IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 12, December 2022

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.118

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 12, December 2022||

| DOI:10.15680/IJIRSET.2022.1112007 |

An Automatic File Classification System Using Sentence BERT

Nakyung Im[1], Dain Kang[1], Sang-Young Cho[2]

U.G. Student, Division of Computer Engineering, Hankuk University of Foreign Studies, Yongin, Korea[1]

Professor, Division of Computer Engineering, Hankuk University of Foreign Studies, Yongin, Korea[2]

ABSTRACT:Process automation using software agents has been attempted to improve work efficiency in various fields and applications. One of them is an automatic file classification system that automatically organizes many files generated through computer work. Existing programs performed static clean-up based on the file's metadata. In this paper, our method classifies the files based on the semantics of file names using natural language processing. File classification based on semantics yields more natural results than classification based on static metadata of files. Our program classifies files using SBERT and K-means Clustering. The UI was constructed using PyQt and can operate regardless of the operating systems.

KEYWORDS:RPA, NLP, Sentence BERT, Word embedding, K-means clustering, File classification

I. INTRODUCTION

RPA(Robotic Process Automation) is a software technology that automatically processes regular and repetitive tasks performed by humans according to a predetermined process. Through software agents that replace human roles, work productivity is increased and accurate work processing is possible[1]. With the recent increase in non-face-to-face work due to the 4th industrial revolution and pandemic, the introduction of task automation using RPA is increasing in various industries[2]. In [3], market research on replacing repetitive tasks with RPA according to the post-COVID-19 situation and introduction of RPA through POC(proof of concept) projects presented cases for improving work efficiency and studied efficiency measures.

Work automation using software agents has been attempted to improve work efficiency in various fields and applications. One of them is an automatic file classification system that automatically organizes many files generated through computer work. [4] dealt with a system that organizes files based on file type, modification date, and number of accesses. Since [4] classifies files based on clear information that the file itself has, it operates with rules for file classification. Therefore, mechanical classification of files is possible, but classification based on the semantics of titles and contents of files is not possible. In fact, it provides a file organization automation system in a file storage environment such as NAS(Network Attached Storage)[5] or cloud drive[6]. These commercially available automatic file classification systems also operate based on static metadata of files, so they do not classify files semantically.

In this paper, unlike existing file organization systems, we deal with file organization automation based on file semantics. In particular, a program that groups unorganized files in a file explorer into folders with similar files based on the file name will be described. Existing file explorers have only a function to sort files and folders based on name, date, etc., and no function to group similar files based on their names or contents. Efficient work activities become possible if files can be automatically grouped and organized for files that are created in large quantities during the work process. If the function of classifying files based on the semantics of the file is linked with the file explorer, it will be a useful tool for PC users.

This paper aims to apply an automatic file classification system based on semantics to the problem of file classification and sorting, one of the chronic problems of personal computer users. To do this, SBERT(Sentence BERT)[7] and the K-Mean algorithm[8] using the Elbow method are used to determine groups with similar meanings for file names. SBERT is used to make a file name into a vector, and K-Mean is used to determine a group of similar files. The operation of the program to be developed is as follows: When a user of a personal computer selects a folder to organize, sentence embedding and clustering are applied based on the file name under the folder, and similar files are grouped

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 12, December 2022||

| DOI:10.15680/IJIRSET.2022.1112007 |

into folders and automatically organized. The developed program has a user-friendly screen composition and classifies files into meaningful groups based on file names.

II. DESIGN OF AN AUTOMATIC FILE CLASSIFICATION SYSTEM

The overall system design is divided into a file classification function based on the file name and a program design using this function. The file classification function uses natural language processing technology, an AI technique. The program is developed using the PyQt[9] library so that it can be used in various operating systems.

2.1. System design

The core of 'An Automatic File Classification System' is to analyze file names using natural language processing and to group and organize similar files. The file name can be written in several languages such as Korean and English. To classify files based on the meaning of the filename, it is necessary to convert the filename into one language. Also, filenames contain special characters and file extensions that are irrelevant to meaning. In order to perform natural language processing for file classification, a file name refinement process is required. Therefore, before the natural language processing of the file name, a pre-processing process of removing meaningless parts from the file name is performed, and a process of translating the file name into English, which is one language, is performed.

