well as the satellite ground link fading channel, introducing different serious Doppler frequency shifts is for the satellite-to-ground uplink system. For a particular user, because the tangential velocity of the satellite in a symbol is the same, the relative carrier frequency is kept constant, so the frequency offset introduced in a symbol period can be regarded as constant. Therefore, the normalized frequency offset factor can be considered to be constant.

The definition of satellite ground link uplink carrier number is 1024; the satellite suburban environment model is given in Table 1. Figure 5 shows that as the number of carrier interval sequence is reduced, energy leakage is more serious.

It can be obtained from formula (13) that a function of the value of the interference and the W weight value with IC algorithm, which can be expressed as a convex function. Signal-to-noise ratio can be obtained to the best value, when

the W weight is to the pole value. The proposed IC algorithm is based on the optimal weights, which can greatly reduce the number of iterations and improve the accuracy of the algorithm. The signal-to-interference ratio can be used by the comb or block pilot signal. By training the initial weights in the iteration, the optimal interference ratio can be obtained, which is close to the optimum.

The number of users is 4, and the number of subcarriers is 2048. In the condition of AWGN, SNR = 5 dB, the allocation is OFDM, and the frequency offset is 0.01, 0.05, 0.15, and 0.2. The multifading channel is shown in Table 1. The curve of the relationship between average CINR and weight w is shown in Figure 6.

The proposed multiuser detection algorithm is to optimize SINR through obtaining optimal cancellation weight; therefore, the IC algorithm can be divided into WSIC and WPIC algorithms; the WSIC algorithm is cancel interference for multiuser access according to each subcarrier, and the WPIC is the multiuser interference cancellation algorithm at the same time.

Figure 6 states that the convex function can reach the optimal value with w weight. Define $\mathrm{CINR}_{\mathrm{opt}}$ as the optimal after WIC algorithm cancellation. Since $\mathrm{CINR}_{\mathrm{opt}}$ is the convex function with w, the optimal CINR could be obtained when w has the extreme. Define the initial value as $w = 1$. In this case, the algorithm becomes the traditional SIC or PIC algorithm weight.

The improved algorithm is based on the WIC algorithm. Optimal weights are iterated to approximate initial weights. It is specific for obtaining the optimal weights below.

Defining E_{CINR} as the error between $\mathrm{CINR}_{\mathrm{opt}}$ and $\mathrm{CINR}_{\mathrm{out}}$, which after cancellation with the WIC algorithm can be expressed as

$$E_{\mathrm{CINR}} = \mathrm{CINR}_{\mathrm{out}} - \mathrm{CINR}_{\mathrm{opt}}, \tag{20}$$

$\mathrm{CINR}_{\mathrm{opt}}$ could be obtained with the Taylor expansion:

$$\mathrm{CINR}_{\mathrm{out}} \approx w_{\mathrm{opt}} + \alpha\left(w_{\mathrm{out}} - w_{\mathrm{opt}}\right), \tag{21}$$

where

$$\alpha \approx \left[\frac{\partial \mathrm{CINR}_{\mathrm{out}}}{\partial w_{\mathrm{out1}}}, \frac{\partial \mathrm{CINR}_{\mathrm{out}}}{\partial w_{\mathrm{out2}}}, \ldots, \frac{\partial \mathrm{CINR}_{\mathrm{out}}}{\partial w_{\mathrm{out}k}}\right]. \tag{22}$$

The influence of the satellite to ground link on this algorithm consists of two parts. Firstly, the influence for the proposed algorithm is also induced by the multiuser access angle differences. Due to differences in relative motion between the user and satellite, multiuser access interference has been generated, which significantly degrades the satellite system performance.

When multiusers access the same satellites, the multiuser access angle differences will introduce different carrier frequency offsets in the total number of the carrier system. Under certain conditions, the carrier frequency deviation will induce the different multiple access interference, including interference simulation as shown in Figure 6; the number of users is increasing, and the serious interference induced by frequency offset is larger.

Secondly, the influence for the proposed algorithm is induced by multipath fading. For urban simulation scenarios, the signal reflection effect caused by buildings is larger, and the diffraction effect caused by multipath is also larger. The more delay the received signal propagation, the more serious the signal fading is.

In the countryside scene, compared with the urban scene, the multipath number is decreased and the fading is relatively flat. This is because that the contryside scene is with a smaller number of building and weaker reflection and refraction.

3.2. Shared Weight Process and Feedback Solution Process. For multiuser received signals to cancel interference, we use shared weights to obtain the best weights and then obtain user detection and weight update.

The cost function established can be expressed as

$$\min_{x} \lambda_t \left\|w_t\right\|_1 + \left\|y_t - w_t x\right\|_2^2. \tag{23}$$

The cost function is to find that the accurate reconstruction, which should be realized. Then, the optimal weight detection error is made. Therefore, the sparse recovery for multiweight sharing can be obtained, and the optimal user detection for all the users can be satisfied in the following:

$$\min_{x} \lambda_t \left\|w_t\right\|_{1,w} + \left\|y_t - w_t x_t\right\|_2^2, \tag{24}$$

where

$$\|w\|_{1,w} = \sum_{i=1}^{K} w_i \left|x_i\right|. \tag{25}$$

The first term is a nonzero regular term whose position is known and is different from the traditional mode of all cost functions.

In addition to the weight constraint cost, the regularization constraint can be established for the corresponding x.

Therefore, when the measurement data are very small, x would become larger. For solving the problem, we used the improved solutions to solve the regularization. The modified cost function is expressed as

$$\min_{x} \lambda_t \|w\|_{1,w} + \alpha_t \left\|x - \widehat{x}_t\right\|_2^2 + \left\|y_t - w_t x\right\|_2^2. \tag{26}$$

The augmented Lagrangian is expressed as

$$L\left(x, z, t_t\right) = \alpha_t \left\|x - \widehat{x}_t\right\|_2^2 + \left\|y_t - w_t x\right\|_2^2 + t_t \left\|w - \widehat{w}_t\right\|_2^2. \tag{27}$$

The scaled problem (6) consists of three iterations:

Shared weight updates: x, w, and u.

$$\begin{aligned}
x_t^{[k+1]} &= \operatorname*{argmin}_{x}\left(\alpha_t \left\|x - \widehat{x}_t\right\|_2^2 + \left\|y_t - w_t x\right\|_2^2 + \beta \left\|w - w_t^{[k]} + u_t^{[k]}\right\|_2^2\right), \\
w_t^{[k+1]} &= \operatorname*{argmin}_{w}\left(\left\|w - \widehat{w}_t\right\|_2^2 + \beta \left\|w_t^{[k+1]} - w_t^{[k]} + u_t^{[k]}\right\|_2^2\right), \\
u_t^{[k+1]} &= u_t^{[k+1]} + x_t^{[k+1]} - w_t^{[k+1]}.
\end{aligned} \tag{28}$$

Table 1: Suburban environment parameter.

Tap	Distribution function	Parameter	Parameter distribution	Numerical value (dB)	Time delay (ns)
1	LOS : Rician	Rice factor	K	9.7	0
2	Rayleigh	Average multipath power	$2\sigma_l^2$	−23.6	100

Figure 5: Multiuser detection system block diagram.

4. Experimental Classification Results and Analysis

Establishing that the orbital altitude is 1100 km, the rural environment measured data proposed in [36] are satellite-to-ground uplink system model as in Table 2 in this paper.

Set the satellite beam spot beam of number 5 with a coverage diameter of 450 km. Satellite-to-ground link model with L-band carrier frequency in rural environments is shown in Table 2. Set the number of uplink users 4 and transmit pilot block type information. For analysis, the user has access to a maximum height at 35°, 25°, 5°, and 15°. The signal bandwidth is 50 MHz, the number of subcarriers is 1024, and the signal mapping method is defined as QPSK, BPSK, 16QAM, OFDM signal.

Figure 7 shows the BER simulation results for multiuser detection based on the WIC algorithm. Compared with the iterative sorting least squares (IORLS) algorithm proposed in [12] and the orthogonal signal tracking (OMP) algorithm proposed in [13], the WIC algorithm is improved by two iterations for sharing weight. The simulation results can be obtained from Figure 7. In first iteration, the multiuser detection algorithm performs better than the WPIC algorithm. Through the optimization iteration, the weight converges to the bump function close to the inflection point, and we can obtain the optimal multiuser detection and thus obtain the best BER performance for multiuser detection. Since the proposed MUD algorithm is superior to the improved conventional algorithm, interference cancellation improves the complexity. The system is enhanced under the condition of $10e-3$ BER.

Figure 8 shows the CINR performance comparison between the MUD algorithm proposed in [12] and the proposed MUD algorithm, since the improved MUD algorithm cancels subcarrier frequency offset interference and CINR system can be optimal.

Figure 9 shows different BER curves for each uplink user at SNR = 5 dB. As the relative interference carrier frequency offset increases, the BER of user error rate increases, which is due to the multiuser access system. And its performance of the algorithm is similar to the traditional algorithm, which is due to interference introduced by carrier frequency offset. After

Figure 6: CINR performance at different weights in the IC algorithm (including WSIC and WPIC) with three-layer network in two iterations.

Table 2: Channel model parameter in rural environment.

Tap	Distribution function	Parameter	Parameter distribution	Numerical value (dB)	Time delay (ns)
1	LOS : Rician	Rice factor	K	6.3	0
	nLOS : Rayleigh	Average multipath power	$2\sigma_l^2$	−9.5	
2	Rayleigh	Average multipath power	$2\sigma_l^2$	−24.1	100
3	Rayleigh	Average multipath power	$2\sigma_l^2$	−25.2	250

Figure 7: BER curves under different SNRs.

multilayer iteration, the weight is close to the optimal value at this time, and then the system is close to the maximum SINR optimization, which can be obtained as an optimization.

Figure 10 is the probability of correct classification of proposed deep learning network at different SNRs, which is also based on different length curves for each user. As the relative interference carrier frequency offset increases, the PCC of user error rate increases, which is due to the user access to the multiuser access system. The performance of the improved traditional multiuser detection algorithm at large freuqency offset is poor, which is due to residual carrier frequency offset. When training length is larger, the system is close to the maximum SINR optimization, which can be obtained as an optimization.

5. Conclusion

A multiuser detection algorithm based on deep learning network has been proposed. The proposed deep learning network for MUD could provide high precision and lower iteration times, which firstly establishes the CINR optimal loss function according to the multiuser access interference mode and then obtains the best multiuser detection weight through the steepest gradient iteration. The important feature of the proposed algorithm is through nonlinear optimal direction learning and to achieve maximum signal-to-noise ratio through gradient iteration, and then share weights. Through establishing a typical satellite communication system simulation platform, compared with the OMP and IORLS algorithms, the proposed deep learning network algorithm has

Figure 8: CINR performance of different subcarriers.

Figure 9: BER performance of different normalized frequency offsets.

better performance in different conditions of SNR, CINR, and carrier frequency offset interference.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Figure 10: Probability of correct classification of the proposed deep learning network for multiuser access to satellite at different length curves.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61501306), the Natural Science Foundation of Liaoning Province of China (no. 2015020026), and the Natural Science Foundation of Liaoning Provincial Department of Education (no. L2015402).

References

[1] L. Cao and R. Viswanathan, "Divergence-based soft decision for error resilient decentralized signal detection," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 5095–5106, 2014.

[2] A. Gomaa, L. M.-A. Jalloul, and A. Jalloul, "Efficient soft-input soft-output MIMO chase detectors for arbitrary number of streams," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1084–1088, 2015.

[3] S. Park, B. Shim, and J. W. Choi, "Iterative channel estimation using virtual pilot signals for MIMO-OFDM systems," *IEEE Transactions on Signal Processing*, vol. 63, no. 12, pp. 3032–3045, 2015.

[4] T. Tang and R. W. Heath, "A space–time receiver with joint synchronization and interference cancellation in asynchronous MIMO-OFDM systems," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 2991–3005, 2008.

[5] J. Chen, Y.-C. Wu, S. Ma, and T.-S. Ng, "Joint CFO and channel estimation for multiuser MIMO-OFDM systems with optimal training sequences," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 4008–4019, 2008.

[6] W. Zhang and F. Gao, "Blind frequency synchronization for multiuser OFDM uplink with large number of receive

antennas," *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2255–2268, 2016.

[7] B. Tang and J. Tang, "Joint design of transmit waveforms and receive filters for MIMO radar space-time adaptive processing," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4707–4722, 2016.

[8] T. Allen and N. Al-Dhahir, "Performance analysis of a secure STBC with coherent and differential detection," in *Proceedings of IEEE Conference on Wireless Communications and Networking Conference (WCNC)*, pp. 522–527, New Orleans, LA, USA, March 2015.

[9] B. Karakaya, H. Arslan, and H. A. Curpan, "Channel estimation for LTE uplink in high doppler spread," in *Proceedings of IEEE International Conference on Wireless Communications and Networking Conference*, pp. 1126–1130, Dalian, China, 2008.

[10] Z. R. Cao, U. Tureli, and Y. D. Yao, "Analysis of two receiver schemes for interleaved OFDMA uplink," in *Proceedings of IEEE International Conference on Signals, Systems and Computers*, pp. 1818–1821, Monterey, CA, USA, November 2002.

[11] Y.-K. Chang, F.-B. Ueng, K.-H. Wu, and Y.-Y. Li, "Pipelined turbo multiuser detection and Kalman channel estimation for MU-MIMO SC-FDMA systems," in *Proceedings of 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, pp. 227–230, Osaka, Japan, October 2015.

[12] A. T. Abebe and C. G. Kang, "Iterative order recursive least square estimation for exploiting frame-wise sparsity in compressive sensing-based MTC," *IEEE Communications Letters*, vol. 20, no. 5, pp. 1018–1021, 2016.

[13] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[14] Y. Wang and W. Yin, "Sparse signal reconstruction via iterative support detection," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 462–491, 2010.

[15] B. C. Wang, L. L. Dai, T. Mir, and Z. C. Wang, "Joint user activity and data detection based on structured compressive sensing for NOMA," *IEEE Communication Letters*, vol. 20, no. 7, pp. 1473–1476, 2016.

[16] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.

[17] C. Wei, H. P. Liu, Z. C. Zhang, J. Dang, and L. Wu, "Approximate message passing-based joint user activity and data detection for NOMA," *IEEE Communication Letters*, vol. 21, no. 3, pp. 640–643, 2017.

[18] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[19] B. Wang, L. Dai, Y. Zhang, T. Mir, and J. Li, "Dynamic compressive sensing-based multi-user detection for uplink grant-free NOMA," *IEEE Communications Letters*, vol. 20, no. 5, pp. 2320–2323, 2016.

[20] M. AsadUllah, M. A. Khan, S. Abbas et al., "Blind channel and data estimation using fuzzy logic-empowered opposite learning-based mutant particle swarm optimization," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 6759526, 12 pages, 2018.

[21] L. Dai, B. Wang, Y. Yuan et al., "Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74–81, 2015.

[22] A. Bayesteh, E. Yi, H. Nikopour, and H. Baligh, "Blind detection of SCMA for uplink grant-free multiple-access," in *Proceedings of ISWCS*, pp. 853–857, Barcelona, Spain, August 2014.

[23] B. Shim and B. Song, "Multiuser detection via compressive sensing," *IEEE Communication Letters*, vol. 16, no. 7, pp. 972–974, 2012.

[24] B. Wang, L. Dai, T. Mir, and Z. Wang, "Joint user activity and data detection based on structured compressive sensing for NOMA," *IEEE Communication Letters*, vol. 20, no. 7, pp. 1473–1476, 2016.

[25] P. Muneer and S. M. Sameer, "Joint ML estimation of CFO and channel, and a low complexity turbo equalization technique for high mobility OFDMA uplinks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 7, pp. 3642–3654, 2015.

