

## *Retraction*

# **Retracted: Media Content Mining Based on Artificial Intelligence and Network Interaction**

### **Mobile Information Systems**

Received 26 September 2023; Accepted 26 September 2023; Published 27 September 2023

Copyright © 2023 Mobile Information Systems. 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.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

### **References**

- [1] K. Kang, Z. Wang, and H. Zhang, "Media Content Mining Based on Artificial Intelligence and Network Interaction," *Mobile Information Systems*, vol. 2022, Article ID 6705986, 6 pages, 2022.

## Research Article

# Media Content Mining Based on Artificial Intelligence and Network Interaction

Kai Kang , Zhixue Wang , and Hongwu Zhang 

School of Mathematics and Computer Science, Ningxia Normal University, Ningxia, Guyuan 756000, China

Correspondence should be addressed to Kai Kang; 20120053@stumail.hbu.edu.cn

Received 12 May 2022; Revised 14 June 2022; Accepted 18 June 2022; Published 30 June 2022

Academic Editor: Muhammad Muzammal

Copyright © 2022 Kai Kang 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.

In recent years, due to the explosive growth of social media information, mining hot information in social media has become a research direction of great concern. In this paper, *Python* crawler technology is used to crawl the semi-structured text data of food safety news from static web pages and dynamic web pages. After preprocessing, the structured text data required to establish a document clustering algorithm (CASC) based on a convolutional neural network is obtained. Using the feature extraction ability of convolutional neural network and self-encoder, while preserving the internal structure of the original data to the greatest extent, it is embedded into the low-dimensional potential space for clustering. Finally, it is compared with the performance of the K-means algorithm and spectral clustering algorithm. The experimental results show that the CASC algorithm reduces the running time and time complexity of the algorithm on the premise of ensuring clustering accuracy. The CASC algorithm is superior to the k-means algorithm and spectral clustering algorithm in precision, recall, and composite index. At the same time, the running time is 91 seconds faster than the K-Means algorithm and 5 seconds faster than the spectral clustering algorithm.

## 1. Introduction

Nowadays, media convergence has become a trend. The construction of media convergence platform provides new channels and ideas for media content management and services, and the all media content library is its key part, which is conducive to the overall management of all media resources. The role of all media content libraries is mainly reflected in the sharing, retrieval, and call of all media information, which enables the deeper development and utilization of all media resources on the platform and also plays a strong supporting role in all media operation, planning, and production. Nowadays, all kinds of new media are developing rapidly, and the network environment is more complex. How to help users obtain valuable media content, optimize the media content service process, and build an industry visual model library has become a problem to be solved. Facing the huge all media content, it is difficult for users to obtain useful information only by relying on traditional classification, search, and other functions. They need to use more intelligent tools to improve retrieval

efficiency and mine valuable content. We should provide accurate services according to users' needs. When users' goals are determined, we should help them get the target content quickly and accurately with the help of search tools. If users are uncertain about their own needs, we should reflect the characteristics of all media content intelligent service, actively and intelligently analyze customers' needs, and help customers gradually clarify and get the required content [1–4]. Figure 1 shows a device and method based on artificial intelligence network interaction.

## 2. Literature Review

For social network cross media big data, it is very difficult to search specific text and images. In the face of online social networks, data forms are far from limited to text, images, and videos. Wang et al. extracted the image features by using the dimensionality reduction method of principal component analysis (PCA), normalized the image data into a digital matrix, and obtained the eigenvalues and eigenvectors in the digital matrix by using the method of orthogonal

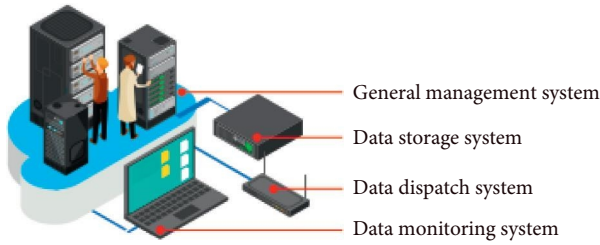


FIGURE 1: A device and method based on artificial intelligence network interaction.