In order to perform natural language processing with refined file names, word or sentence embedding must be performed. The embedding operation converts the file name into a computer representation, mapping the name to the expression vector space. As such an embedding technique, there is mBERT(multilingual BERT), a multilingual parallel technique[10]. mBERT converges in a similar vector space if the parts of speech are the same even if the meaning of the words is different. This property is unsuitable for our system, which wants to classify filenames based on meaning. On the other hand, SBERT has the property of gathering words based on meaning. In our system, word embedding is performed using SBERT. File name clustering is a step of classifying file names with word embedding into groups having the same meaning. It is classified using a clustering technique[11]. Clustering has been widely studied in data mining, and this paper uses the K-means method[8].

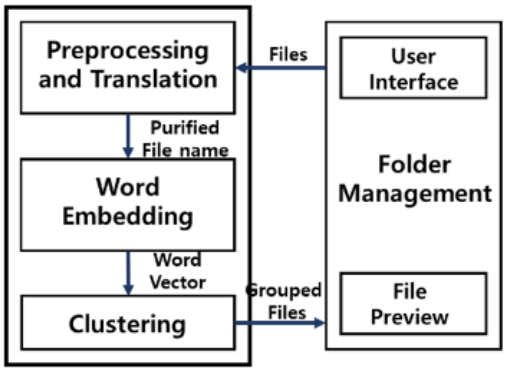


Figure1. System structure

Figure 1 shows the configuration diagram of the automatic file classification system. The entire program has folder management capabilities through a user interface. In folder management, when a folder to be classified is designated, the names of files in the folder are transferred to the file name classification function. In addition, it receives the classification result of the file name classification function and previews it to the user so that the classified result can be confirmed. As previously stated, the file name classification function is largely composed of three parts. These include preprocessing to remove special characters and extensions from file names, translation functions for language unification, sentence embedding for natural language processing, and grouping of files. The file classification results are presented and reviewed to the user in the file management function.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 12, December 2022||

| DOI:10.15680/IJIRSET.2022.1112007 |

2.2. GUI design

The system is implemented in Python so that it can operate regardless of the operating system. PyQt is used to implement the graphical user interface. The system can run on Linux, Windows, and MacOS PC environments, and can run on any system that supports Python and PyQt. Figure 2 shows the flow of the program.

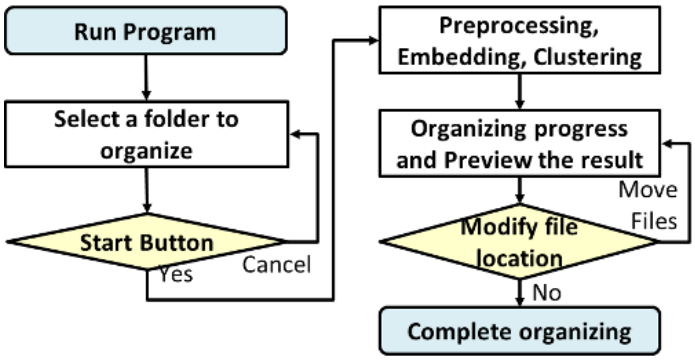
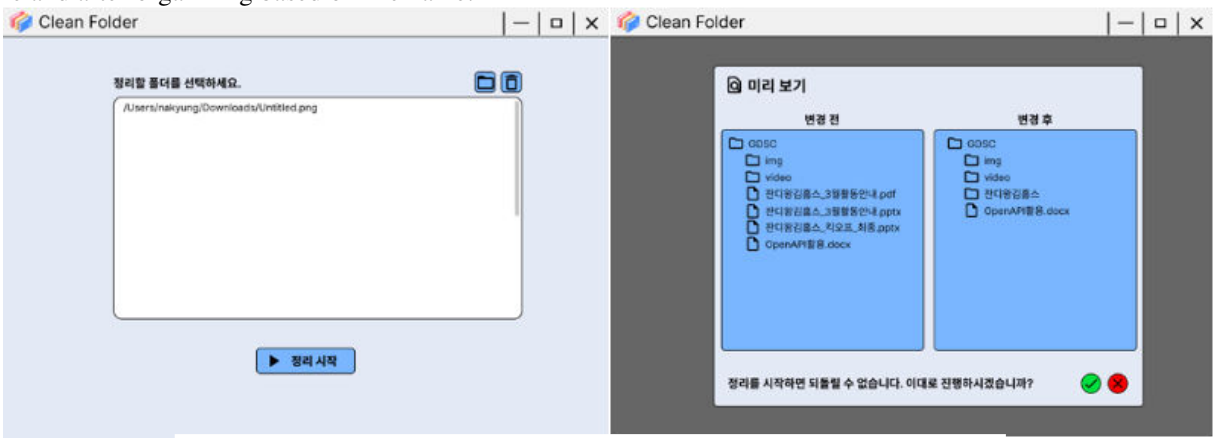


Figure 2. GUI workflow

The operation of the program is straightforward. Run the program, select the folder containing the files to be cleaned, and place the files in the list of candidates to be cleaned. When you click the Start Organizing button, files are organized through data preprocessing and translation, sentence embedding, and clustering steps. When the files are sorted, the automatically sorted result is shown to the user, and the user can modify the file location. Modifying the location of the files is optional. If you do, the cleanup will proceed again. If you click OK without modifying the location, the folder cleanup will be completed.