[26] F. Rui and L. Zhang, "Generalised grouped minimum mean-squared errorbased multi-stageinterference cancellation scheme for orthogonal frequency division multiple access uplink systems with carrier frequency offsets," *IET Communications*, vol. 7, no. 7, pp. 685–695, 2013.

[27] T. N. Nguyen, X. Lai, B. Tran, and Q. K. Nguyen, "Performance analysis of the VBLAST algorithm for MIMO-OFDMA systems on spatial correlated channels," in *Proceedings of 2014 International Conference on Computing, Management and Telecommunications (ComManTel)*, pp. 146–151, 2014.

[28] S. W. Hou and C. Ko, "Intercarrier interference suppression for OFDMA uplink in time and frequency selective Rayleigh fading channels," in *Proceedings of IEEE International Conference on Vehicular Technology*, pp. 1438–1442, Alberta, Canada, 2008.

[29] C. Durand, F. Durnez, and P. Sehier, "Multiuser detection for a multimedia LEO satellite system," in *Proceedings of IEEE International Conference on Universal Personal Communications*, pp. 653–657, Florence, Italy, October 1998.

[30] A. Kiayani, L. Anttila, Y. Zou, and M. Valkama, "Channel estimation and equalization in multiuser UplinkOFDMA and SC-FDMA systems under transmitter RF impairments," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 82–99, 2016.

[31] Y. Chu, C. Huang, X. Xie et al., "Multilayer hybrid deep-learning method for waste classification and recycling," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 5060857, 9 pages, 2018.

[32] S. Albawi, O. Bayat, S. Al-Azawi et al., "Social touch gesture recognition using convolutional neural network," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 6973103, 10 pages, 2018.

[33] L. Zhou, Q. Li, G. Huo, and Y. Zhou, "Image classification using Biomimetic pattern recognition with convolutional neural networks features," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 3792805, 12 pages, 2017.

[34] H. Chao, H. Zhi, L. Dong, and Y. Liu, "Recognition of emotions using multichannel EEG data and DBN-GC-based ensemble deep learning framework," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 9750904, 11 pages, 2018.

[35] W. Wei, Q. Jia, Y. Feng, and G. Chen, "Emotion recognition based on weighted fusion strategy of multichannel physiological signals," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 5296523, 9 pages, 2018.

[36] J. F. Ossanna, "A model for mobile radio fading due to building reflections:theoretical and experimental fading waveform power spectra," *Bell System Technical Journal*, vol. 43, no. 6, pp. 2935–2971, 1964.

[37] R. H. Clarke, "A statistical theory of mobile radio reception," *Bell System Technical Journal*, vol. 47, no. 2, pp. 957–1000, 1968.

[38] C. Loo, "A statistical model for a land mobile satellite link," *IEEE Transactions on Vehicular Technology*, vol. 34, no. 3, pp. 122–127, 1985.

[39] A. Abdi, W. C. Lau, M. S. Alouini et al., "A new simple model for land mobile satellite channels: first- and second-order statistics," *IEEE Transactions on Wireless Communications*, vol. 2, no. 3, pp. 519–528, 2003.

[40] W. J. Vogel and J. Goldhirsh, "Fade measurements at L-band and UHF in mountainous terrain for land mobile satellite systems," *IEEE Transactions on Antennas and Propagation*, vol. 36, no. 1, pp. 104–113, 1988.

[41] R. Barts, W. Stutzman, W. Smith et al., "Land mobile satellite propagation modeling," in *Proceedings of International Symposium on Antennas and Propagation Society*, pp. 20–23, Blacksburg, VA, USA, 1987.

[42] J. Goldhirsh and W. J. Vogel, *Propagation Effects for Land Mobile Satellite Systems: Overview of Experimental and Modeling Results*, Vol. 1274, NASA Reference Publications, Washington, DC, USA, 1992.

[43] T. Sofos and P. Constantinou, "Propagation model for vegetation effects in terrestrial and satellite mobile systems," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 7, pp. 1917–1920, 2004.

Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 5072918, 8 pages
https://doi.org/10.1155/2019/5072918

Research Article

Using Morphological Data in Language Modeling for Serbian Large Vocabulary Speech Recognition

Edvin Pakoci,[1,2] **Branislav Popović**,[1,3] **and Darko Pekar**[1,2]

[1]*Department for Power, Electronic and Telecommunication Engineering, Faculty of Technical Sciences, University of Novi Sad, 21000 Novi Sad, Serbia*
[2]*AlfaNum Speech Technologies, 21000 Novi Sad, Serbia*
[3]*Department for Music Production and Sound Design, Academy of Arts, Alfa BK University, 11000 Belgrade, Serbia*

Correspondence should be addressed to Edvin Pakoci; edvin.pakoci@uns.ac.rs and Branislav Popović; bpopovic@uns.ac.rs

Received 30 December 2018; Accepted 11 February 2019; Published 3 March 2019

Guest Editor: Zoran Peric

Copyright © 2019 Edvin Pakoci et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Serbian is in a group of highly inflective and morphologically rich languages that use a lot of different word suffixes to express different grammatical, syntactic, or semantic features. This kind of behaviour usually produces a lot of recognition errors, especially in large vocabulary systems—even when, due to good acoustical matching, the correct lemma is predicted by the automatic speech recognition system, often a wrong word ending occurs, which is nevertheless counted as an error. This effect is larger for contexts not present in the language model training corpus. In this manuscript, an approach which takes into account different morphological categories of words for language modeling is examined, and the benefits in terms of word error rates and perplexities are presented. These categories include word type, word case, grammatical number, and gender, and they were all assigned to words in the system vocabulary, where applicable. These additional word features helped to produce significant improvements in relation to the baseline system, both for n-gram-based and neural network-based language models. The proposed system can help overcome a lot of tedious errors in a large vocabulary system, for example, for dictation, both for Serbian and for other languages with similar characteristics.

1. Introduction

There are two main components in any contemporary automatic speech recognition (ASR) system. The first is the acoustic model (AM), which describes acoustical characteristics of different speech components (most often context-dependent phonemes) for a single speaker (in speaker-dependent or speaker-adapted systems) or multiple speakers (in speaker-independent systems). The other component, on which this manuscript will be focused, is the language model (LM), which describes the vocabulary and sentence forming rules of the language or speech domain in question. The language model is used to provide the speech recognizer with allowed word sequences in limited-vocabulary and grammar-based environments, as well as to help the acoustic model to decide on the correct word sequence by introducing costs for all different sequences (i.e., language model costs, or scores), where the more likely sequences will have a lesser cost (a better score). In a lot of applications, a well-trained language model can even overcome certain flaws in the acoustic model, by eliminating unnatural, unlikely word sequences from the list of recognition result possibilities. It has been shown that language models have the capability to become very close to human language understanding [1].

For a long time, the best language models in existence were statistical models based on n-grams—frequencies or probabilities of individual word sequences up to and including length n [2]. These LMs proved to be highly effective for an array of applications, even though they had several known problems, e.g., data sparsity (smoothing requirement [3]) and modeling of longer contexts (more than n words long). Recently, approaches based on recurrent neural networks (RNNs) have been proposed to overcome n-gram issues without raising the implementation difficulty and computational complexity too much. They have shown their

superiority in relation to n-grams [4], but still they are computationally more demanding, and that usually results in a lot longer training duration.

For the Serbian language, in the past couple of years, several variants of RNN-based LMs (RNNLMs) as language models were examined and compared [5]. All of them produced big improvements over the baseline n-gram system, while the best approach seemed to be the TensorFlow-based LSTM-RNNLM (long short-term memory-based RNNLM) approach with pruned lattice rescoring, both in the resulting word error rates (WERs) and training duration. Unfortunately, a number of problems from the n-gram system seemed to remain. The biggest ones were errors where the lemma was correct, but the word ending was wrong, which resulted in a very low character error rate (CER) in comparison to the actual WER. The source of this issue was deemed to be high language inflectivity of Serbian—the same basic word form (lemma) can have a lot of different word suffixes describing different grammatical or syntactic roles (Table 1). In Serbian grammar, there are seven word cases (nominative, genitive, dative, accusative, vocative, instrumental, and locative), which apply to all nouns and most adjectives, as well as some pronouns and numerals, two grammatical numbers (singular and plural), and three grammatical genders (masculine, feminine, and neuter). Grammatical numbers and genders apply to most verbs as well. Cases, numbers, and genders do not apply to invariable words (prepositions, adverbs, conjunctions, particles, and exclamations), even though certain prepositions are always followed by certain cases.

In this manuscript, incorporation of the mentioned morphological features into both n-gram based and RNN-based language models for Serbian is examined, and the obtained results are presented on the largest Serbian audio database for acoustic modeling, as well as all the currently available textual materials in Serbian for language model training.

The following sections will describe relevant previous work, details of the available resources, training methods, the experimental setup, and the results, followed by conclusions.

2. Relevant Previous Work

There were several approaches for incorporation of morphology knowledge into speech recognition systems for other languages, and most of them require some sort of a parser (word decomposer) to determine significant morphological units (morphemes, affixes, etc.) to represent lexical items and word classes, and then that information is used to provide additional constraints to the decoder (in combination or instead of regular words in the conventional approach). A lot of morphologically rich languages face similar issues [6–10].

Another approach is using factored language models (FLMs) [11], which explicitly model relationships between morphological and lexical items in a single language model, and a generalised back-off procedure is used during training to improve the robustness of the resulting FLM during decoding, especially for rarely seen words and n-grams. In the approach in this manuscript, additional morphological information about all the words in the textual corpus for LM training is explicitly embedded into the words themselves, and the LM training is performed on this modified vocabulary. Given the fact that Serbian is a fusional language, which are distinguished from agglutinative languages by their tendency to use a single inflectional morpheme to denote multiple grammatical, syntactic, or semantic features (and that has been a problem for some morphology models [8]), and the planned usage of the ASR system in question in large but relatively finite vocabulary environments (specific domains with a lot of expected words and phrases), such an approach is justifiable, but future research should look into the possibilities of creating open vocabulary systems as well [12, 13].

Table 1: Morphological categories for words in the Serbian language.

Morphological category	Possible category values
Word type	Noun, pronoun, adjective, numeral, verb (variable), preposition, adverb, conjunction, particle, exclamation (invariable)
Word case	Nominative, genitive, dative, accusative, vocative, instrumental, locative
Grammatical number	Singular, plural
Grammatical gender	Masculine, feminine, neuter

3. Materials and Methods

3.1. Audio Database. For all the experiments, the recently expanded speech database for Serbian was used. This database consists of three smaller parts (Table 2). The first part contains audio book recordings, recorded in studio environment by professional speakers. A large part of this database was already mentioned in previous papers [14], but lately it has been expanded by several new audio books. This database part is responsible for 168 hours of data in total, out of which about 140 hours is pure speech (the rest is silence). There are 32 different male and 64 different female recognized speakers (with a slight possibility that a few of them are actually the same speaker), but male speakers had a lot more material by speaker on average. The original data for each speaker were further separated into chunks of 30–35 minutes at most, and all chunks except the first were modified by a carefully chosen combination of speech tempo or pitch changes, basically producing new, mutually distinct subspeakers. The purpose of this procedure was to equalize the amount of material per speaker, as in the original data some speakers have several hours of speech, while others have half an hour or even less. In this way, the trained acoustic models should not be biased towards those speakers with a lot of material. The described procedure resulted in 398 distinct subspeakers. The second part of the database contains radio talk show recordings, separated by speaker. This part totals 179 hours of data, 150 of which are nonsilence, and there are

Table 2: Audio database overview.

Database part	Amount of data (h)	Amount of speech (h)	Male (sub) speakers	Female (sub) speakers
Audio books	168	140	208	190
Radio talk shows	179	150	350	70
Phone recordings	61	41	169	181
Total	408	331	727	441
For training	379	308	677	410

21 male and 14 female speakers in total, again with a lot more material for males. Speaker equalization (in the same manner as above) was also performed here to produce 420 subspeakers. These recordings contain mostly more spontaneous speech, with a lot more background noise, mispronounced words, etc., but are crucial for better modeling of conversational speech. The final database part is the so-called Serbian "mobile" speech database, also mentioned in previous papers [15], and consists of mobile phone recordings of read commands, questions, numbers, dates, names, locations, spellings, and other inquiry-based utterances, like those to be expected in an interaction with a voice assistant type application on a smartphone. These are also more freely spoken, but the utterances are a lot shorter than those in previous database parts, the vocabulary is very domain-oriented and relatively small, and the material is already evenly distributed among speakers. This part contains 61 hours of material, out of which 41 are pure speech, and there are 169 male and 181 female distinct speakers. All audio data for acoustic model training were sampled at 16 kHz, 16 bits per sample, mono PCM.

In addition to this, for testing purposes, 29 hours of material was extracted in total (between 5% and 10% from all database parts), 23 of which is speech, from 81 total test subspeakers. All subspeakers used in the test set were completely used for testing (i.e., excluded completely from training) to avoid biased test results.

3.2. Textual Corpus. All the language models that are going to be mentioned were trained on the same textual corpus. The largest part of it are texts previously collected for Serbian language model training [5, 15], divided into segments corresponding to different functional styles—the largest journalistic corpus, followed by literary, administrative, scientific, popular-scientific, and conversational segments. The whole corpus was used in an attempt to cover as much variability as possible, as it has been shown that sentence structures in different functional styles can differ significantly [16]. Additionally, the transcriptions of the training part of the audio data for acoustic modeling were appended to the existing corpus. In total, there are about 1.4 million sentences and 26 million words. Out of these, 20000 sentences were used only for evaluation (the development, or "dev" set), while the rest were used in the language model training procedure (Table 3).

Table 3: Textual database overview.

Corpus part	#Sentences	#Words	#Characters
Journalistic	737k	17M	94M
Literary	303k	3.9M	18M
Scientific	23k	503k	3M
Administrative	15k	378k	2M
Popular-scientific	18k	357k	2M
Conversational	38k	128k	530k
Transcriptions	251k	3.2M	15M
Total	1.4M	26M	135M
"Dev" set	20k	470k	2.6M

3.3. Training Method—Acoustic Model. The used acoustic models were subsampled time-delay neural networks (TDNNs), which are trained using cross-entropy training within the so-called "chain" training method [17]. For this purpose, the Kaldi speech recognition toolkit [18] was used. The trained neural network is 9 layers deep, with 625 neurons per layer. The initial layers (1–5) were spliced in a {−1, 0, 1} manner (they see 3 consecutive frames), while {−3, 0, 3} splicing was used for the most hidden layers (layers 5–9; they also see 3 frames, but separated by 3 frames from each other). Using this configuration, the most hidden layers need to be evaluated only every 3 frames. No artificial data expansion was used for these experiments. The training was performed in 5 epochs (145 iterations based on the amount of data). Alignments for the deep neural network (DNN) training were provided by a previously trained speaker-adaptive HMM-GMM (hidden Markov model—Gaussian mixture model) system [19] with 3500 states and 35000 Gaussians. Acoustic features used for DNN training were 40 high-resolution MFCC features (Mel-frequency cepstral coefficients), alongside their first- and second-order derivatives, as well as 3 pitch-based features—weighted log-pitch, delta-log-pitch, and warped normalized cross-correlation function (NCCF) value (which is originally between −1 and 1, and higher for voiced frames), and their derivatives, producing a 129-dimensional feature vector, which is a configuration already used in other experiments [5, 15, 17]. The context dependency tree used for the "chain" training with its special model topology that allows a subsampling factor of 3 had 2000 leaves (output states). The effective learning rate was in the range from 0.001 (initial) to 0.0001 (final).