transformation. The image data are represented by the feature vector corresponding to the maximum eigenvalues obtained by orthogonal transformation [5]. Xie Daoping used the multiscale method (MDS) to model the image data, mine the potential feature relationship between the image data, and use the dimension reduction method to map the image data in low dimension, and the obtained result is used as the feature vector of the image data [6]. Mukhejee et al. proposed the method of association rule mining to mine a large amount of data stored in structured database [7]. Gan et al. proposed an Apriori algorithm for rule mining of structured databases, which uses an algorithm idea of information co-occurrence to make statistics on structured data and mine the potential association relationship between data [8]. There is a semantic gap between big data across media. Benesty et al. use canonical correlation analysis (CCA) to solve the problem of semantic gap between cross media data. The core idea is to map the correlation between text data and image data and calculate the similarity between cross media data in the mapped public space. The search of cross media big data in social networks is realized through the similarity between cross media big data in public space [9]. Dai Gang et al. first used the dimensionality reduction method to represent the low-dimensional vector of cross media big data, then used the CCA analysis algorithm to model the isomorphic space between cross media big data, and selected the appropriate similarity calculation method to model the CCA isomorphic space [10]. Compared with traditional information media such as news and newspapers, the cross media data generated by social networks has the characteristics of complex structure, difficult processing, and more noise information, which makes it difficult to process using traditional data mining technology. Traditional methods include evolution mining based on statistics and evolution mining based on logical analysis. The evolution mining method based on statistics can make various statistics on big data and conduct stage analysis according to the statistical results, so as to realize the evolution law mining of topics in data. The evolutionary mining method based on logical analysis needs to logically analyze a large number of data, which is subject to the logical ability and knowledge range of data analysts [11, 12].

This paper proposes a research method of media content mining based on the interaction between artificial intelligence and network, which crawls the required media text data on the network. Using a convolution self-coding text feature extraction algorithm, the self-coding algorithm is

combined with convolution neural network, and the intermediate result of self-coding algorithm is used as the input of convolution neural network to explore the syntax and semantic structure in the text. Then, the cross media data are modeled by deep learning algorithm, and the data of different modes are associated for subsequent search to realize the retrieval between cross media data.

### 3. Research Methods

**3.1. Introduction to Web Crawler.** Web crawler technology is a technology that can automatically extract a large amount of information from the network by simulating users' web browsing behavior. It is the core of network information retrieval. Web crawler usually contains at least three functional modules: network request module, process control module, and content analysis and extraction module. Writing web crawler program requires specific assembly language. Assembly language supporting network communication can write crawler program [13]. The working principle of web crawler is shown in Figure 2.

The web crawler needs to set a URL queue of the target web page according to the required target data [14], crawl the data from the Internet through the target queue, send a request to the Internet web page according to the URL, wait for the web page response and feedback information, and crawl the information content into the downloaded Web page library. When crawling, get a new web page URL from the Internet and expand the target URL queue. After the URL crawling in each URL queue is completed, it enters the crawled URL queue, filters out the failed URLs and the newly generated URLs from the crawled URL queue, and expands them to the target URL queue. Finally, all URLs in the target URL queue are crawled, and all target data are crawled and saved in the downloaded Web page library.

Among many programming languages, *Python* is the most widely used scripting language for writing web crawler programs. Based on *Python* language, there are many excellent libraries and crawler frameworks, such as *scraper*, *beautiful soup*, *Crawley*, *Python* *goose*, and *mechanize* [15, 16]. *Python*, an assembly language, will provide a web page interface that can be used directly to facilitate the operation of crawler code. After crawler crawls web page data, it can temporarily store the page source code in the *Python* interface, or directly download the page source code locally to facilitate subsequent analysis of web page content. *Python* crawlers should include at least three main parts:

- (1) URL queue manager: the main function of this part is to manage the URL queue to be crawled in the *Python* crawler, including the original crawled URL and the URL queue to be crawled, as well as the update of the new URL queue.
- (2) Web source crawler: the function of this part is to crawl the web page source code, send a request to the target web page according to the web page URL provided in the previous step, crawl the web page source code and convert it into a string format after