In order to implement the file classification system as a program, the user interface must be created along with the workflow function. The program allows users to use it conveniently with a simple and intuitive screen composition. Figure 3(a) and Figure 3(b) are the configuration of the operation screen in the design stage. Figure 3(a) is the Main Page screen and Figure 3(b) is the Preview Page screen. In Figure 3(a), you can select a folder to organize with the folder icon placed at the top right, and you can delete the selected folder from the text box with the trash icon. Click the Start Cleanup button to begin cleaning up files. Figure 3(b) is a preview screen that visually shows before and after organizing based on file name.



(a) Main page in design (b) Preview page in design

Figure 3. GUI design of the system

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 12, December 2022||

| DOI:10.15680/IJIRSET.2022.1112007 |

III. IMPLEMENTATION OF THE SYSTEM

3.1. Implementation of automatic file classification function

File name clusteringconsists of file name preprocessing and translation, sentence embedding, clustering. Figure 4 shows the entire clustering process implemented.

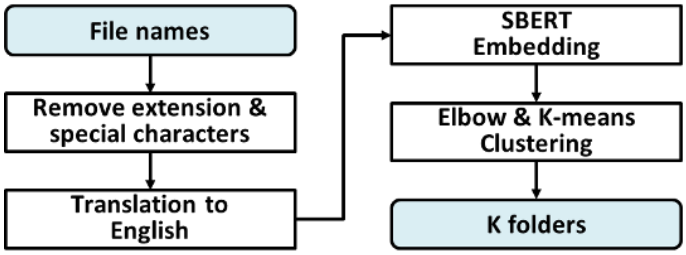


Figure 4. File classification process

Firstly, in the data pre-processing and translation steps, characters that become obstacles in determining the similarity in a later step are removed. The characters include special characters and extension names. After that, it is assumed that Korean, English, numbers, etc. are mixed in the file names, and all file names are translated into English. We use the Kakao I Translation API to translate all file names into English[12]. Table 1 is a data storage structure for file name processing. Data necessary for the clustering process is saved in a csv file.

Table 1. Data structure for clustering process

Field	Contents
origin	Original file name
no-ext	Preprocessed file name
dir	File path
trans	Translated file name
emb	Embedding vector
cluster	Clustering result
distance_from_centeroid	Data distance from cluster center

The original data of the file name is stored in the ‘origin’ column, the path of the original file is stored in the ‘dir’ column, the preprocessed data is stored in the ‘no-ext’ column, and the translated data is stored in the ‘trans’ column. The translated filename is used as input to the sentence embedding step. The embedding step was implemented using the Sentence BERT library[13]. Libraries used are nltk(Natural Language Toolkit) for natural language processing, spaCy, and data analysis library pandas. In spaCy's function, we use the SBERT model to convert the translated names into vectors. Embedding vectors are stored in the ‘emb’ column in the csv file.

Finally, in the clustering step, similar sentences are clustered using the K-means library. When using the elbow function to determine the number of clusters and specifying a minimum of 2 to a maximum of 5 clusters, the K value with the highest cohesion of data is selected in the K-means model. Clustering is performed using the KMeansClusterer model using the K value. Clustering data is stored in the ‘cluster’ column, and the distance of data from the center of gravity of each cluster is stored in the ‘distance_from_centeroid’ column. The files are grouped according to the original file name stored in the 'origin' column, the path of the 'dir' column, and the results of clustering of the 'cluster' column, and placed in each of the K new folders.