3.4. Training Method—Language Models. The referent n-gram language model is a 3-gram model trained on the described textual data using the SRILM toolkit [20], with Kneser-Ney smoothing and previously optimized pruning cut-off parameter of 10^{-7} [15]. The vocabulary for the LM was chosen in such a way to include all different words from

the acoustic training data transcriptions, plus all other words that are mentioned at least 3 times in the whole textual corpus. Additionally, previously unseen words from the test dataset transcriptions were also added into the vocabulary, so there were no actual out-of-vocabulary (OOV) words, but these transcriptions were not used in probability estimation for the LM. Still, it should be acknowledged that adding OOV words to the LM training vocabulary can affect the recognition accuracy of the ASR system. This approach was related to the planned use of this system (relatively finite-vocabulary domains) and the fact that the experimental results needed to demonstrate the expected WERs in such conditions. Moreover, a similar approach was used previously as well [5, 15], but future research and experiments should measure the WER without adding all the OOV words from the test dataset into the vocabulary and even consider an open vocabulary language model capable of learning new words. The procedure used here resulted in 249809 total words (unigrams), while there were also 1.87 million bigrams and 551 thousand trigrams with the given parameters. The test data perplexity was calculated to be around 634.0.

The RNN-based language model was trained using Kaldi-RNNLM [21], an extension to the Kaldi toolkit, which supports RNN-based language modeling within the Kaldi framework and weighted finite state transducer-(WFST-) based decoding. This method involves subword features; more precisely, letter n-gram counts for better prediction of rare words, as well as augmented features such as scaled word unigram log-probability and word length, the former of which is used for better out-of-domain results. Kaldi-RNNLM also shares the input and output embeddings for the neural network based on work given in [22], which alongside subword features can produce good results on very large vocabularies without having data sparsity issues (which is otherwise usually combatted by using shortlists during LM training). Finally, each of the most frequent N words receives an additional feature, so the top words end up having a one-hot representation in addition to their letter n-gram counts vector and the two augmented features (Table 4).

The baseline RNNLM is a 4-layer combined TDNN plus fast LSTMP (LSTM projected [23]) network, with an embedding dimension of 1024 and both recurrent and non-recurrent projection dimension of 256. The number of most frequent words to receive a special feature is 97636 (calculated to be up to 100000, but to draw a line under a group of words with the same count in the input data). Letter 2-grams and 3-grams are utilized, the minimum frequency of any letter n-gram to be considered a feature was 0.0001, and the training was run for 30 epochs (180 iterations based on input data), with the possibility for the best iteration to be before the last one (best iteration is calculated based on the objective function value on the "dev" dataset previously mentioned in the textual corpus section). For RNNLM rescoring, the pruned lattice rescoring method was used [24] with a 4-gram approximation to prevent lattice explosion and a RNNLM interpolation weight of 0.8 (previously determined to be optimal). The baseline perplexity with this RNNLM on the given test set was calculated using Kaldi tools to be about 119.0.

The approach to incorporation of morphological information into the language model for Serbian in this manuscript is to explicitly embed that information into the words themselves, thus modifying the vocabulary of the ASR system. In order to figure out all the different morphological categories for each word in the input textual corpus sentences, a part-of-speech (POS) tagging tool for Serbian [25] was used, alongside the Serbian morphologic dictionary [26]. Previously, morphological clustering of words into classes using a part of the Serbian textual corpus was examined, where the relevant features were defined for each word type (case, number, and gender, as briefly mentioned in the introduction section of this manuscript, alongside subtype, e.g., proper, common, or abstract for nouns and degree of comparison for adjectives) [27]. Not all the additional features are available for all word types, even within a certain type some words do not behave like others, e.g., there are some invariable adjectives. For the following experiments, word type and case, alongside grammatical number and gender, are chosen as additional word features, and for the final one, the corresponding lemma was taken into account as well.

The POS tagging tool and an additional postprocessing tool were used to convert all input textual data for LM training into sentences with tagged words, i.e., words with one or more delimited suffixes for each word denoting its determined type, case, number, and gender, where applicable. Alongside the ten word types in Serbian, two additional types were introduced—abbreviation and isolated letter (e.g., used when spelling something), since they do not really belong in any other category. Some words were marked by the POS tagger as of unknown type (e.g., badly pronounced words which were written as such in transcriptions or words with typographical errors), so they were not assigned any other morphological features. The POS tagger, with the help of the morphologic dictionary, could distinguish six different cases, as dative and locative in Serbian tend to share the same word form. A case is also assigned to certain prepositions, if they are known to always be followed by a word in a particular case in Serbian. The grammatical number and gender did not receive any special treatment; they are used as already described above (Table 5).

Using the proposed procedure, the number of different words in the LM vocabulary grew to 380747, as some words could, as expected, have different values of certain POS features (sometimes the same word form could be a different combination of case/number/gender in different sentences, even a different word type in some cases). Using the same parameters for smoothing and pruning, the new 3-gram language model has 2.2 million bigrams and 523 thousand trigrams (relatively similar to the referent 3-gram LM). This time though, the perplexity was calculated to be 378.6, which is a lot better, likely because now there is a distinction between formerly same words that could have completely unrelated functions in sentences. On the other hand, the perplexity for the new RNNLM was a bit larger than that for the referent one—147.1, which may be explained by the

Table 4: Example of a feature vector in RNNLM experiments for word *sam* (Eng. *am*).

Index	Feature type	Feature	Value	Remarks
0	Constant	Constant	0.01	For math reasons
1–5	Special	Special word feat.	0	For bos/eos/unk/brk/silence
6	Unigram	Unigram prob.	0.00788	Scaled unigram log-prob.
7	Length	Word length	0.00186	Scaled word length
8–36	Word	1-hot vect. elem.	0	—
37	Word	1-hot vect. elem.	0.21	Scale based on unigram prob.
38–97644	Word	1-hot vect. elem.	0	—
97645–97657	Final	Lett. *n*-gram prob.	0	—
97658	Final	3-gram *-am$* pr.	0.12	Scaled letter 3-gram prob.
97659–97758	Final	Lett. *n*-gram prob.	0	—
97759	Final	2-gram *-m$* pr.	0.047	Scaled letter 2-gram prob.
97760–97869	Final	Lett. *n*-gram prob.	0	—
97870–98050	Initial	Lett. *n*-gram prob.	0	—
98051	Initial	2-gram *ˆs-* pr.	0.03	Scaled letter 2-gram prob.
98052	Initial	3-gram *ˆsa-* pr.	0.069	Scaled letter 3-gram prob.
98053–98144	Initial	Lett. *n*-gram prob.	0	—
98145–98300	Match	Lett. *n*-gram prob.	0	—
98301	Match	2-gram *-am-* pr.	0.064	Scaled letter 2-gram prob.
98302–100451	Match	Lett. *n*-gram prob.	0	—
100452	Match	2-gram *-sa-* pr.	0.057	Scaled letter 2-gram prob.
100453–100459	Match	Lett. *n*-gram prob.	0	—
100460	Match	3-gram *-sam-* pr.	0.11	Scaled letter 3-gram prob.
100461–101306	Match	Lett. *n*-gram prob.	0	—

Table 5: Some of the most frequent words, with and without morphology-based suffixes.

Without POS data	With POS data	Explanation
je	je_gl	*gl*agol = verb
i	i_vez	*vez*nik = conjunction
u	da_vez	—
da	u_pred_dat	*pred*log = preposition
se	se_zam	*zam*enica = pronoun
na	na_pred_dat	*dat*ive/locative
koji	koji_zam_nom_jd_mr	*nom*inative
bi	bi_gl_jd	*jed*nina = singular
Srbije	Srbije_im_gen_jd_mr	*im*enica = noun, *gen*itive, *m*uški *r*od = masculine

implicit vocabulary size increase, which had more effect here in relation to the *n*-gram case (possibly due to applied smoothing and pruning techniques there).

4. Results and Discussion

4.1. 3-Gram Results. The baseline 3-gram language model (250k words, without using morphological information) in combination with the resulting acoustic models trained with the "chain" method produced a word error rate of 8.89%. Problems with the inflectivity of Serbian can be observed when comparing that to the character error rate, which in this experiment was measured to be only 2.63%. The largest number of recognition errors happened in the radio talk show test set (12.64% WER), and the error rate for audio books was in the middle of the road (6.25% WER), while the mobile phone test set produced a very small WER of less than 1% (0.96%), just like in past experiments [15], which can be explained by the very small vocabulary (less than 4000 different words) and repeating word patterns and sentence structures for basically all speakers in this dataset, so the language model could learn to predict such sentences very well. When looking through the list of the most substituted words, on the top of it, the typical confusion between similarly sounding *i* and *je* (Eng. *and*, *be*) can be found, as well as a lot of wrong cases, grammatical genders, and numbers (*koja* instead of *koji*, *koje* or *koju*, and vice versa; Eng. *which*), but also words that have two slightly different but functionally completely equivalent forms (e.g., *kad* and *kada*, Eng. *when*), as well as several obvious typographical errors and words that are often shortened in spontaneous speech (e.g., *znači* and *'nači*, where the starting "*z*" sound is often not pronounced at all, likewise *rekao* and *rek'o*; Eng. *so*, *told*). Some of these errors should be automatically corrected by taking morphological features into account. On the other hand, the typographical errors can only be fixed by carefully looking through all the texts by a group of text checkers.

In comparison, when applying the new 3-gram language model which differentiated POS categories of words, the WER was lowered to 6.90%, and the CER to 2.20%, which is a 22% relative improvement in WER and a 16% relative improvement in CER (Table 6). A breakdown by test database part (audio books, radio talk shows, and phone recordings) shows that the most relative improvement occurred in audio books, possibly due to professionally read texts (no unexpected or mispronounced words and sentence structures most of the time). Somewhat less improvement can be observed for radio talk shows, while a very small deterioration happened for the mobile phone database, even though the error rate is still around the 1% WER mark, probably because of more spontaneity in speech for these

Table 6: WER and CER results for 3-gram experiments, without and with additional POS data taken into account. Breakdown by test database part is shown as well.

Result	Total (%)	Books (%)	Shows (%)	"Mobile" (%)
WER 3-gram	8.89	6.25	12.64	**0.96**
WER 3-gram + POS	**6.90**	**4.12**	**10.45**	1.06
CER 3-gram	2.63	1.45	4.11	**0.40**
CER 3-gram + POS	**2.20**	**1.05**	**3.59**	0.42

two test set parts and likely POS tagger mistakes and/or limitations when used on unconventional word forms encountered there (maybe even transcription errors for talk shows or erroneously recorded audio files for the mobile phone database). The total number of substitutions dropped by more than 25%. The number of wrong-POS-category errors dropped as well, and they were more spaced-out through the list of most common errors (they were more rarely seen in relation to other errors). Insertion rate dropped by 19% and deletion rate by 9%—these errors mostly included very short invariable words, and with the new LM, some occurrences of longer and variable words disappeared from the top of those error lists (Table 7).

4.2. RNNLM Results. The first RNNLM, without morphological features, already gave improvements across the board in comparison to both 3-gram systems. The average WER of 4.90% is a 46% relative improvement to the baseline 3-gram system and a 29% improvement to the 3-gram-POS system. The CER vas measured to be 1.61%. The biggest step forward occurred in the audio books database part again (2.77% WER), but a large step forward was made for radio shows as well (7.56% WER), and even for mobile phone recordings (0.73% WER). Looking at substitutions, insertions, and deletions, the same distribution of errors exists as for the baseline 3-gram system, there are just a lot less of them in absolute numbers.

The RNNLM system with morphological data taken into account produced further improvements in WER and CER—4.34% WER on average and 1.48% CER (Table 8). Best relative improvement was seen for audio books (21%), while radio show error rate lowering was a bit smaller (8%), and phone recordings suffered a 10% relative WER increment (the absolute error rate is still very low), just like in n-gram experiments, and probably for the same reasons. Likelihoods during training, both for the actual training data, and the "dev" data, show consistently slightly better values for the baseline RNNLM, probably due to the same set of reasons as for the difference in perplexities (Figure 1), and it has also been shown that a better perplexity does not necessarily mean a better WER and vice versa [28]. A better way to choose a representative "dev" set should be considered as well. The top list of errors by type, especially the substitutions list, now mostly holds errors that can be categorized as of lower significance. As mentioned before, there are a lot of either typographical errors or badly-pronounced-word errors, words with more than one equivalent similar form in regular usage, etc. The effect is even clearer than in the n-gram case.

The last experiment was related to the usage of lemmas, i.e., basic forms of words, as additional information for RNNLM training. Similarly to how the most frequent words had their own feature (a one-hot vector representation as a subvector of their own word features), the most frequent lemmas were also given special features, so the words whose lemmas are in this set had an additional one-hot vector as a feature, representing the lemma. The number of top lemmas was chosen to be equal to the number of top words (97k). This experiment produced the best results on the given test database so far—a WER of 4.23% and CER of 1.45%. Even though the resulting feature and word embedding matrices for the RNNLM are quite larger in this configuration (as there are a lot more individual word features), the decoding speed does not suffer (but memory consumption issues have to be prevented in this case by not using machines with insufficient memory capacity).

Further improvements can be made in several different parts of the ASR system. Firstly, the acoustic models can probably be improved a bit, both with neural network parameter optimization and with audio database augmentation (e.g., by using speech speed perturbance algorithms, or audio with artificially added noise to improve system robustness). There can be improvements in RNNLM training as well—one way is to optimize training parameters a bit more and another is to make more complex networks, but that will lead to slower decoding speeds. Finally, the textual data can be cleaned up and expanded—one way to do the cleaning is by using a simple text processor tool to fix at least the most common mistakes from the top errors lists, which can also be used as a recognition result postprocessor with the current system. Currently, there is an additional textual database in preparation for future trainings. For usage in specific domains, one type of RNNLM training texts can be used with a larger weight, so the final system would prefer sentence structures mostly found in the desired type of text.

5. Conclusions

The experiments and the obtained results described in this manuscript show that using additional morphological knowledge for language model training can solve a large part of problems for highly inflective languages, as the Serbian language is. The proposed method incorporated the additional data into the words themselves, and one experiment used additional RNNLM word features on top of that. Big improvements were obtained both in n-gram systems and in RNNLM-based systems in relation to baseline systems which did not use any morphological data. The used Kaldi-RNNLM toolkit has also proven to be superior to any other previously used language model training toolkit for Serbian. There is still room for improvement, and there are future plans to create even better both acoustic and language models and even to further optimize the usage of morphological category information in the modeling of the Serbian language. Finally, an

Table 7: Lists of some of the most frequent word errors by type, with #occurrences (3-gram LM).

Substitutions without POS	Substitutions with POS	Insertions without POS	Insertions with POS	Deletions without POS	Deletions with POS
je ⟶ i (88)	je ⟶ i (79)	i (271)	je (242)	je (769)	je (742)
i ⟶ je (61)	i ⟶ je (50)	je (260)	i (235)	i (713)	i (669)
iz ⟶ i (48)	iz ⟶ i (39)	u (112)	u (88)	u (332)	u (302)
reko ⟶ rekao (42)	koji ⟶ koju (36)	da (87)	da (85)	da (215)	da (204)
koji ⟶ koju (40)	reko ⟶ rekao (32)	a (69)	a (54)	a (129)	a (130)
koja ⟶ koje (39)	sa ⟶ s (29)	na (54)	na (37)	on (121)	on (114)
koju ⟶ koje (37)	se ⟶ su (28)	po (31)	on (25)	na (99)	na (82)
sa ⟶ s (33)	je ⟶ oni (27)	o (28)	se (24)	to (76)	to (75)
nači ⟶ znači (31)	koji ⟶ koje (25)	ne (25)	o (22)	ja (75)	ja (63)
se ⟶ su (31)	nači ⟶ znači (25)	se (25)	pa (19)	od (63)	se (60)
je ⟶ koje (30)	koja ⟶ koje (24)	on (23)	od (17)	ne (62)	od (56)
tu ⟶ to (28)	mi ⟶ i (23)	—	—	se (61)	mi (54)
—	—	s (11)	ne (14)	—	—
kada ⟶ kad (22)	kada ⟶ kad (19)	kaže (10)	s (11)	joj (29)	sam (29)
imo ⟶ imao (19)	imo ⟶ imao (18)	koje (10)	kaže (10)	sam (28)	koji (25)
bilo ⟶ bila (19)	bila ⟶ bilo (18)	koji (10)	ovo (9)	koji (25)	joj (23)

Table 8: WER and CER results for RNNLM experiments, without and with additional POS data taken into account. Breakdown by test database part is shown as well.