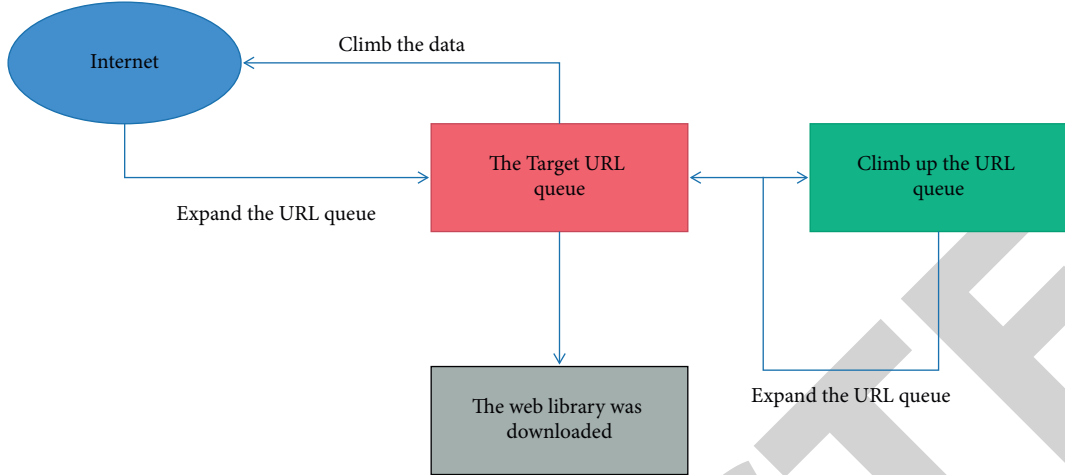


FIGURE 2: Schematic diagram of reptile.

the target web page responds, and store the web page source code in the *Python* interface temporarily after successful crawling.

- (3) Web content parser: the function of this part is to analyze the web page source code temporarily stored in the *Python* interface or downloaded to the local web page on the basis of the completion of web page source code crawling, and extract the useful information in the web page according to the requirements of web page content.

### 3.2. Introduction to Convolutional Self Encoder.

Autoencoder is a kind of feedforward neural network. The simplest autoencoder consists of three layers: input layer, hidden layer, and output layer, which are called the neural network from the input layer to the hidden layer. The number of input nodes of the encoder is equal to the number of output nodes of the decoder. The purpose is to learn an identity map through training to make the input equal to the output as much as possible, so as to find out the potential hidden correlation between the original data. Suppose that the original data are represented by  $x = \{x_1, x_2, \dots, x_m\}$ , the output of the decoder is represented by  $\tilde{x} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m\}$ , the middle hidden layer is represented by  $h = \{h_1, h_2, \dots, h_k\}$ , the mapping functions are represented by  $\sigma_e$  and  $\sigma_d$ , the connection weights of the two adjacent layers are represented by  $w^{(1)} = \{w_{11}^{(1)}, w_{12}^{(1)}, \dots, w_{km}^{(1)}\}$  and  $w^{(2)} = \{w_{11}^{(2)}, w_{12}^{(2)}, \dots, w_{km}^{(2)}\}$ , and the offset terms are represented by  $b^{(1)} = \{b_1^{(1)}, b_2^{(1)}, \dots, b_k^{(1)}\}$  and  $b^{(2)} = \{b_1^{(2)}, b_2^{(2)}, \dots, b_m^{(2)}\}$ , respectively. Through the training of back propagation algorithm, the connection weight and offset term between the input layer, the output layer, and the middle hidden layer are continuously adjusted to make  $\tilde{x} = \sigma_d(\sigma_e(x)) \approx x$ ; that is, the reconstructed output obtained by the decoder is equal to the original data as much as possible. The output  $h_j$  of the encoder and the reconstructed output of the final original data  $\tilde{x}$  are shown in equations (1) and (2), respectively:

$$h_j = \sigma \left( \sum_{i=1}^n w_{ji}^{(1)} x_{(i)} + b_j^{(1)} \right), \quad (1)$$

$$\tilde{x}_j = \sigma \left( \sum_{i=1}^k w_{ij}^{(2)} h_{(j)} + b_i^{(2)} \right). \quad (2)$$

The goal of training the self-encoder is to minimize the error  $L$  between the original data and the reconstructed data. In document clustering, since the document has been represented as a continuous vector, the square error can be used to calculate  $L$ , as shown in the following equation:

$$L = \sum \|\tilde{x} - x\|^2 + \eta \|W\|^2. \quad (3)$$

The second term is the regular term, which is used to avoid the over fitting problem of the model, and  $\eta$  is the regular coefficient [17, 18].