3.2. Implementation of system GUI

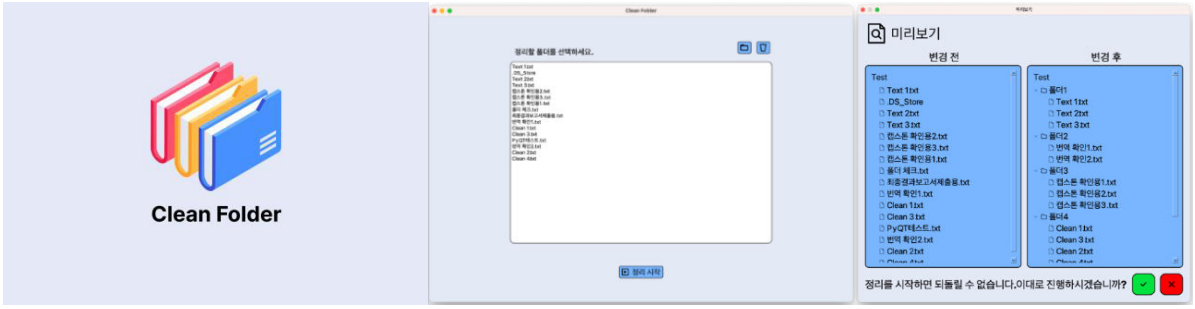
PyQt is a framework that allows you to create GUI programs by linking Python code to Qt layouts. In this paper, PyQt5 was used. Figure 5 shows the screen when the program starts up and after starting and selecting the folder containing the files to be cleaned up. Figure 5(a) is the screen shown to the user as a Splash screen during the preparation time of the model at the start. The loading time takes about 1 to 3 seconds, and the loading progress is placed at the bottom of the screen using the progress bar. After the program finishes loading, select a folder to organize, and a screen showing the files in the folder appears. After the files are organized through clustering, the

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 12, December 2022||

| DOI:10.15680/IJIRSET.2022.1112007 |

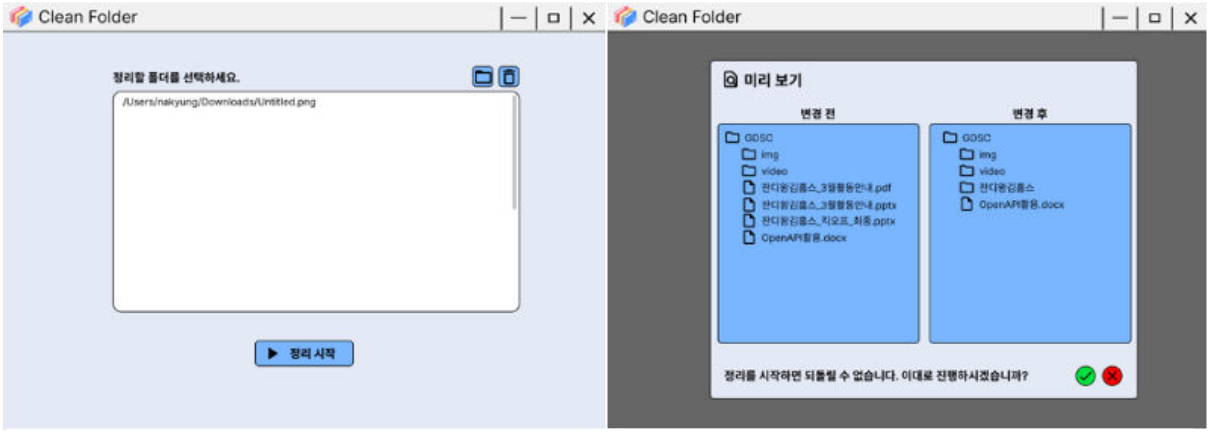
system shows a preview view (Figure 5(c)) and finally requests confirmation from the user. After user confirmation, save the result.



(a) Starting screen (b)The folder-selected screen (c) The implemented preview screen

Figure 5. Main screens of the system

Figure 6 shows the screen before (a) and after (b) executing the file organization automation program in the file explorer of the user's PC. With this system, users can easily organize complex files inside folders with just a few clicks.



(a) File explorer view of before classification (b) File explorer view of after classification