Result	Total (%)	Books (%)	Shows (%)	"Mobile" (%)
WER RNNLM	4.90	2.77	7.56	**0.73**
WER RNNLM + POS	**4.34**	**2.18**	**6.93**	0.81
CER RNNLM	1.61	0.70	2.69	**0.28**
CER RNNLM + POS	**1.48**	**0.59**	**2.52**	0.33

Figure 1: Plot of average log-likelihoods per word during training with respect to training iterations, recorded over 180 iterations in total, for both RNNLM experiments. Likelihoods for the training set (blue lines) and for the "dev" set (red lines) are separated as well.

open-vocabulary language model capable of learning new words needs to be considered as well.

Data Availability

The audio and textual databases used to support the findings presented in this manuscript are partly available online and partly collected and owned by the Faculty of Technical Sciences in Novi Sad and the AlfaNum Company. All the mentioned data can be made available by the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This paper was supported in part by the Ministry of Education, Science and Technological Development of the Republic of Serbia, within the project "Development of Dialogue Systems for Serbian and Other South Slavic Languages" (TR32035), EUREKA project DANSPLAT, "A Platform for the Applications of Speech Technologies on Smartphones for the Languages of the Danube Region" (E Platf), and the Provincial Secretariat for Higher Education and Scientific Research, within the project "Central Audio-Library of the University of Novi Sad" (114-451-2570/2016-02).

References

[1] J. T. Goodman, "A bit of progress in language modeling: extended version," Tech. Rep. MSR-TR-2001-72, Microsoft Research, Redmond, WA, USA, 2001.

[2] R. Rosenfeld, "Two decades of statistical language modeling: where do we go from here?," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.

[3] R. Kneser and H. Ney, "Improved backing-off for M-gram language modeling," in *Proceedings of 20th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 181–184, Detroit, MI, USA, May 1995.

[4] T. Mikolov, S. Kombrink, L. Burget, J. H. Černocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proceedings of 36th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5528–5531, Prague, Czech Republic, May 2011.

[5] B. Popović, E. Pakoci, and D. Pekar, "A comparison of language model training techniques in a continuous speech recognition system for Serbian," in *Proceedings of 20th International Conference on Speech and Computer (SPECOM)*, vol. 11096 of Lecture Notes in Artificial Intelligence, pp. 522–531, Leipzig, Germany, September 2018.

[6] H. Sak, M. Saraçlar, and T. Güngör, "Morphology-based and sub-word language modeling for Turkish speech recognition," in *Proceedings of 35th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5402–5405, Dallas, TX, USA, March 2010.

[7] T. Müller, H. Schütze, and H. Schmid, "A comparative investigation of morphological language modeling for the languages of the European Union," in *Proceedings of 12th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, pp. 386–395, Montréal, Canada, June 2012.

[8] R. Sarikaya, M. Afify, Y. Deng, H. Erdogan, and Y. Gao, "Joint morphological-lexical language modeling for processing morphologically rich languages with application to dialectal Arabic," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 7, pp. 1330–1339, 2008.

[9] M. Kurimo, A. Puurula, E. Arisoy et al., "Unlimited vocabulary speech recognition for agglutinative languages," in *Proceedings of 8th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, pp. 104–111, New York, NY, USA, June 2006.

[10] O.-W. Kwon and J. Park, "Korean large vocabulary continuous speech recognition with morpheme-based recognition units," *Speech Communication*, vol. 39, no. 3-4, pp. 287–300, 2003.

[11] K. Kirchhoff and M. Yang, "Improved language modeling for statistical machine translation," in *Proceedings of ACL 2005 Workshop on Building and Using Parallel Text: Data-Driven Machine Translation and Beyond*, pp. 125–128, Ann Arbor, MI, USA, June 2005.

[12] A. Matthews, G. Neubig, and C. Dyer, "Using morphological knowledge in open-vocabulary neural language models," in *Proceedings of 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, vol. 1, pp. 1435–1445, New Orleans, LA, USA, June 2018.

[13] L. Qin, *Learning out-of-vocabulary words in automatic speech recognition*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2013.

[14] S. Suzić, S. Ostrogonac, E. Pakoci, and M. Bojanić, "Building a speech repository for a Serbian LVCSR system," *Telfor Journal*, vol. 6, no. 2, pp. 109–114, 2014.

[15] E. Pakoci, B. Popović, and D. Pekar, "Language model optimization for a deep neural network based speech recognition system for Serbian," in *Proceedings of 19th International Conference on Speech and Computer (SPECOM)*, pp. 483–492vol. 10458 of Lecture Notes in Artificial Intelligence, Hatfield, UK, September 2017.

[16] S. Ostrogonac, D. Mišković, M. Sečujski, D. Pekar, and V. Delić, "A language model for highly inflective non-agglutinative languages," in *Proceedings of 10th International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 177–181, Subotica, Serbia, September 2012.

[17] E. Pakoci, B. Popović, and D. Pekar, "Improvements in Serbian speech recognition using sequence-trained deep neural networks," *SPIIRAS Proceedings*, vol. 3, no. 58, pp. 53–76, 2018.

[18] D. Povey, A. Ghoshal, G. Boulianne et al., "The Kaldi speech recognition toolkit," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 1–4, Waikoloa, HI, USA, December 2011.

[19] D. Povey, H.-K. J. Kuo, and H. Soltau, "Fast speaker adaptive training for speech recognition," in *Proceedings of 9th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1245–1248, Brisbane, Australia, September 2008.

[20] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "SRILM at sixteen: update and outlook," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 5–9, Waikoloa, HI, USA, December 2011.

[21] H. Xu, K. Li, Y. Wang et al., "Neural network language modeling with letter-based features and importance sampling," in *Proceedings of 43rd International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6109–6113, Calgary, Canada, April 2018.

[22] O. Press and L. Wolf, "Using the output embedding to improve language models," in *Proceedings of 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, vol. 2, pp. 157–163, Valencia, Spain, April 2017.

[23] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proceedings of 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 338–342, Singapore, September 2014.

[24] H. Xu, T. Chen, D. Gao et al., "A pruned RNNLM lattice-rescoring algorithm for automatic speech recognition," in *Proceedings of 43rd International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5929–5933, Calgary, Canada, April 2018.

[25] S. Ostrogonac, "Automatic detection and correction of semantic errors in texts in Serbian," in *Proceedings of 5th International Congress "Applied Linguistics Today," New Tendencies in Theory and Practice*, Novi Sad, Serbia, November 2015.

[26] M. Sečujski, "Accentuation dictionary for Serbian intended for text-to-speech technology," in *Proceedings of 4th Conference on Digital Speech and Image Processing (DOGS)*, pp. 17–20, Bečej, Serbia, May 2002.

[27] S. Ostrogonac, E. Pakoci, M. Sečujski, and D. Mišković, "Morphology-based vs unsupervised word clustering for training language models for Serbian," *Acta Polytechnica Hungarica: Journal of Applied Sciences*, 2019, In press.

[28] D. Klakow and J. Peters, "Testing the correlation of word error rate and perplexity," *Speech Communication*, vol. 38, no. 1-2, pp. 19–28, 2002.

Hindawi
Computational Intelligence and Neuroscience
Volume 2019, Article ID 8162567, 9 pages
https://doi.org/10.1155/2019/8162567

Research Article

KeratoDetect: Keratoconus Detection Algorithm Using Convolutional Neural Networks

Alexandru Lavric and Popa Valentin

Computers, Electronics and Automation Department, Stefan cel Mare University of Suceava, Suceava 720229, Romania

Correspondence should be addressed to Alexandru Lavric; lavric@eed.usv.ro

Received 25 October 2018; Revised 19 November 2018; Accepted 24 December 2018; Published 23 January 2019

Guest Editor: Vlado Delic

Copyright © 2019 Alexandru Lavric and Popa Valentin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Keratoconus (KTC) is a noninflammatory disorder characterized by progressive thinning, corneal deformation, and scarring of the cornea. The pathological mechanisms of this condition have been investigated for a long time. In recent years, this disease has come to the attention of many research centers because the number of people diagnosed with keratoconus is on the rise. In this context, solutions that facilitate both the diagnostic and treatment options are quickly needed. The main contribution of this paper is the implementation of an algorithm that is able to determine whether an eye is affected or not by keratoconus. The KeratoDetect algorithm analyzes the corneal topography of the eye using a convolutional neural network (CNN) that is able to extract and learn the features of a keratoconus eye. The results show that the KeratoDetect algorithm ensures a high level of performance, obtaining an accuracy of 99.33% on the data test set. KeratoDetect can assist the ophthalmologist in rapid screening of its patients, thus reducing diagnostic errors and facilitating treatment.

1. Introduction

The cornea is the outer layer of the eye, the surface covering the front of the eye. The structural and repair properties of the cornea are essential for its function: protecting the inner contents of the eye, maintaining the shape of the eye, and achieving light refraction.

The cornea is composed of proteins and cells and does not contain blood vessels, unlike most tissues in the human body. The existence of blood vessels may affect its transparency, which in turn may affect the proper light refraction, hence worsening vision. Because there are no blood vessels that supply nutrients in the cornea, tears and aqueous humor (an aqueous liquid) provide nutrients to the cornea.

The cornea is composed of five layers: the epithelium, the Bowman's layer, the stroma, Descemet's membrane, and the endothelium. The first layer, the epithelium, is a cell layer that covers the cornea. This layer absorbs oxygen from tears and passes it to the rest of the cornea. The cornea also contains free nerve endings. Keratoconus is a noninflammatory condition characterized by progressive thinning, deformation, and scarring of the cornea. The pathological mechanisms of keratoconus have been investigated for a long time. Both the genetic and environmental factors have been associated with the disease, but in recent years, a new theory emerges that keratoconus could also have an inflammatory component [1].

However, since many patients with allergies rub their eyes excessively, it has not been clearly established whether eye rubbing is a factor related to the keratoconus pathology but can definitely increase the cornea deformation [2]. Keratoconus eyes are characterized by a cornea that has a progressive cone deformation that gradually reduces the visual accuracy of the patient. Over the years, all kinds of methods have been proposed to correct this condition, such as rigid RGP contact lenses and scleral lenses, or different surgeries to reduce the progression of the disease. In recent years, this disease has come to the attention of many research centers because the number of people diagnosed with keratoconus is continually growing and quick solutions are needed to facilitate both the diagnosis and treatment options.

Hidalgo et al. [3] use a support vector machine (SVM) algorithm that processes 22 parameters in order to differentiate patients suffering from keratoconus from those who are healthy. The algorithm processes corneal topography parameters and measurements of the patient's eye. The development environment used in the study is Weka open source software [4]. The results show an accuracy of 95.2%, which represents a high level of performance. According to the authors, the most suitable algorithm for KTC classification is one of the SVM types. From the obtained results, the reduction of parameters does not affect the performance of the classification algorithm. Contrarily, it may even improve the accuracy in some particular cases, considerably reducing the processing time. The best accuracy was obtained when only 7 parameters were integrated in the model.

Accardo et al. [5] present a keratoconus detection algorithm that uses a neural network (NN). The accuracy achieved by the implemented algorithm is 96.4%. This accuracy was obtained by simultaneously using the topographic parameters of both eyes, which improves the discriminative ability of the neural network.

Valdés-Mas et al. [6] propose the use of an artificial neural network (NN) to predict the evolution of keratoconus. The quality of vision is assessed in the work by analyzing corneal curvature and the astigmatism. The paper proposes a series of models, the best results being obtained through an artificial neural network based on a multilayer perceptron (MLP).

The main benefits of integrating machine learning algorithms in ophthalmology are immense [7]. The machine learning domain is a subbranch of artificial intelligence that, as Arthur Samuel said, "gives computers the ability to learn without being explicitly programmed."

The first step of a machine learning algorithm is data processing, followed by the training mechanism; then, the algorithm is deployed to predict new features meanwhile monitoring the accuracy. After the algorithm is prepared and trained, the new data (test data) are fed into the network.

Machine learning algorithms have the potential to interrupt classical health screening programs, being able to provide diagnostics in a very short time helping to increase patient care and comfort. One aspect that should not be neglected is the validation of these algorithms, which must be accomplished under objective conditions and by specialized programs.

Fatemeh et al. [8] present an automatic keratoconus detection algorithm by using artificial intelligence. The algorithm uses as input a set of topography images obtained using a Pentacam [9] that has been labelled by specialists in two categories (keratoconus eyes and nonkeratoconus eyes). The disadvantage of the proposed approach is that the training data set contains a small number of images, respectively, 82.

Four groups of classifiers have been tested: multilayer perceptron (MLP) [10], radial basis function network (RBFN) [11], neural network (NN), and support vector machine (SVM) [12]. Simulations show that the results of the classification schemes have roughly the same results in terms of accuracy. The best accuracy obtained was 92.2% for the MLP algorithm, and the lowest accuracy of 84.42% for the SVM algorithm.

In [13], a learning algorithm is presented that is based on a neural network that allows the diagnosis of keratoconus. From the obtained results, the highest accuracy 97.33% was achieved on the test data set. The main disadvantage of the algorithm is the use of a large number of parameters which makes it difficult to implement and test.

Perissutti et al. [14] are among the first to propose the use of a neural network in diagnosing keratoconus. The maximum level of accuracy was 92%. The paper compares the way of detection that integrates both a monocular approach and a binocular approach. The best results are obtained by the authors when the parameters obtained from both eyes of the patient are used in the model.

Arbelaez et al. [15] develop a classification algorithm showing high precision in terms of differentiation between normal eyes and keratoconus-affected eyes. The model also includes posterior corneal surface parameters and corneal thickness. The integration of these parameters significantly improved the sensitivity of the algorithm when considering subclinical keratoconus diagnosis. The classification mechanism may be particularly useful when detecting early signs associated with keratoconus. The accuracy of the classifier developed by the authors was approximately 97.2%. Diagnosing keratoconus at an early stage is very important for the prognosis of the disease.

Maeda et al. [16] combine a discriminatory analysis and a classification tree to analyze data from the corneal topography in order to diagnose keratoconus. The sensitivity of the mechanism was 89%, which is less than the value obtained with an SVM-type algorithm of 95%.

Twa et al. [17] consider modelling the anterior corneal surface with a seventh-order Zernike polynomial and applying a decision tree type to differentiate between a normal eye and a keratoconus one. The sensitivity, specificity, and precision of the method were 92%, 93%, and 94%, respectively.

Chastang et al. [18] integrated a binary decision tree based on the data generated by the corneal topography and achieved a precision of about 95%. The disadvantage of the proposed approach is the existence of a reduced input data set [19]. In conclusion, the algorithms that can be integrated in the keratoconus diagnosis process are as follows: multilayer perceptron (MLP), decision tree (DT), convolutional neural network (CNN), radial basis function network (RBFN), artificial neural network (ANN), and support vector machine (SVM).