Self-encoder can be said to be an algorithm idea, and its encoder and decoder can use any form of neural network to form different self-encoders. Because the convolution neural network can effectively extract the internal features of the original data in layers, it can be used to form the encoder part and the deconvolution network to form the decoder part, that is, a convolution self-encoder.

Suppose the convolution self-encoder has  $N$  convolution cores, the  $k$  convolution core is  $n^k$ , and the offsets are  $b^k$  and  $c^k$ , respectively [19–21]. The other parameters are the same as those of the self encoder. For the input data  $x$ , the input of the  $k$  convolution core and the reconstructed data  $\tilde{x}$  of the decoding layer are shown in equations (4) and (5), respectively:

$$h^k = \sigma(x * W^k + b^k), \quad (4)$$

$$\tilde{x} = \sigma \left( \sum_{j=1}^N h^k * \overline{W}^k + c^k \right), \quad (5)$$

where  $\overline{W}^k$  represents the transposition of the weight matrix  $W^k$  of the  $k$ th characteristic graph and  $*$  represents the two-dimensional convolution. The training method of



convolutional self-encoder is the same as that of self-encoder.

**3.3. Construction of CASC Model.** The CASC model is mainly composed of three parts: document preprocessing, document vector embedding representation, and clustering. Document preprocessing mainly includes Chinese word segmentation, removal of stop words, and document vector representation. Chinese word segmentation and removal of stop words are completed by word segmentation library Jieba, and document vector representation is based on word2vec. Word2vec is a method to obtain the vector representation of words by training neural network. It has two implementation methods: CBOW and skip Gram. This paper uses skip Gram method to obtain the word vector. This method predicts the words around the word according to the current word, so as to learn the vector representation of the current word. After obtaining the word vectors of all words, each document is regarded as a combination of the words contained, and the word vectors of the words contained in the document are stacked to form a document word matrix to represent the document. Assuming that  $D_j$  represents the vector representation of document  $j$ ,  $n_j$  represents the number of words contained in document  $j$ , and  $w_i$  represents the word vector obtained through skip Gram training, the final  $n_j \times d$  document word matrix is shown in the following equation:

$$D_j = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1d} \\ w_{21} & w_{22} & \cdots & w_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n_j,1} & w_{n_j,2} & \cdots & w_{n_j,d} \end{bmatrix}. \quad (6)$$

Generally speaking, the document vector obtained above has high dimension and contains more redundant information. If it is clustered directly, it is difficult to capture its internal potential correlation. Therefore, in the CASC model, the convolutional self-encoder is used to embed the obtained high-dimensional document matrix into the low-dimensional potential vector space through training and learning, so as to reduce the vector dimension and preserve the internal structure of the original data to the greatest extent, so as to shorten the time required for clustering. After embedding the document matrix into the low-dimensional potential space, the obtained low-dimensional vector representation is used for spectral clustering, and then the final clustering result is obtained [22–25].

**3.4. Experimental Setup.** In order to test the clustering effect of the CASC model, this paper uses the data set composed of crawling food safety news. The key technology used in the data extraction module is *Python* crawler technology. Because different food safety websites and major news websites include static pages and dynamic pages. The crawler program is written in *Python* language. First, the static web page is crawled. After the static web page is successfully crawled, the dynamic web page is crawled, and the project source code

required for web page analysis is obtained. On the basis of successful crawling of the source code, analyze the web page of the source code, compare the source code and the original web page, and find out the corresponding part of the project related information in the source code. The first mock exam is programmed automatically according to its compilation rules. After the data are preprocessed, the data are stored in local folder in form so that the next module can be used. Add regular monitoring at the beginning of the program to detect whether there is updated project information and supplement the crawled new web pages in real time.

Run the web crawler process through *Python*, as shown in Figure 3.

According to the flow chart, at the beginning of crawling, you need to obtain a specific URL to distinguish between static pages and dynamic pages. The crawler simulates the browser to send a request to the target web page through the HTTP protocol. After the web page passes a series of verification measures, it responds to the request and feeds back the web page content to the *Python* program in the form of web page source code. The obtained web page source code is temporarily stored in the *Python* interface or directly downloaded to the local file. On the basis of obtaining the source code, start parsing the web page content. The parsing process also needs to distinguish between static web pages and dynamic web pages. Through some third-party libraries in *Python*, the static content and dynamic content are parsed, respectively. The parsed web page content required by the target is stored in the local file and the next web page URL for crawling is filtered. After filtering, if there is a new URL, repeat the above crawler steps to realize the automatic operation of the crawler. If there is no new URL, end the whole crawler.