Figure 6. Classification results of the system

origin	no-ext	dir	trans	cluster	distance_from_centroid
Text 1.txt	Text 1	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Text 1	0	1.6732741656338035
Text 2.txt	Text 2	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Text 2	0	1.455129108983634
Text 3.txt	Text 3	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Text 3	0	1.3178460535830998
캡스톤 확인용2.txt	캡스톤 확인용2	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Capstone identification 2.	2	0.4074579164950796
캡스톤 확인용3.txt	캡스톤 확인용3	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Capstone Verification 3	0	3.4390191418888496
캡스톤 확인용1.txt	캡스톤 확인용1	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Capstone identification 1.	2	0.4085772850386253
폴더 체크.txt	폴더 체크	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	folder check	4	2.772257535356947
최종결과보고서제출용.txt	최종결과보고서제출용	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	final result report submission	4	2.905172855599654
번역 확인1.txt	번역 확인1	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Translation check 1.	1	0.3521927835577672
Clean 1.txt	Clean 1	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Clean one.	3	2.21692574822317
Clean 3.txt	Clean 3	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Clean 3.	3	0.9773896545462444
PyQT테스트.txt	PyQT테스트	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	pyqt test	4	2.75712369261202
번역 확인2.txt	번역 확인2	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Translation check 2.	1	0.3531603565771061
Clean 2.txt	Clean 2	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Clean 2.	3	0.8497647550107026
Clean 4.txt	Clean 4	/Users/nakyung/Library/CloudStorage/OneDrive-개인/Test	Clean 4.	3	1.3213872851179849

Figure 7. The result window of saving the data in the data.csv file after completing clustering

Figure 7 is the result window saved in the data.csv file according to the data storage structure defined in Table 1. In order to cluster file names, data is stored in stages from original data to preprocessing, translation, and clustering

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 12, December 2022||

| DOI:10.15680/IJIRSET.2022.1112007 |

results. After embedding, check the data in the 'cluster' column where the clustering results are stored. Based on the translated data, it can be seen that similar sentences are clustered. That is, it is clear that similar files are grouped and organized in the same folder.

IV. CONCLUSION

An automatic file classification system using natural language processing was developed. Existing file organization systems categorize files based on their static metadata. Because our system categorizes files based on the meaning of the filenames, more natural classification is possible. It has a simple UI so that users can use it easily. The program uses PyQt so that it can operate regardless of the operating system.

Our system improves work efficiency and provides convenience by automating simple and repetitive tasks that require users to manually check and classify various files. It is a programming that can be used by most companies doing office work and personal computer users. However, selecting a folder in the program requires user intervention and decision-making. An RPA system that returns to its initial settings 24 hours a day without user intervention in the RPA system is called Unattended RPA.As in the case of Unattended RPA in the future, it can be developed to automate file organization in a fixed process without a separate user command after initially designating the folder to be organized.

ACKNOWLEDGEMENT

This research was supported by the MIST(Ministry of Science, ICT), Korea, under the National Program for Excellence in SW), supervised by the IITP(Institute of Information & communications Technology Planning & Evaluation) in 2022"(2019-0-01816). This work was supported by Hankuk University of Foreign Studies Research Fund.

REFERENCES

[1] Peter Hofmann, Caroline Samp, and Nils Urbach3, Robotic process automation, Electron Markets 30, pp. 99–106, 2020. https://doi.org/10.1007/s12525-019-00365-8
[2] Young Geun Hyun, JooYeoun Lee, Trends Analysis and Future Direction of Business Process Automation, RPA(Robotic Process Automation) in the Times of Convergence, Journal of digital convergence, Vol. 16, No. 11, pp.313-327, 2018.
[3] Jaewook Choi, MyungsikYoo, A study on the introduction of RPA (Robotic Process Automation) and improvement of efficiency, Proceedings of the Korean Institute of Communication Sciences Conference, pp.378-380, 2021.
[4] Hyungjoon Jeon, GayoungGim, Jaerin Kim, Deukkyu Lee, Hyunsoo Joe, KeunhyukYeom, Machine learning-based cloud storage services for supporting automated file classification, Proceedings of the Korean Information Science Society Conference, pp. 1620-1622, 2016.
[5] Qfiling: https://www.qnap.com/ko-kr/software/qfiling
[6] Dropbox: https://blog.dropbox.com/topics/product/new-features-to-keep-files-organized
[7] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Hong Kong, pp. 3982–3992, Nov. 2019. [Online].Available: https://aclanthology.org/D19-1410
[8] D. Marutho, S. Hendra Handaka, E. Wijaya and Muljono, "The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News," 2018 International Seminar on Application for Technology of Information and Communication, pp. 533-538, 2018, doi: 10.1109/ISEMANTIC.2018.8549751.
[9] PyQt: https://en.wikipedia.org/wiki/PyQt
[10] Ethan A. Chi, John Hewitt, and Christopher D. Manning, Finding Universal Grammatical Relations in Multilingual BERT. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5564–5577, 2020.
[11] Clustering: https://en.wikipedia.org/wiki/Cluster_analysis
[12] Kakao I Translation API: https://www.kakaoicloud.com/
[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, 2018.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

Scan to save the contact details