The paper is structured as follows: after a short introduction, the motivation that led to this work is presented. In Section 3, the proposed keratoconus detection algorithm (KeratoDetect) is discussed; meanwhile, in Section 4, the performance evaluation of the KeratoDetect algorithm is presented. Section 5 concludes this work. The main contribution of this work is the integration of a convolutional neural network in the diagnostic process of keratoconus. From the best knowledge of the authors, this approach is the first one. In the scientific literature, a handful of algorithms [3–18, 20] have been presented that use machine learning

technologies in keratoconus detection, but none of them obtained such a high level of accuracy.

2. Theoretical Background, Symptoms, and Treatment

Keratoconus (KTC) is found in the general population with a ratio of 1 in 2000 persons; also, the incidence among children has increased significantly over the past few years. The number of people diagnosed with keratoconus is rising because more and more people are initially diagnosed when performing the screening for laser refracting surgery which includes an eye topography. Thus, the incidence of keratoconus in the population can be even greater.

The epidemiological reports published in recent years show that in Russia only 0.3 cases per 100,000 people (0.0003%) are reported, in India 2.3% cases, in Israel 2.34%, and 2.5% in Iran [1]. Usually, the illness debuts in the second decade of life and affects both sexes and all ethnicities. Keratoconus usually affects both eyes; however, one eye may be initially affected.

In keratoconus, because of the structural changes caused by the thinning of the cornea, intraocular pressure is no longer provided by the cornea. Thence, the cornea deforms by taking a conical shape. The name of the condition is given by this happening. KTC affects both the young and the elderly, so it is urgently necessary to find new ways of diagnosing so that the keratoconus can be dealt with beforehand.

In advanced cases, most of the time, there is a significant distortion of vision, worsening the quality of life. The most advanced form of vision correction requires a corneal transplant, which involves risks such as rejection and infection. The safest bet is to have a correct diagnosis as soon as possible, so that the patient has the opportunity to undertake treatment that slows the progression of the disease.

A basic proven treatment is the corneal cross-linking that aims to prevent disease progression [2]. The Corneal Cross-Linking (CXL) treatment is aimed at restoring the integrity of the corneal matrix, hence increasing the resistance to keratoconus progression. For the effectiveness of this treatment to be maximum, it must be done when the disease is at an early stage. Thus, it is imperative to correctly diagnose the early stage of keratoconus, which most often has puberty onset.

Continuous development of large sets of ophthalmic data, sustained by improvement of learning algorithms, and the increased processing power have led to heightened interest in applying machine learning algorithms in ophthalmology.

The main contribution of this paper is the implementation of an algorithm named "KeratoDetect" that is able to determine if an eye suffers from keratoconus. In Figure 1, the differences between a normal cornea and one affected by keratoconus (KTC) disease (e.g., cone-like cornea) are presented.

3. KeratoDetect: Keratoconus Detection Algorithm

The main goal is to implement and test an algorithm that allows keratoconus detection by facilitating the diagnostic process. The algorithm uses a convolutional neural network (CNN). The most used modality to diagnose and confirm keratoconus is to make a corneal topography which is then interpreted by the ophthalmologist specialist. These images will consider the input of KeratoDetect algorithm, within the learning process associated with the convolutional neural network (CNN).

Figure 1: The keratoconus (a) and normal (b) cornea.

The developed neural network processes the input data (e.g., pixel values of an image representing the corneal topography) using weights on connections between neurons. The learning process involves the continuous adjustment of these weights so as to reduce the error in both the classification and learning processes.

Many learning algorithms use multilayer networks between inputs and outputs. These neural networks allow the identification of features, patterns, and characteristics within the classification relationship. Technological progress has led to the development and refinement of these algorithms, with them being used in many areas of medicine with confidence.

In neural networks, a convolutional neural network (CNN) is one of the main methods of recognizing and classifying images. CNNs are currently used in applications such as object recognition and face detection. A CNN that is capable of diagnosing the keratoconus disease is implemented in this paper.

Classifying images using the CNN algorithm involves an image preprocessing step. The algorithm decomposes the image at the pixel level; the obtained matrix is then applied at the input of the neural network. The algorithm uses color images that have a size of $180 \times 240 \times 3$ pixels. The CNN model assumes that each image passes through a series of kernel convolutional filters, pooling layers, and fully connected (FC) layers; meanwhile, the classification is performed by using the Softmax function to classify an object with probabilistic values between 0 and 1.

In Figure 2, the structure of such a CNN is presented. One can observe the input layer, the convention layer, the pooling layer, the fully connected layer, and the output layer.

The convolution layer is the first layer that performs feature extraction from an image applied to the networks' input. Convolution layer keeps the relationship between pixels by learning the characteristics of images using certain filters, thus generating the image matrix.

In Figure 3, two corneal topographies are shown; the former, an eye affected by keratoconus, and the latter, a

Figure 2: Structure of a CNN.

(a) (b)

Figure 3: Cornea topographies. (a) Keratoconus eye. (b) Healthy eye.

healthy eye. The colors used in a topography show any changes to the elevation parameter of the cornea. Thus, colors such as red indicate the existence of a high elevation (e.g., the possible existence of a cone). Light colors such as green or yellow show a uniform distribution of the elevation parameter.

Topographic maps use a color scale to identify corneal curvature data. Curved steep areas are displayed in warm colors, such as red and orange, whereas flat bend areas are shown in cold colors, such as green and blue [21]. Topography displays color maps in "absolute" and "normalized" scales. The topography gives an overview of the whole cornea in terms of its curvature. The implemented algorithm processes typical corneal topographies and classifies them into two categories, detecting patterns specific to the keratoconus pathology.

Figure 4 presents the steps integrated in algorithm proposed by KeratoDetect. The first step involves the preprocessing of the images that will be applied to the input of the algorithm.

Input images must have the same resolution before being applied to the convolutional neural network; hence, they need to be preprocessed. The next step is to divide the images into three sets: one used for neural network training, another data set used for validation, and a third set that will be used to test the implemented algorithm after completing the learning and validation processes.

The next step is to train the neural network using the training image set. Once the accuracy obtained is acceptable, the test image set is applied at the algorithm input. The final parameter analysed and evaluated is the accuracy with which the algorithm was able to correctly classify topography into two categories: keratoconus-affected eye and healthy eye with a normal topography. The accuracy is computed on the test data set.

The colors used in a topography show changes in the elevation parameter of the cornea. Thus, colors nearer to red indicate the existence of a cone. The implemented algorithm processes topical corneal topographies and classifies them into two categories, detecting patterns specific to the keratoconus pathology.

4. Performance Evaluation of KeratoDetect

One of the initial problems that we encountered was the hindrance to gathering a large set of corneal topographies that we can apply to the proposed algorithm input. Usually, clinical data are difficult to obtain. So, as a solution, we integrated the SyntEyes KTC model [22]. The model used to generate the input data (e.g., the corneal topographies) is a stochastic one, allowing for the automatic generation of corneal topographies [23].

In Figure 5, a selection of topographies generated by the SyntEyes is presented.

Figure 4: CNN-proposed algorithm.

Figure 5: Algorithm input.

The SyntEyes model uses as baseline 145 Scheimpflug tomographs, also incorporating eye biometry-related information. Information is processed to reduce the number of integrated parameters followed by a multivariable Gaussian analysis that produces the stochastic model of the keratoconus (KTC) eye. The results of this model are filtered to remove incorrect topographic models either by an automatic or a manual procedure. The evolution and characteristics of keratoconus can be studied and analysed by comparing SyntEyes KTC output with the output provided by the pattern that generates healthy eye with normal topographies because the synthetic data provided by the model resemble current clinical data and can be integrated as learning data within the CNN. The KeratoDetect algorithm was developed using 1500 healthy eye topographies and 1500 topography of the eyes diagnosed with keratoconus. CNN's training set uses 1350 topographies. The algorithm validation data set includes 150 eyes; meanwhile, the test dataset will include 200 eyes on which the performance level will be measured. The CNN algorithm is implemented in Matlab software [24].

Figure 6 presents the CNN structure implemented in Matlab software. The CNN uses a filter with a size of 3 by 3 pixels for scanning procedure of the training set. The number of neurons used in the convolutional layer is 16 (e.g., the number of filters) neurons that connect to the same region of the input data set.

At the convolutional layer, a padding feature is used in order to create the features map. In our case (a filter of 3), a padding of 1 is used in order to ensure that the spatial output size is the same as the input size. The next integrated layer is a normalization layer that allows the optimization of the network by normalizing the activations.

This layer is followed by a ReLU (rectified linear units) layer. The purpose of this layer is to add nonlinearity after

Figure 6: CNN-designed algorithm.

each convolutional layer. This layer lowers the training time of the network and reduces the sensitivity of the CNN.

The next layer of the designed CNN is a max pooling layer which has the role to make a downsampling operation in order to remove redundant information from the layers. This process is used as to increase the number of filters without the supplementary increase in computational power. The max pooling layer returns the maximum values of the defined rectangular regions, using a stride combination with a defined step size of 2 for the training process. The learning rate of the network is set to 0.01.

The next step is to increase the number of filters from 16 to 32 and perform the normalization operations followed by the ReLU layer. After these operations have been fulfilled, the max pooling function is applied. The data are prepossessed again this time by a 64-neuron convolutional layer followed by a normalization and a ReLU layer.

The last layer is a fully connected layer where all the neurons connect to the neurons from the previous layers and exchange data. This layer combines all the features extracted and learned by the previous layers as to identify patterns in the input data. In this layer, all the extracted features are used to classify the corneal topography. The size of the fully connected layer is equal to the classification classes, in our Case 2 (one for keratoconus eyes and one for the normal eyes).

The next layer is a Softmax layer that determines the classification probabilities used by the final classification layer. The classification layer uses an activation function for each input to assign one of the two classes and calculate the loss parameter.

Figure 7 shows the accuracy parameter when the number of iterations is varied. One epoch is a set of data used to train the convolutional neural network and coincides with the complete pass of the input data set. One epoch includes 21 iterations. An iteration represents a neural parameter update in the convolutional neural network. The test data include a number of 400 corneal topographies, 200 topographies of keratoconus eyes, and 200 topographies of normal eyes. For a number of 10 epochs (i.e., 210 iterations), we obtain an accuracy on the data test set of approximately 97.01%. The accuracy is computed for the test data set.

Figure 7: Algorithm accuracy parameter.

Figure 8 presents the accuracy parameter when 630 iterations equivalent to 30 epochs are used. Each iteration is an estimate of the gradient, which corresponds to an update of the neural network parameters. In this situation, the developed algorithm has an accuracy of 97.67% for the test data set that includes 400 topographies. By default, the CNN validates the network every 50 iterations by making predictions on the validation data set and calculating the accuracy parameter.

Figure 9 shows the loss parameter of the CNN. This parameter represents the cross entropy of the entire CNN. When we increase the accuracy of the network, the loss parameter significantly reduces contributing to the elimination of incorrect topographies classification.

Figure 10 presents the accuracy parameter for 798 iterations equivalent to 38 epochs.

Each iteration represents an update of the neural network weights, thus increasing the network performance. For the test data set, the developed algorithm has an accuracy of 99.33%. Obviously, in this case, the learning time of the neural network has increased significantly. The algorithm allows detection of keratoconus suspect corneas with very high accuracy.

In Figure 11 is presented the loss parameter in case the accuracy of the proposed algorithm is 99.33%.

In Figure 12, the initial convolutional layer weights before the learning process are shown as well as the first

Figure 8: Accuracy parameter for 630 learning iterations.

Figure 9: Losses for 630 learning iterations.

Figure 10: Accuracy parameter for 798 learning iterations.

Figure 11: Losses for 798 learning iterations.

(a) (b)

Figure 12: Initial convolution layer weights (before training) (a) and the first convolutional layer weights (after training) (b).

convolutional layer weights of the CNN after the training process. These weights are used for feature extraction and further processing.

The main advantage of the proposed algorithm is that it ensures the highest level of performance when we compare the obtained accuracy with the accuracy of other algorithms presented in the scientific literature. The obtained level of performance also represents the main contribution of this work.

5. Conclusions

The main advantage of the proposed algorithm is that it can be used as an integrated part of the diagnostic process. From the obtained results, we can conclude that the proposed KeratoDetect algorithm ensures a high level of performance. The main contribution of this work is the development and integration in the diagnostic process of an assistant software to help the ophthalmologist. Machine learning algorithms have the potential to interrupt classical medical screening programs, being able to provide diagnostics in a very short time as well as helping to increase patient care and comfort. The contribution of this paper consists in applying a machine learning mechanism to keratoconus disease detection. The high level of performance can come to help the medical staff in correctly diagnosing keratoconus.

Thus, after the ophthalmological consultation, the corneal topography is applied as an input to the already trained neural network, and this will determine whether the patient is suffering from keratoconus or not. By optimizing the parameters associated with the convolutional neural network, the accuracy of the proposed algorithm was increased to 99.33% for the test set. The implemented algorithm processes topical corneal topographies and classifies them into two categories, detecting patterns specific to the keratoconus pathology.

In conclusion, this paper presents the development of a screening tool based on a learning algorithm that automatically detects the keratoconus disease based on corneal topographies.

The algorithm can be implemented in the device that performs the topography as an add-on in order to assist the ophthalmologist in rapid screening of its patients. Although the results show a high level of performance, this mechanism

cannot be used as a stand-alone diagnosis procedure but should be seen as an additional tool to help the ophthalmologist who also analyzes other clinical data such as family history, refraction, and corneal shape evolution, and performs lamp examination. In the future, these algorithms will become more efficient and contribute to easy diagnosis of keratoconus and the reduction of corneal transplant cases.

Data Availability

The data results used to support the findings of this study are presented in this paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI-UEFISCDI, Project Nos. PN-III-P1-1.2-PCCDI-2017-0776 and 36 PCCDI/15.03.2018, within PNCDI III.

References

[1] V. Galvis, T. Sherwin, A. Tello, J. Merayo, R. Barrera, and A. Acera, "Keratoconus: an inflammatory disorder?," *Nature Eye*, vol. 29, no. 7, pp. 843–859, 2015.

[2] H. A. Salmon, D. Chalk, K. Stein, and N. A. Frost, "Cost effectiveness of collagen crosslinking for progressive keratoconus in the UK NHS," *Nature Eye*, vol. 29, no. 11, pp. 1504–1511, 2015.

[3] I. R. Hidalgo, P. Rodriguez, J. J. Rozema et al., "Evaluation of a machine-learning classifier for keratoconus detection based on Scheimpflug tomography," *Cornea*, vol. 35, no. 6, pp. 827–832, 2016.

[4] Weka, https://deeplearning.cms.waikato.ac.nz/.

[5] P. A. Accardo and S. Pensiero, "Neural network-based system for early keratoconus detection from corneal topography," *Journal of biomedical informatics*, vol. 35, no. 3, pp. 151–159, 2002.

[6] M. A. Valdés-Mas, "A new approach based on Machine Learning for predicting corneal curvature (K1) and astigmatism in patients with keratoconus after intracorneal ring implantation," *Computer Methods and Programs in Biomedicine*, vol. 116, no. 1, pp. 39–47, 2014.

[7] A. Lee, P. Taylor, J. Kalpathy-Cramer, and A. Tufail, "Machine learning has arrived!," *Ophthalmology*, vol. 124, no. 12, pp. 1726–1728.

[8] F. Toutounchian, J. Shanbehzadeh, and M. Khanlari, "Detection of keratoconus and suspect keratoconus by machine vision," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, China, March 2012.

[9] M. W. Belin and S. S. Khachikian, "An introduction to understanding elevation-based topography: how elevation data are displayed–a review," *Clinical & Experimental Ophthalmology*, vol. 37, no. 1, pp. 14–29, 2009.

[10] A. G. Bors and I. Pitas, "Median radial basis function neural network," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1351–1364, 1996.