A total of 11530 news document datasets are obtained by *Python* crawler, which is manually divided into 8 categories, and the number of clusters is the same. The data set is preprocessed with Chinese word segmentation and removal of stop words, in which Jieba thesaurus is used for word segmentation. After preprocessing, a total of 3932805 words remain, and 95334 independent words remain after de duplication. The experimental environment adopts the Windows 7 flagship 64-bit operating system and *Python* 3.5.2.

In the experiment, the convolutional self-encoder has four layers, including two identical convolution layers, middle hidden layer and output layer. The excitation function of convolution layer and hidden layer uses the ReLU function in the form of  $f(x) = \max(0, x)$ , and batch normalization is used to avoid over fitting.

In this paper, RMSProp optimization algorithm with learning rate of 0.001 is used to train convolutional self-encoder, and the number of training steps is 10000. Other super parameters are shown in Table 1. This algorithm is compared with the classical k-means algorithm and spectral clustering algorithm.

## 4. Result Analysis

This paper uses precision, recall, F1 value, and algorithm running time as the evaluation criteria of the model.

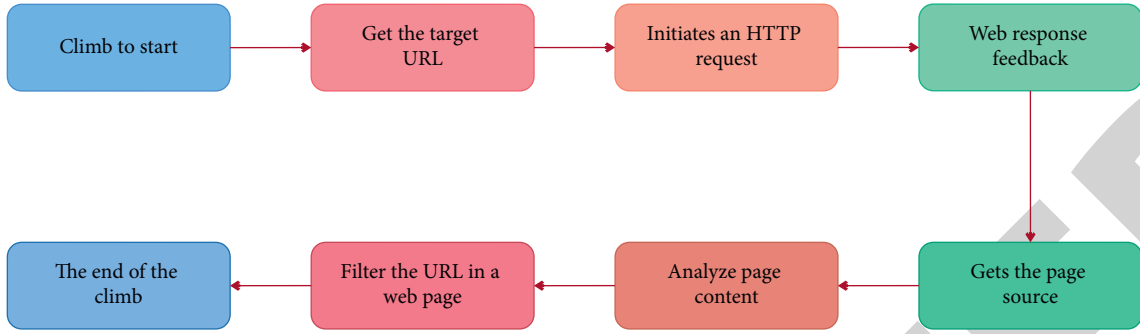


FIGURE 3: Python crawler flow chart.

TABLE 1: Super parameters of CASC model.

Parameter		Value
Input layer	Number of nodes	512
Convolution layer	Number of convolution kernels;	64
	Convolution kernel size	3 × 3
Hidden layer	Number of nodes	64
Output layer	Number of nodes	512

TABLE 2: Precision, recall, and  $F_1$  of the three algorithms.

Algorithm	CASC algorithm	K-means algorithm	Spectral clustering algorithm
$P$	85%	72%	81%
$R$	60%	43%	53%
$F_1$	0.7	0.53	0.64

Precision refers to the number of samples correctly clustered after the clustering algorithm. The larger the value, the better the effect. Recall rate refers to the proportion of the number of samples correctly clustered in the actual clustering results in the number of samples that should be correctly clustered.  $F_1$  is a composite index of accuracy and recall. The larger the value, the better the effect [26, 27]. The three formulas are as follows:

$$\begin{aligned}
 P &= \frac{m}{m+n}, \\
 R &= \frac{p}{q}, \\
 F_1 &= \frac{2PR}{P+R},
 \end{aligned}
 \tag{7}$$

where  $P$  refers to the accuracy, its variable  $m$  refers to the number of documents correctly clustered,  $n$  refers to the number of documents not correctly clustered,  $R$  refers to the recall rate,  $p$  refers to the number of documents clustered into the same category, and  $q$  refers to the number of documents in this category.

The experimental results are shown in Table 2.