[11] K. K. Tao, "A closer look at the radial basis function (RBF) networks," in *Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems, and Computers*, A. Singh, Ed., IEEE Computer Society Press, Los Alamitos, CA, USA, 1993.

[12] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics*, pp. 275–285, 2004.

[13] M. K. Smolek and S. D. Klyce, "Current keratoconus detection methods compared with a neural network approach," *Investigative Ophthalmology & Visual Science*, vol. 38, no. 11, pp. 2290–2299, 1997.

[14] P. Perissutti, A. P. Accardo, S. Pensiero, and M. L. Salvetat, "Automatic keratoconus detection by means of a neural network: comparison between a monocular and a binocular approach," in *Proceedings of the 20th Annual International Conference of the IEEE*, vol. 3, pp. 1397–1399, Engineering in Medicine and Biology Society, Hong Kong, China, November 1998.

[15] M. C. Arbelaez, F. Versaci, G. Vestri, P. Barboni, and G. Savini, "Use of a support vector machine for keratoconus and subclinical keratoconus detection by topographic and tomographic data," *Ophthalmology*, vol. 119, no. 11, pp. 2231–2238, 2012.

[16] N. Maeda, S. D. Klyce, M. K. Smolek, and H. W. Thompson, "Automated keratoconus screening with corneal topography analysis," *Investigative Ophthalmology & Visual Science*, vol. 35, pp. 2749–2757, 1994.

[17] M. D. Twa, S. Parthasarathy, C. Roberts et al., "Automated decision tree classification of corneal shape," *Optometry and Vision Science*, vol. 82, no. 12, pp. 1038–1046, 2005.

[18] P. J. Chastang, V. M. Borderie, S. Carvajal-Gonzalez et al., "Automated keratoconus detection using the EyeSys videokeratoscope," *Journal of Cataract & Refractive Surgery*, vol. 26, no. 5, pp. 675–683, 2000.

[19] S. Goebels, T. Eppig, S. Wagenpfeil, A. Cayless, B. Seitz, and A. Langenbucher, "Complementary keratoconus indices based on topographical interpretation of biomechanical waveform parameters: a supplement to established keratoconus indices," *Computational and Mathematical Methods in Medicine*, vol. 2017, Article ID 5293573, 7 pages, 2017.

[20] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Learning*, C. Scholkopf, C. Burges, and A. Smola, Eds., MIT Press, Cambridge, MA, USA, 1998.

[21] D. Anderson, "Understanding corneal topography," *Paraoptometric Resource Center*, vol. 1, pp. 1–12, 2000.

[22] J. J. Rozema, P. Rodriguez, I. Ruiz Hidalgo, R. Navarro, M. J. Tassignon, and C. Koppen, "SyntEyes KTC: higher order statistical eye model for developing keratoconus," *Ophthalmic and Physiological Optics*, vol. 37, no. 3, pp. 358–365, 2017.

[23] J. J. Rozema, P. Rodriguez, R. Navarro, and M.-J. Tassignon, "SyntEyes: a higher-order statistical eye model for healthy eyes," *Investigative Ophthalmology & Visual Science*, vol. 57, pp. 683–691, 2016.

[24] Online: Matlab Software, https://www.mathworks.com/products/matlab.html.

Hindawi
Computational Intelligence and Neuroscience
Volume 2018, Article ID 7913952, 18 pages
https://doi.org/10.1155/2018/7913952

Research Article

Image Processing-Based Recognition of Wall Defects Using Machine Learning Approaches and Steerable Filters

Nhat-Duc Hoang

Lecturer, Faculty of Civil Engineering, Institute of Research and Development, Duy Tan University, P809-03 Quang Trung, Da Nang, Vietnam

Correspondence should be addressed to Nhat-Duc Hoang; hoangnhatduc@dtu.edu.vn

Received 26 August 2018; Revised 12 October 2018; Accepted 18 October 2018; Published 15 November 2018

Guest Editor: Zoran Peric

Copyright © 2018 Nhat-Duc Hoang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Detection of defects including cracks and spalls on wall surface in high-rise buildings is a crucial task of buildings' maintenance. If left undetected and untreated, these defects can significantly affect the structural integrity and the aesthetic aspect of buildings. Timely and cost-effective methods of building condition survey are of practicing need for the building owners and maintenance agencies to replace the time- and labor-consuming approach of manual survey. This study constructs an image processing approach for periodically evaluating the condition of wall structures. Image processing algorithms of steerable filters and projection integrals are employed to extract useful features from digital images. The newly developed model relies on the Support vector machine and least squares support vector machine to generalize the classification boundaries that categorize conditions of wall into five labels: longitudinal crack, transverse crack, diagonal crack, spall damage, and intact wall. A data set consisting of 500 image samples has been collected to train and test the machine learning based classifiers. Experimental results point out that the proposed model that combines the image processing and machine learning algorithms can achieve a good classification performance with a classification accuracy rate = 85.33%. Therefore, the newly developed method can be a promising alternative to assist maintenance agencies in periodic building surveys.

1. Introduction

During the construction and maintenance of high-rise buildings, it is very crucial to attain good surface quality of structures due to safety and esthetics aspects. Because of the combined effects of aging, weather conditions, and human activities, the condition of building structures deteriorates over time [1]. If left untreated, damages such as cracks and spalls obviously cause inconvenience for the building's occupants, deteriorate the structural integrity, and lead to a significant reduction of the value of the poorly maintained asset. Therefore, identifying defective areas that appear on surface structure is one of the main tasks in periodic survey buildings.

In high-rise buildings, the components that have large surface areas typically include walls (both concrete walls and brick walls covered by mortar) and slabs. The assessment of concrete slabs is usually performed during the construction phase. It is because during the operation phase, the surface of slab structure is concealed by floor coverings such as ceramic or stone tiles. Therefore, this study focuses on the visual assessment of wall structures.

In Vietnam, periodic surveys on building condition are usually performed by visual assessment of human inspectors. This fact is also common in other countries because visual changes in structures can directly point out the potential problems of building structures [2]. For instances, cracks can be indicators of structural problems including building settlement and degradation of building materials; particularly for concrete walls, spalls can be caused by the corrosion of embedded reinforcement bars [3–5].

In the current practice of building condition assessment, the damages on the surface of building structures are usually inspected by qualified technicians. These technicians often utilize contact-type equipment including profilometer and

measuring tape for identifying the defective areas [6]. Although the manual procedure can help to obtain accurate condition of the structure, it also has several disadvantages. First, the surveying process is strongly affected by the knowledge, experience, and subjective judgment of human inspectors; therefore, this issue can lead to inconsistency of the assessment outcome. Second, the process of visual assessment, measurement, data processing, and report can be very time consuming especially for high-rise buildings with large surface areas needed to be inspected periodically.

Accordingly, it is immensely beneficial for the building owners and maintenance agencies if the manual inspection process can be replaced by a more productive and consistent method of surveying [7]. Among automated methods for building condition evaluation, machine vision based approaches are widely employed due to their ease of access to equipment, their fast computing processes, and the rapid advancements of image processing techniques [8–14].

Hutchinson and Chen [15] presented a statistical-based method for evaluating concrete damage including cracks and spalls and relied on a Bayesian method for recognizing cracks automatically from images. Chen et al. [16] employed the first derivative of a Gaussian filter to analyze multitemporal images for measuring cracks. Zhu [17] put forward an intelligent method using three circular filters to detect air pockets appearing on the surfaces of concrete.

The level set method and morphological algorithms for image processing have been used by Chen and Hutchinson [18] to identify and analyze cracks in laboratory environment. Lee et al. [19] proposed a model that integrates various image processing operations (brightness adjustment, binarisation, and shape analysis) to facilitate the accuracy of crack detection; in addition, a neural network-based model was implemented to classify crack patterns of cracks. Valença et al. [20] introduced a method based on multispectral image analysis to evaluate and delineate defective areas.

An image processing-based approach for detecting bugholes on concrete surface has been established by Liu and Yang [21]; the employed techniques are contrast enhancement and Otsu thresholding. Kim et al. [22] compared different image binarisation algorithms for identifying cracks in concrete structures. Hoang [23] employed the Otsu method and a gray intensity modification approach for binarizing images and isolating cracks.

Silva and Lucena [24] demonstrated the capability of deep learning approach for concrete crack detection. Dorafshan et al. [25] has recently compared the performances of deep convolutional neural networks and edge detection methods for the task of recognizing concrete cracks. The notable advantage of deep learning approaches is that their feature extraction operators are autonomously constructed during the model training phase [26]. However, methods based on deep learning often necessitate a considerable amount of training samples and require a large computational cost.

As can be seen from the existing literature, most of the previous works have dedicated in constructing models for the classification of crack and noncrack conditions. Few studies have constructed an integrated image processing model for detect cracks and spalls. The objective of the current study is to combine image processing techniques and advanced machine learning algorithms into an integrated model that is capable of detecting and categorizing the defective areas on wall structures. By using an intelligent model that can recognize and categorize types of cracks and spalling areas simultaneously, the task of periodic building condition survey can be executed in a more effective manner.

The feature extraction phase of the new model relies on image processing algorithms of steerable filters and projection integrals. It is because these two algorithms of image processing have demonstrated their usefulness in recognizing defects in pavements [27, 28]. Based on the extracted features, machine learning algorithms including support vector machine (SVM) and least squares support vector machine (LSSVM) are employed to classify input images into five labels: longitudinal cracks, transverse cracks, diagonal cracks, spalls, and intact walls. The reason for the selection of these machine learning approaches is that their outstanding performances in classification tasks have been reported in the literature [29–33].

Based on the aforementioned features, the main contributions of the newly constructed model can be summarized as follows:

(i) Multiple types of cracks (longitudinal cracks, transverse cracks, diagonal cracks) existing in wall structures can be detected and categorized in an integrated model. This can be considered to be a significant improvement since most of the existing models can only produce the prediction outcome of crack or noncrack conditions [13, 23–25].

(ii) Although projection integrals have been utilized in structural defect classification [27, 28], diagonal projection integrals which have been rarely exploited in concrete surface crack categorization are employed in this study to specifically deal with diagonal cracks.

(iii) Cracks and spall damages can be recognized simultaneously in an integrated model which has also rarely been achieved in the current literature.

(iv) Neural networks have been extensively used in concrete crack categorization [19, 34, 35]. However, the applications of SVM and LSSVM in this task are still limited and a comparative work needs to be performed to evaluate the potential of these two advance machine learning algorithms in dealing with the problem of wall defect recognition.

The rest of the paper is organized as follows: the research methodology is reviewed briefly in the next section, followed by the description of the newly constructed automatic approach for wall defect detection; the fourth section reports experimental results of this study, followed by the conclusion in the final section.

2. Research Methodology

2.1. Image Processing Approaches

2.1.1. Steerable Filter (SF). SF [36] is an orientation-selective convolution kernel used widely used for feature extraction. In image processing field, oriented filters are often employed in various vision and image processing tasks including edge detection and texture analysis [37]. Since cracks and spalls on wall surface have distinctive edges and patterns; the utilization of SF can be helpful to recognize these defects.

SF is based on the computation of directional derivatives of Gaussians; accordingly, these filters can be used to construct local orientation maps of a digital image [37]. SF is essentially a linear combination of Gaussian second derivatives. For an image $I(x,y)$, a 2-D Gaussian at a certain pixel is computed in the following formula [38]:

$$G(x, y, r) = \frac{1}{\sqrt{2\pi r}} \exp \frac{-(x^2 + y^2)}{2r^2}, \tag{1}$$

where r is a free parameter which denotes the Gaussian function variance.

The expression of the SF formulation with an orientation of θ is given as follows:

$$F(x, y, r, \theta) = G_{xx} \cos^2(\theta) + 2G_{xy} \cos(\theta) \sin(\theta) + G_{yy} \sin^2(\theta), \tag{2}$$

where G_{xx}, G_{xy}, and G_{yy} denote the Gaussian second derivatives and their expression are given in the following equations:

$$\begin{aligned} G_{xx}(x, y, r) &= \frac{(x^2 - r^2)\exp(-(x^2 + y^2)/2r^2)}{\sqrt{2\pi} r^5}, \\ G_{yy}(x, y, r) &= \frac{(y^2 - r^2)\exp(-(x^2 + y^2)/2r^2)}{\sqrt{2\pi} r^5}, \\ G_{xy}(x, y, r) &= G_{xy}(x, y, r) = \frac{xy \exp(-(x^2 + y^2)/2r^2)}{\sqrt{2\pi} r^5}. \end{aligned} \tag{3}$$

Notably, if the value of the parameter r which is the variance of the Gaussian function variance is fixed, the final response map of an image is obtained by combining the outcomes of individual SFs with different values of θ. In this study, the values of θ vary from 0° to 360° with an interval of 30°. The responses of SFs of images containing defects are demonstrated in Figure 1 with different values of the Gaussian function variance.

In addition, the final response map created by SFs for an image I is calculated using the equation below:

$$R(x, y) = F(x, y, \sigma, \theta) * I(x, y), \tag{4}$$

where "$*$" is the symbol of the convolution operator.

2.1.2. Projection Integral (PI). PI is a widely used method for image analysis. This approach is particularly useful for shape and texture categorization and has been extensively employed for face and facial recognition [39, 40]. In the field of civil engineering, this image analysis method has been successfully applied in pavement crack classification tasks [27, 38, 41] as well as pavement pothole recognition [28].

The two PIs along the horizontal and vertical axes of an image are denoted as horizontal PI (HPI) and vertical PI (VPI). These two PIs are computed according to the following equations:

$$\begin{aligned} \mathrm{HPI}(y) &= \sum_{i \in x_y} I(i, y), \\ \mathrm{VPI}(x) &= \sum_{j \in y_x} I(x, j), \end{aligned} \tag{5}$$

where x_y and y_x are the set of horizontal pixels at the location y and the set of vertical pixels at the location x, respectively (Figure 2).

Since HPI and VPI are incapable of detecting diagonal cracks [42], the two diagonal PIs (DPIs) of an image are employed. The directions of the two DPs, denoted as DPI1 and DPI2, are illustrated in Figure 3. The two DPIs are calculated as follows:

$$\begin{aligned} \mathrm{DPI1}(x, y) &= \sum_{x,y \in D1} I(x, y), \\ \mathrm{DPI2}(x, y) &= \sum_{x,y \in D2} I(x, y), \end{aligned} \tag{6}$$

where $D1$ and $D2$ are the set of pixels along the two diagonal directions of an image (as illustrated in Figure 3).

As illustrated in Figure 4, images containing diagonal cracks have PIs in which there are exceptionally high intensities in the two DPIs. In addition, as can be shown in Figure 5, HPI and VPI have the strongest responses of PI in images having longitudinal and transverse cracks. On the other hand, an image containing a spall damage results in PIs which do not have a significant peak of signal and the average value of its SF response is higher than that of an image without defective areas.

2.2. Support Vector Machine and Least Squares Support Vector Machine. Support vector machine (SVM) is a machine learning based classifier which is established on the basis of the statistical learning theory [43]. The aim of this learning algorithm is to find a predictive function based on the collected data set. The standard version of SVM is designed to cope with binary or two-class pattern-recognition problems. Through the model construction phase, SVM constructs a hyperplane to classify data points so that the distance from it to the nearest data sample of each class label is maximized [44].

Moreover, this algorithm relies on the kernel trick to better deal with nonlinearly separable cases. Using the kernel trick, the data points are mapped from an original input space to a high-dimensional feature space so that linear separability is easier to achieve (Figure 6). Superior

Figure 1: SF responses: (a) longitudinal crack, (b) transverse crack, (c) diagonal crack, (d) spall damage, and (e) intact wall.