It can be seen from Table 2 that under the same number of clusters, the CASC algorithm is better than the other two algorithms in terms of precision, recall, and F1 index. It can be seen that the convolutional self-encoder does not degrade

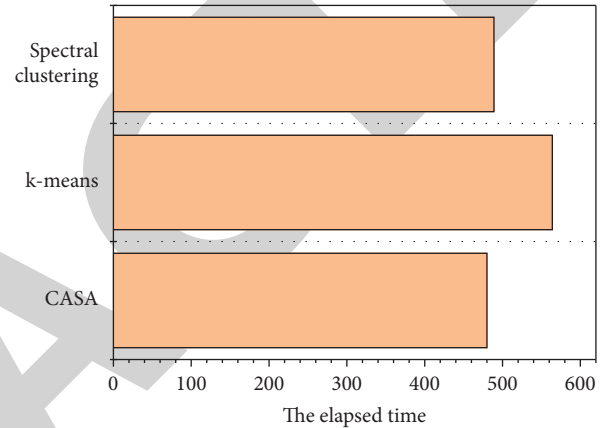


FIGURE 4: Running time of three algorithms on food safety news documents.

the clustering performance due to the loss of too much information when embedding high-dimensional vectors into low-dimensional space. The spectral clustering algorithm based on graph theory is better than the k-means algorithm in dealing with high-dimensional sample vectors to a certain extent. As can be seen from Figure 4, the CASC algorithm has the shortest running time, which is due to the fact that the convolutional self-encoder greatly reduces the dimension of document vector and reduces the time complexity of the algorithm. The other two algorithms are time-consuming due to the large number of operations of high-dimensional vectors.

## 5. Conclusion

This paper compiles the Python crawler program through major food safety websites and news reports of static web pages and dynamic web pages and stores them in local files after preprocessing. While transforming the semistructured text data obtained by the crawler into structured data, it brings the problem of high dimension of data. This paper presents a CASC algorithm based on artificial intelligence, which uses convolutional neural network and self-encoder to extract the features of structured data. This method can preserve the internal structure of the original data to the greatest extent, explore its potential association, reduce the original high-dimensional vector to the dimension of

document vector for clustering, shorten the running time of the algorithm, and reduce the time complexity of the algorithm without reducing the clustering accuracy.

At the same time, as an unsupervised model, the clustering algorithm plays an important role in document automatic processing. How to further combine artificial intelligence technology to improve the performance of clustering algorithm is worthy of our further research.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declared that they have no conflicts of interest.

## References

- [1] J. C.-W. Lin, P. Fournier-Viger, V. S. Tseng, and P. S. Yu, "Ieee access special section editorial: utility-pattern mining: theoretical analytics and applications," *IEEE Access*, vol. 9, pp. 16604–16607, 2021.
- [2] F. T. Giuntini, K. Moraes, M. T. Cazzolato, L. Kirchner, and J. Ueyama, "Tracing the emotional roadmap of depressive users on social media through sequential pattern mining," *IEEE Access*, vol. 99, p. 1, 2021.
- [3] R. Franceschini, A. Rosi, F. Catani, and N. Casagli, "Exploring a landslide inventory created by automated web data mining: the case of Italy," *Landslides*, vol. 19, no. 4, pp. 841–853, 2022.
- [4] Y. Li, R. K. Shyamasundar, and X. Wang, "Special issue on computational intelligence for social media data mining and knowledge discovery," *Computational Intelligence*, vol. 37, no. 2, pp. 658–659, 2021.
- [5] X. Q. Wang and F. J. Meng, "Unstructured road detection based on improved region growing with PCA-SVM rule[J]," *Journal of Computer Applications*, 2017.
- [6] D. Xie, "Research on multi-scale image edge detection based on wavelet transform [J]," *The public science and technology*, vol. 17, no. 11, pp. 3–5, 2017.
- [7] N. Mukherjee, S. Chavan, M. Colgan et al., "Fault-tolerant real-time analytics with distributed oracle database in-memory," in *Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp. 1298–1309, IEEE, Helsinki, Finland, 16–20 May 2016.
- [8] W. S. Gan, J. Chun, P. Fournier-Viger, H.-C. Chao, T.-P. Hong, and H. Fujita, "A survey of incremental high-utility itemset mining[J]," *Wiley Interdisciplinary Review Data Mining & Knowledge Discovery*, vol. 8, no. 2, 2018.
- [9] J. Benesty and I. Cohen, "Canonical correlation analysis in speech enhancement," *Springerbriefs in Electrical & Computer Engineering*, 2018.
- [10] G. Dai and H. Zhang, "Cross-media retrieval algorithm based on semantic correlation and topological relation [J]," *Computer Applications*, vol. 38, no. 9, pp. 2529–2534, 2018.
- [11] M. H. Amin, E. Mohamed, and A. Elragal, "Corporate disclosure via social media: a data science approach," *Online Information Review*, vol. 44, no. 1, pp. 278–298, 2020.
- [12] A. Ng and B. Mp, "Cognitive systems, artificial neural networks and differential equations: social media data," *Procedia Computer Science*, vol. 186, pp. 677–684, 2021.
- [13] B. Xu, Y. Yang, B. Zhang, Y. Yan, and Z. Yi, "Bionic design and experimental study for the space flexible webs capture system," *IEEE Access*, vol. 8, no. 99, pp. 45411–45420, 2020.
- [14] M. Li and J. Zhang, "Integrating kano model, ahp, and qfd methods for new product development based on text mining, intuitionistic fuzzy sets, and customers satisfaction," *Mathematical Problems in Engineering*, vol. 2021, no. 5, pp. 1–17, Article ID 2349716, 2021.
- [15] A. Sr and B. Cn, "Keyword weight optimization using gradient strategies in event focused web crawling," *Pattern Recognition Letters*, vol. 142, pp. 3–10, 2021.
- [16] M. R. Murugudu and L. Reddy, "Efficiently harvesting deep web interfaces based on adaptive learning using two-phase data crawler framework," *Soft Computing*, vol. 1, pp. 1–11, 2021.
- [17] H. Ali, G. Mubeen, T. S. Ali, Z. Tehseen, and A. Waqas, "High efficiency video coding (hevc)-based surgical telementoring system using shallow convolutional neural network," *Journal of Digital Imaging*, vol. 32, no. 6, pp. 1027–1043, 2021.
- [18] W. Xingjian and D. Shupeng, "Fraudulent phone call recognition method based on convolutional neural network," *High Technology Letters*, vol. 26, no. 4, pp. 21–25, 2020.
- [19] B. Chen, "Image processing operations identification via convolutional neural network," *Science China Information Sciences*, vol. 63, no. 3, pp. 1–3, 2020.
- [20] Z. Lv, H. Ding, L. Wang, and Q. Zou, "A convolutional neural network using dinucleotide one-hot encoder for identifying dna n6-methyladenine sites in the rice genome," *Neurocomputing*, vol. 422, no. 4, pp. 214–221, 2021.
- [21] Z. Huang, J. Zhu, J. Lei, X. Li, and F. Tian, "Tool wear monitoring with vibration signals based on short-time fourier transform and deep convolutional neural network in milling," *Mathematical Problems in Engineering*, vol. 2021, no. 6, pp. 1–14, Article ID 9976939, 2021.
- [22] S. Sunita Agarwala, A. Kumar, A. K. Dhara, S. B. Thakur, A. Sadhu, and D. Nandi, "Special convolutional neural network for identification and positioning of interstitial lung disease patterns in computed tomography images," *Pattern Recognition and Image Analysis*, vol. 31, no. 4, pp. 730–738, 2021.
- [23] C. Peng, H. Jiang, and L. Qu, "Deep convolutional neural network for passive rfid tag localization via joint rssi and pdoa fingerprint features," *IEEE Access*, vol. 9, pp. 15441–15451, 2021.
- [24] K. Jin, H. Seo, H. Ha et al., "Recent advances in heterogeneous Mn-based electrocatalysts toward biological photosynthetic Mn4Ca cluster," *Catalysis Today*, vol. 353, pp. 232–241, 2020.
- [25] R. Huang, "Framework for a smart adult education environment," *World Transactions on Engineering and Technology Education*, vol. 13, no. 4, pp. 637–641, 2015.
- [26] M. Bradha, N. Balakrishnan, S. Suvi et al., "Experimental, computational analysis of Butein and Lanceoletin for natural dye-sensitized solar cells and stabilizing efficiency by IoT," *Environment, Development and Sustainability*, vol. 1, 2021.
- [27] M. Fan and A. Sharma, "Design and implementation of construction cost prediction model based on svm and lssvm in industries 4.0," *International Journal of Intelligent Computing and Cybernetics*, vol. 14, no. 2, 2021.