Figure 2: Illustration of PIs of an image: (a) HPI and (b) VPI.

Figure 3: Illustration of DPIs of an image.

classification performance of SVM has been widely reported in a large number of previous studies [45–49].

Given a set of training data points $\{x_k, y_k\}_{k=1}^{N}$ with input data $x_k \in R^n$ and a set of class labels $y_k \in \{-1, +1\}$, the model construction phase of SVM is equivalent to solving the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad J_p(w,e) = \frac{1}{2}w^T w + c\frac{1}{2}\sum_{k=1}^{N} e_k^2, \\ &\text{subject to} \quad y_k\left(w^T\varphi(x_k) + b\right) \geq 1 - e_k, \quad k = 1, \ldots, N, e_k \geq 0, \end{aligned} \tag{7}$$

where $w \in R^n$ and $b \in R$ denote the model parameters, $e_k > 0$ represents a slack variable, c denotes a penalty constant which determines severity of learning error, and $\varphi(x)$ denotes a nonlinear mapping from the input space to the feature space.

One of the advantages of SVM is that it does not require expressing the mapping function $\varphi(x)$ explicitly. Due to the concept of kernel trick, the model identification only necessitates the computation of the kernel function $K(.)$ which is the dot product of $\varphi(x)$. The kernel function is shown below:

$$K(x_k, x_l) = \varphi(x_k)^T\varphi(x_l). \tag{8}$$

Radial basis function (RBF) is often selected to be used in SVM [30]; its formula is given as follows:

$$K(x_k, x_l) = \exp\left(-\frac{\|x_k - x_l\|^2}{2\sigma^2}\right), \tag{9}$$

where σ denotes the kernel function parameter.

To solve the aforementioned constrained optimization, the Lagrangian is given as follows:

$$\begin{aligned} L(w,b,e;\alpha;\nu) = J_p(w,e) - \sum_{k=1}^{N}\alpha_k\left\{y_k\left(w^T\varphi(x_k) + b\right) - 1 + e_k\right\} \\ - \sum_{k=1}^{N}\nu_k e_k, \end{aligned} \tag{10}$$

where $\alpha_k \geq 0$, $\nu_k \geq 0$ denote Lagrange multipliers for $k = 1, 2, \ldots, N$.

Accordingly, the conditions for optimality are stated as follows:

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \longrightarrow w = \displaystyle\sum_{k=1}^{N}\alpha_k y_k \varphi(x_k), \\ \dfrac{\partial L}{\partial b} = 0 \longrightarrow \displaystyle\sum_{k=1}^{N}\alpha_k y_k = 0, \\ \dfrac{\partial L}{\partial e_k} = 0 \longrightarrow 0 \leq \alpha_k \leq c, \quad k = 1, \ldots, N. \end{cases} \tag{11}$$

Figure 4: PIs of diagonal cracks.

Based on the equations found by the conditions for optimality, it is able to attain the following dual quadratic programming problem:

$$\max_{\alpha} \quad J_D(\alpha) = -\frac{1}{2}\sum_{k,l=1}^{N} y_k y_l \varphi(x_k)^T \varphi(x_l)\alpha_k\alpha_l + \sum_{k=1}^{N}\alpha_k, \tag{12}$$

$$\text{Subject to} \quad \sum_{k=1}^{N}\alpha_k y_k = 0, \quad 0 \le \alpha_k \le c, \ k = 1, \ldots, N,$$

In addition, the kernel function is applied in the following manner:

$$\omega = y_k y_l \varphi(x_k)^T \varphi(x_l) = y_k y_l K(x_k, x_l). \tag{13}$$

Finally, the classification model based on SVM can be derived as follows:

$$y(x) = \operatorname{sign}\left(\sum_{k=1}^{SV}\alpha_k y_k K(x_k, x_l) + b\right), \tag{14}$$

where SV is the number of support vectors which are training data points that have $\alpha_k > 0$.

Least squares support vector machine (LSSVM) [50] is a least square version of the original SVM. Instead of solving a quadratic programming problem required by SVM, the model construction phase of LSSVM is equivalent to solving a system of linear equation. Therefore, the computational expense of LSSVM can be much lower than that of the standard SVM.

To construct a LSSVM based classifier, it is needed to solve the following minimization problem:

$$\text{minimize} \quad J_p(w, e) = \frac{1}{2}w^T w + \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2,$$

$$\text{subject to} \quad y_k\left(w^T\varphi(x_k) + b\right) = 1 - e_k, \quad k = 1, \ldots, N, \tag{15}$$

where $w \in R^n$ and $b \in R$ are also the model parameters; $e_k \in R$ denote error variables; $\gamma > 0$ is called a regularization constant.

Figure 5: PIs of image samples: (a) longitudinal crack, (b) transverse crack, (c) spall damage, and (d) intact wall.

Figure 6: The learning phase of SVM.

Subsequently, the Lagrangian is applied in the following way:

$$L(w,b,e;\alpha) = J_p(w,e) - \sum_{k=1}^{N} \alpha_k \left\{ y_k \left(w^T \varphi(x_k) + b \right) - 1 + e_k \right\}, \tag{16}$$

where α_k is the kth Lagrange multiplier; $\varphi(x_k)$ represents a nonlinear mapping function.

Relied on the KKT conditions for optimality, it is able to convert the aforementioned constrained optimization problem to a linear system [51]. The classifier based on LSSVM is compactly expressed as follows:

$$y(x) = \operatorname{sign}\left(\sum_{k=1}^{N} \alpha_k y_i K(x_k, x_l) + b \right), \tag{17}$$

where α_k and b are found by solving a linear system. Similar to the standard SVM, $K(x_k, x_l)$ denotes the kernel function.

3. Image Sample Collection

Because the machine learning algorithms of SVM and LSSVM are supervised learning approach, a set of wall images with the corresponding ground truth labels must be prepared in advance of the model construction and classification phases. To establish the required data set, images of walls have been collected during field surveys at high-rise buildings in Da Nang city (Vietnam).

To ease the computational process, the size of each image sample is fixed to be 200 × 200 pixels. Moreover, there are five classes of wall condition, namely, longitudinal cracks (LC), transverse crack (TC), diagonal cracks (DC), spall damage (SD), and intact wall (IW). The number of image samples in each class is 100. Thus, the collected image data set contains 500 samples and is illustrated in Figure 7. It is also noted that all the image samples have been preprocessed by the median filter with a window size of 5 × 5 pixels. This preprocessing step aims at suppressing the noise existing in the collected digital image [52]. In addition, the data set is divided into two folders that contain the training set of images (90%) and the testing set of images (10%). The first set is used in the phase of model construction and the second set is utilized to verify the generalization capability of the wall defect classification model.

4. The Proposed Hybrid Approach of Image Processing and Machine Learning for Detection of Wall Defects

This section describes the proposed model used for automatic classification of wall defects. The overall model structure is illustrated in Figure 8. The model can be divided into two separated modules:

(i) Feature extraction phase that employs the image processing methods of SFs and IPs

(ii) Machine learning based classification phase that relies on the SVM and LSSVM algorithms

In the first step of feature extraction, SFs are employed to compute a salient defect map from a digital image. The minimum and maximum angles of SFs are 0° and 360°, respectively. The parameter r of SFs is selected from a set of [1.0, 1.5, 2.0, 2.5, 3.0]. The value of r which results in the highest accuracy for the training data set is selected as the optimal one. Based on the map generated by SFs, PIs including HPI, VPI, DPI1, and DPI2 are then computed to characterize the texture of the captured images. As earlier mentioned, each image of the data set has the size of 200 × 200 pixels. Thus, each PI contains 200 sampled points and the total number of features to be analyzed by the machine learning algorithm is 200 × 4 = 800. Herein, 4 is the number of PIs.

Obviously, the original PIs can be very rough and have many peaks and valleys due to local fluctuations of the gray intensity of an image. Moreover, the large number of input features can create difficulty for the machine learning algorithms of SVM and LSSVM due to the curse of dimensionality [53]. Therefore, it is beneficial to smooth the original PIs by the use of moving average method [42]. In detail, the average value of W_{PI} consecutive values along the PIs is calculated to create smoothed PIs with fewer data points. For example, if $W_{\text{PI}} = 10$, then the total number of

Figure 7: The collected image samples: (a) longitudinal crack, (b) transverse crack, (c) diagonal crack, (d) spall damage, and (e) intact wall.

features in the contracted PIs is reduced from 800 to 80; if $W_{PI} = 20$, then the machine learning algorithms only have to deal with a data set having 40 features.

The process of feature number reduction using moving average for an image is demonstrated in Figure 9. As can be seen from this example, the smoothed PIs even with the window size $W_{IP} = 20$ still preserve crucial features of rises and ebbs of the original PIs. Therefore, the value of $W_{IP} = 20$ is selected for the feature extraction step. Accordingly, the set of 40 features extracted from PIs is used as input pattern to categorize the four labels of wall defect (LC, TC, DC, and SD) and the label of intact wall (IW).

Furthermore, to obtain the two DPIs of DPI1 and DPI2, the original map of the SF response has been rotated with the angles of +45 and −45 [42]. Accordingly, the two DPIs are derived from the computation of the HPIs of the two rotated SF maps. The process of computing DPIs of images containing diagonal cracks is demonstrated in Figure 10. In addition, the overall feature extraction module is depicted in Figure 11. Based on the 40 input features (IF) extracted from

Figure 8: The proposed model structure.

Figure 9: PIs of an image: (a) the original image, (b) the original PIs ($W_{PI} = 1$), (c) the smoothed PIs ($W_{PI} = 10$), and (d) the smoothed PIs ($W_{PI} = 20$).

the PIs, the machine learning algorithms of SVM and LSSVM are employed to perform the model learning and classification of images in the second module of the model.

It is noted that the standard versions of SVM and LSSVM are designed two-class pattern recognition problems. Hence, the one-versus-one (OvO) strategy [54] has been used with SVM and LSSVM to make them capable of dealing with the five-class recognition tasks at hand. Previous works have confirmed the advantages of OvO strategy in coping with multiclass classification problems [53, 55, 56].

Figure 10: The process of computing DPIs: (a) a −45° diagonal crack and (b) a + 45° diagonal crack.

Figure 11: The whole feature extraction process.

Figure 12: Model performance with different values of the parameter r: (a) LSSVM and (b) SVM.

5. Experimental Results

Because the detection of wall defects is formulated as a five-class pattern recognition problem, classification accuracy rate (CAR) computed for each individual class and for all of the classes is employed. CAR for the class i is computed as follows:

$$\mathrm{CAR}_i = \frac{R_C^i}{R_A^i} \times 100\,(\%), \tag{18}$$

where R_C^i and R_A^i denote the number of data samples in class ith being correctly classified and the total number of data instances in this class, respectively. It is reminded that that there are five class labels in the data set: longitudinal crack (LC), transverse crack (TC), diagonal crack (DC), spall damage (SD), and intact wall (IW).

The overall classification accuracy rate (CAR) for all the five class labels is simply computed as follows:

Table 1: Result comparison.

Statistics	CAR (%)	Classification models					
		LSSVM	SVM	BPANN	CT	LDA	NBC
Average	CAR_{LC}	85.33	79.00	79.33	67.00	74.33	73.00
	CAR_{TC}	83.00	75.67	77.33	68.67	74.67	72.67
	CAR_{DC}	90.00	94.67	63.67	62.67	53.00	54.67
	CAR_{SD}	78.33	84.33	69.33	55.33	31.33	26.00
	CAR_{IW}	89.00	88.00	81.33	57.67	74.67	72.33
	CAR_{O}	85.13	84.33	74.20	62.27	61.60	59.73
Std.	CAR_{LC}	10.42	12.69	11.12	16.01	17.94	13.43
	CAR_{TC}	10.22	15.24	17.21	13.83	12.24	11.43
	CAR_{DC}	11.74	7.30	14.26	13.11	16.43	17.56
	CAR_{SD}	13.41	11.35	17.80	16.97	15.02	12.76
	CAR_{IW}	10.62	8.05	13.58	18.88	12.79	13.05
	CAR_{O}	5.89	5.28	6.75	5.80	6.53	5.63

Figure 13: Result comparison based on CAR_o.

Figure 14: Classification accuracy comparison based on CAR of each class (LD, TC, DC, SD, and IW).

Figure 15: Box plots of overall CARs.

Table 2: p values of the Wilcoxon signed-rank test.

	LSSVM	SVM	BPANN	CT	LDA	NBC
LSSVM	0.00000	0.60134	0.00001	0.00000	0.00000	0.00000
SVM	0.60134	0.00000	0.00002	0.00000	0.00000	0.00000
BPANN	0.00001	0.00002	0.00000	0.00000	0.00000	0.00000
CT	0.00000	0.00000	0.00000	0.00000	0.06728	0.00444
LDA	0.00000	0.00000	0.00000	0.06728	0.00000	0.02230
NBC	0.00000	0.00000	0.00000	0.00444	0.02230	0.00000

$$\mathrm{CAR}_{\mathrm{Overall}} = \sum_{i=1}^{5} \frac{\mathrm{CAR}_i}{5}. \tag{19}$$

As stated earlier, the data set including 500 samples is employed to train and test the wall defect classification model. This data set is randomly separated into two subsets: data for model training (90%) and data for testing (10%). Because a single run of model training and testing may not help to reveal the true performance of a classifier, this study performs a repeated subsampling process that includes 20 times of model training and prediction. In each time of running, 10% of the data set is randomly extracted to form the testing data subset; the rest of the data set is reserved for model testing phase.

As described in the formulation of the two machine learning algorithms of SVM and LSSVM, these two algorithms both require a proper setting of their tuning parameters. In the case of SVM, the tuning parameters are the penalty coefficient and the kernel function parameter. In the case of LSSVM, the regularization and the kernel function parameters need to be selected appropriately. In this section, the grid search method described in the previous work of [57] is employed for automatically setting those tuning parameters of SVM and LSSVM. In addition, the feature selection stage requires the setting of the parameter r in the computation of SFs. The model performance with different values of the parameter r for the cases of SVM and LSSVM is reported in Figure 12. As can be observed from this figure, $r = 2$ results in the highest $\mathrm{CAR}_{\mathrm{Overall}}$ values for both SVM and LSSVM.

Besides the two machine learning algorithms of SVM and LSSVM, the backpropagation artificial neural network (BPANN), classification tree (CT), linear discriminant analysis (LDA), and naive Bayes classifier (NBC) are also employed as benchmark classifiers. It is also noted that CT, LDA, and NBC are also equipped with the OvO strategy to deal with the current five-class recognition problem of wall defect classification. The SVM and the benchmark algorithms implemented in MATLAB with the help of the statistics and machine learning toolbox [58]. In addition, the LSSVM model is constructed by the built-in functions provided in the toolbox developed by De Brabanter et al. [59].

In addition, the training phases of BPANN and CT require a proper setting of their model hyper-parameters. Similar to the cases of SVM and LSSVM, the model hyper-parameters of those models that result in the best performance for the testing set are selected. In the case of the DT model, the suitable value of the minimal number of observations per tree leaf is found to be 2. The appropriate structure of the BPANN model consists of 35 neurons in the hidden layers; furthermore, the scaled conjugate gradient algorithm with the maximum number of training epochs = 3000 is used to train the neural network model. Moreover, the processes of selecting an appropriate value of the tuning parameter r used in constructing the SF maps of the benchmark models of BPANN, CT, LDA, and NBC are

Figure 16: Examples of incorrect classifications.

similar to those of the SVM and LSSVM models. Based on result comparisons, the suitable value of the tuning parameter r used with BPANN, CT, LDA, and NBC is also 2.

The performances of the machine learning classifiers used for wall damage recognition are summarized in Table 1. Observed from this table, LSSVM has achieved the best predictive performance in terms of CAR_o (85.13%), followed by SVM (CAR_o = 84.33%), BPANN (CAR_o = 74.20%), CT (CAR_o = 62.27%), LDA (CAR_o = 61.00%), and NBC (CAR_o = 59.73%). Thus, it can be seen that classifiers based on LSSVM and SVM are more suitable for the task of wall defect classification than other machine learning and statistical approaches of BPANN, CT, LDA, and NBC.

Figures 13 and 14 graphically compare the results obtained from the prediction models in terms of CAR_o and CAR of each individual class label. It is shown that the CARs of LC (85.33%), TC (83.00%), and IW (89.00%) obtained from LSSVM are higher than those yielded by SVM (79.00%, 75.67%, and 88.00% for the LC, TC, and IW, respectively). However, results of SVM for the classes of DC (94.67%) and SD (84.33%) are better than those of LSSVM (90.00% and 78.33% for the classes of DC and SD, respectively).

In addition, the classification results of all the models in terms of CAR_o are displayed by the box plots in Figure 15. Moreover, to better demonstrate the statistical difference between each pair of classifiers employed in the task of wall defect recognition, the Wilcoxon signed-rank test (WSRT) is utilized in this section. WSRT is a nonparametric statistical hypothesis test that is widely used for verifying the statistical difference of model performances [57]. With the significance level of the test = 0.05, if p value computed from the test is smaller than 0.05, it can be confirmed that the performances of the two selected classifiers are statistically different. The p values obtained from WSRT for each pair of classifiers are reported in Table 2. The outcomes shown in this table confirm that LSSVM and SVM are significantly better than other benchmark models in the task of recognizing wall damages. In addition, with p values = 0.60134, there is no statistical difference between the performances of LSSVM and SVM.

Although the LSSVM model has delivered the highest CARs, this model also commits wrong classification cases. These misclassifications are investigated and examples of them are illustrated in Figure 16. In Figures 16(a) and 16(b), images with the ground truth label of LC and TC have been assigned the label of IW. The reasons for these misclassifications are that the crack objects are too thin; moreover, there is a line of stain existing in Figure 16(b). The case in Figure 16(c) shows an image with its ground truth class of DC which has been categorized as the class of TC. The possible reason of this phenomenon is that the crack object appears too close to corner of the image; therefore, the signals of the SF responses of the two DPIs are not significantly stronger than those of other PIs. In Figure 16(d), an image with complex background texture has caused the machine to misclassify an image with the ground truth category of SD. Moreover, an object of stain (Figure 16(e)) leads to the classification of an image into the class of LC while it actually belongs to the class of IW.

6. Conclusion

This study has proposed an automatic approach for periodic survey of concrete wall structures. The newly constructed approach mainly consists of the feature extraction step and the pattern classification step. In the first step, image processing techniques of SF and PI have been employed to characterize the texture and the pattern existing in images. In the second step, machine learning algorithms of SVM and LSSVM have been used to analyze the features extracted by the image processing techniques and to assign input images into one of the five class labels of LC, TC, DC, SD, and IW. Experimental results using a repeated random subsampling with 20 runs show that the predictive performances of LSSVM (CAR_o = 85.13%) and SVM (CAR_o = 84.33%) are superior to other benchmark models of BPANN, CT, LDA, and NBC. These facts confirm that the proposed approach can be a time- and cost-effective solution for the task of building periodic survey. The future developments of the current study include the integration of other advanced image processing methods (e.g., image segmentation, color/texture analyses) to enhance the CARs via the reduction of falsely classified cases.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interests regarding the publication of this research work.

Supplementary Materials

The supplementary file contains the data set used in this study. In this file, the first 500 columns are the input features (X1 to X40) of the data (which are the smoothed projection integrals); the last column is the class labels (1 = longitudinal crack, 2 = transverse crack, 3 = diagonal crack, 4 = spall damage, and 5 = intact wall). (*Supplementary Materials*)

References

[1] W. Zhang, Z. Zhang, D. Qi, and Y. Liu, "Automatic crack detection and classification method for subway tunnel safety monitoring," *Sensors*, vol. 14, no. 10, pp. 19307–19328, 2014.

[2] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting Multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.

[3] T. Dawood, Z. Zhu, and T. Zayed, "Machine vision-based model for spalling detection and quantification in subway networks," *Automation in Construction*, vol. 81, pp. 149–160, 2017.

[4] S. German, I. Brilakis, and R. DesRoches, "Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety

assessments," *Advanced Engineering Informatics*, vol. 26, no. 4, pp. 846–858, 2012.

[5] M.-K. Kim, H. Sohn, and C.-C. Chang, "Localization and quantification of concrete spalling defects using terrestrial laser scanning," *Journal of Computing in Civil Engineering*, vol. 29, no. 6, article 04014086, 2015.

[6] P. Tang, D. Huber, and B. Akinci, "Characterization of laser scanners and algorithms for detecting flatness defects on concrete surfaces," *Journal of Computing in Civil Engineering*, vol. 25, no. 1, pp. 31–42, 2011.

[7] H. Kim, E. Ahn, M. Shin, and S.-H. Sim, "Crack and noncrack classification from concrete surface images using machine learning," *Structural Health Monitoring*, vol. 0, article 1475921718768747, 2018.

[8] R. Davoudi, G. R. Miller, and J. N. Kutz, "Structural load estimation using machine vision and surface crack patterns for shear-critical RC beams and slabs," *Journal of Computing in Civil Engineering*, vol. 32, no. 4, article 04018024, 2018.

[9] J.-Y. Jung, H.-J. Yoon, and H.-W. Cho, "A study on crack depth measurement in steel structures using image-based intensity differences," *Advances in Civil Engineering*, vol. 2018, Article ID 7530943, 10 pages, 2018.

[10] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.

[11] C. Koch, S. G. Paal, A. Rashidi, Z. Zhu, M. König, and I. Brilakis, "Achievements and challenges in machine vision-based inspection of large concrete structures," *Advances in Structural Engineering*, vol. 17, no. 3, pp. 303–318, 2014.

[12] H. Pragalath, S. Seshathiri, H. Rathod, B. Esakki, and R. Gupta, "Deterioration assessment of infrastructure using fuzzy logic and image processing algorithm," *Journal of Performance of Constructed Facilities*, vol. 32, no. 2, article 04018009, 2018.

[13] Y.-S. Yang, C.-l. Wu, T. T. C. Hsu, H.-C. Yang, H.-J. Lu, and C.-C. Chang, "Image analysis method for crack distribution and width estimation for reinforced concrete structures," *Automation in Construction*, vol. 91, pp. 120–132, 2018.

[14] Z. Zhu, S. German, and I. Brilakis, "Detection of large-scale concrete columns for automated bridge inspection," *Automation in Construction*, vol. 19, no. 8, pp. 1047–1055, 2010.

[15] T. C. Hutchinson and Z. Chen, "Improved image analysis for evaluating concrete damage," *Journal of Computing in Civil Engineering*, vol. 20, no. 3, pp. 210–216, 2006.

[16] L.-C. Chen, Y.-C. Shao, H.-H. Jan, C.-W. Huang, and Y.-M. Tien, "Measuring system for cracks in concrete using multitemporal images," *Journal of Surveying Engineering*, vol. 132, no. 2, pp. 77–82, 2006.

[17] Z. Zhu and I. Brilakis, "Detecting air pockets for architectural concrete quality assessment using visual sensing," *Electronic Journal of Information Technology in Construction*, vol. 13, pp. 86–102, 2008.

[18] Z. Chen and T. C. Hutchinson, "Image-based framework for concrete surface crack monitoring and quantification," *Advances in Civil Engineering*, vol. 2010, Article ID 215295, 18 pages, 2010.

[19] B. Y. Lee, Y. Y. Kim, S.-T. Yi, and J.-K. Kim, "Automated image processing technique for detecting and analysing concrete surface cracks," *Structure and Infrastructure Engineering*, vol. 9, no. 6, pp. 567–577, 2013.

[20] J. Valença, L. M. S. Gonçalves, and E. Júlio, "Damage assessment on concrete surfaces using multi-spectral image analysis," *Construction and Building Materials*, vol. 40, pp. 971–981, 2013.

[21] B. Liu and T. Yang, "Image analysis for detection of bugholes on concrete surface," *Construction and Building Materials*, vol. 137, pp. 432–440, 2017.

[22] H. Kim, E. Ahn, S. Cho, M. Shin, and S.-H. Sim, "Comparative analysis of image binarization methods for crack identification in concrete structures," *Cement and Concrete Research*, vol. 99, pp. 53–61, 2017.

[23] N.-D. Hoang, "Detection of surface crack in building structures using image processing technique with an improved Otsu method for image thresholding," *Advances in Civil Engineering*, vol. 2018, Article ID 3924120, 10 pages, 2018.

[24] W. R. L. de Silva and D. S. d. Lucena, "Concrete cracks detection based on deep learning image classification," *Proceedings*, vol. 2, no. 8, p. 489, 2018.

[25] S. Dorafshan, R. J. Thomas, and M. Maguire, "Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete," *Construction and Building Materials*, vol. 186, pp. 1031–1045, 2018.

[26] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

[27] A. Cubero-Fernandez, F. J. Rodriguez-Lozano, R. Villatoro, J. Olivares, and J. M. Palomares, "Efficient pavement crack detection and classification," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 39, 2017.

[28] N.-D. Hoang, "An artificial intelligence method for asphalt pavement pothole detection using least squares support vector machine and neural network with steerable filter-based feature extraction," *Advances in Civil Engineering*, vol. 2018, Article ID 7419058, 12 pages, 2018.

[29] G. M. Hadjidemetriou, P. A. Vela, and S. E. Christodoulou, "Automated pavement patch detection and quantification using support vector machines," *Journal of Computing in Civil Engineering*, vol. 32, no. 1, article 04017073, 2018.

[30] B. T. Pham, A. Jaafari, I. Prakash, and D. T. Bui, "A novel hybrid intelligent model of support vector machines and the MultiBoost ensemble for landslide susceptibility modeling," *Bulletin of Engineering Geology and the Environment*, 2018, In press.

[31] H. Shin and J. Paek, "Automatic task classification via support vector machine and crowdsourcing," *Mobile Information Systems*, vol. 2018, Article ID 6920679, 9 pages, 2018.

[32] M. Wang, Y. Wan, Z. Ye, and X. Lai, "Remote sensing image classification based on the optimal support vector machine and modified binary coded ant colony optimization algorithm," *Information Sciences*, vol. 402, pp. 50–68, 2017.

[33] Y. Zhou, W. Su, L. Ding, H. Luo, and P. E. D. Love, "Predicting safety risks in deep foundation pits in subway infrastructure projects: support vector machine approach," *Journal of Computing in Civil Engineering*, vol. 31, no. 5, article 04017052, 2017.

[34] G. K. Choudhary and S. Dey, "Crack detection in concrete surfaces using image processing, fuzzy logic, and neural networks," in *Proceedings of 2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, pp. 404–411, Nanjing, China, October 2012.

[35] A. Mohan and S. Poobal, "Crack detection using image processing: a critical review and analysis," *Alexandria Engineering Journal*, vol. 57, no. 2, 2017.

[36] W. T. Freeman and E. H. Adelson, "Steerable filters for early vision, image analysis, and wavelet decomposition," in *Proceedings of Third International Conference on Computer Vision*, pp. 406–415, Osaka, Japan, December 1990.

[37] E. P. Simoncelli and H. Farid, "Steerable wedge filters," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 189–194, Cambridge, MA, USA, June 1995.

[38] N.-D. Hoang and Q.-L. Nguyen, "A novel method for asphalt pavement crack classification based on image processing and machine learning," *Engineering with Computers*, 2018, In press

[39] A. Chouchane, M. Belahcene, and S. Bourennane, "3D and 2D face recognition using integral projection curves based depth and intensity images," *International Journal of Intelligent Systems Technologies and Applications*, vol. 14, no. 1, pp. 50–69, 2015.

[40] A. Hernandez-Matamoros, A. Bonarini, E. Escamilla-Hernandez, M. Nakano-Miyatake, and H. Perez-Meana, "Facial expression recognition with automatic segmentation of face regions using a fuzzy based classification approach," *Knowledge-Based Systems*, vol. 110, pp. 1–14, 2016.

[41] S. Li, Y. Cao, and H. Cai, "Automatic pavement-crack detection and segmentation based on steerable matched filtering and an active contour model," *Journal of Computing in Civil Engineering*, vol. 31, no. 5, article 04017045, 2017.

[42] N.-D. Hoang and Q.-L. Nguyen, "Automatic recognition of asphalt pavement cracks based on image processing and machine learning approaches: a comparative study on classifier performance," *Mathematical Problems in Engineering*, vol. 2018, Article ID 6290498, 16 pages, 2018.

[43] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, Inc, Hoboken, NJ, USA, 1998, ISBN-10: 0471030031.

[44] L. H. Hamel, *Knowledge Discovery with Support Vector Machines*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2009, ISBN: 978-0-470-37192-3.

[45] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296–303, 2017.

[46] W. Chen, H. R. Pourghasemi, and S. A. Naghibi, "A comparative study of landslide susceptibility maps produced using support vector machine with different kernel functions and entropy data mining models in China," *Bulletin of Engineering Geology and the Environment*, vol. 77, no. 2, pp. 647–664, 2018.

[47] H. Hasni, A. H. Alavi, P. Jiao, and N. Lajnef, "Detection of fatigue cracking in steel bridge girders: a support vector machine approach," *Archives of Civil and Mechanical Engineering*, vol. 17, no. 3, pp. 609–622, 2017.

[48] B. Kalantar, B. Pradhan, S. A. Naghibi, A. Motevalli, and S. Mansor, "Assessment of the effects of training data selection on the landslide susceptibility mapping: a comparison between support vector machine (SVM), logistic regression (LR) and artificial neural networks (ANN)," *Geomatics, Natural Hazards and Risk*, vol. 9, no. 1, pp. 49–69, 2018.

[49] A. Saxena and S. Shekhawat, "Ambient air quality classification by grey wolf optimizer based support vector machine," *Journal of Environmental and Public Health*, vol. 2017, Article ID 3131083, 12 pages, 2017.

[50] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[51] J. Suykens, J. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, *Least Square Support Vector Machines*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2002, ISBN-13: 978-9812381514.

[52] H. Rababaah, "Asphalt pavement crack classification: a comparative study of three ai approaches: multilayer perceptron, genetic algorithms, and self-organizing maps," M.S. Thesis, Indiana University South Bend, South Bend, IN, USA, 2005.

[53] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics*, Springer, Berlin, Germany, ISBN-10: 0387310738, 2011.

[54] A. Rocha and S. K. Goldenstein, "Multiclass from binary: expanding one-versus-all, one-versus-one and ECOC-based approaches," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 289–302, 2014.

[55] K.-B. Duan, J. C. Rajapakse, and M. N. Nguyen, *One-Versus-One and One-Versus-All Multiclass SVM-RFE for Gene Selection in Cancer Classification*, pp. 47–56, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[56] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, pp. 415–425, 2002.

[57] N.-D. Hoang and D. T. Bui, "Predicting earthquake-induced soil liquefaction based on a hybridization of kernel Fisher discriminant analysis and a least squares support vector machine: a multi-dataset study," *Bulletin of Engineering Geology and the Environment*, vol. 77, no. 1, pp. 191–204, 2018.

[58] Matwork, *Statistics and Machine Learning Toolbox User's Guide*, Matwork Inc., Natick, MA, USA, 2017, https://www.mathworks.com/help/pdf_doc/stats/stats.pdf.

[59] K. De Brabanter, P. Karsmakers, F. Ojeda et al., *LS-SVMlab Toolbox User's Guide Version 1.8*, 2010.