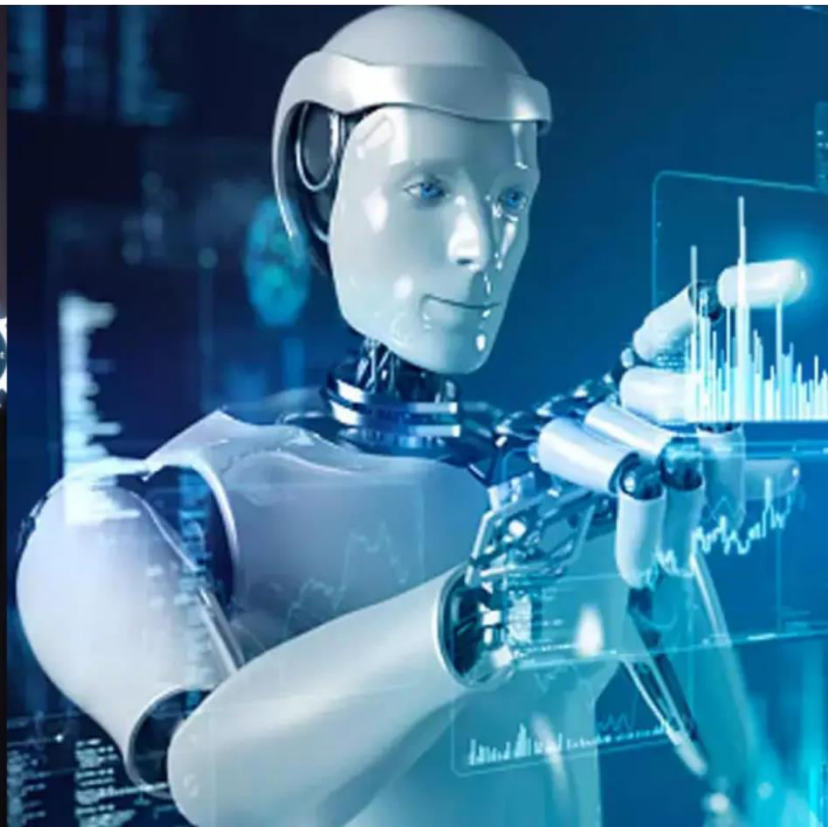
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

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

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 5, May 2026

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

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505174|

Enhanced Android Malware Detect Models Based on Explainable Generative Adversarial Networks

S. Priyavarshini, R.Gayathri, MCA, M. Phil., NET

Department of MCA, Vivekanandha Institute of Information and Management Studies, Tiruchengode, Namakal, Tamil Nadu, India

Assistant Professor, Department of MCA, Vivekanandha Institute of Information and Management Studies, Tiruchengode, Namakal, Tamil Nadu, India

ABSTRACT: The widespread adoption of smartphones over the past decade, mobile applications have become a primary target for malicious attacks, usually in the form of malware. Recent studies have leveraged artificial intelligence (AI) techniques for malware detection and classification. However, applying such approaches, particularly deep learning (DL) techniques, to mobile malware detection poses significant challenges. These challenges arise from the difficulty of collecting large quantities of mobile malware samples and the inherent class imbalance in the collected datasets. To tackle these issues and enhance the performance of machine learning (ML) and DL detection models, we propose novel detection models based on a generative adversarial network (GAN). Furthermore, our approach not only employs a conditional tabular GAN (CTGAN) for data augmentation to explore the impact of augmentation but also identifies the optimal multiplication factor for achieving the best results. The evaluation results demonstrate that the proposed data augmentation approach significantly improves the performance of mobile malware detection models, especially those based on DL.

KEYWORDS: android applications; malware; machine learning; deep learning; cybersecurity

I. INTRODUCTION

The number of Android applications has been increasing rapidly owing to the significantly increased use of Android-based mobile devices. Currently, Google Play has over three million applications. Unfortunately, there is also a significant amount of malicious code within these applications. Android-based systems and devices have become primary targets for attackers seeking access to other people's money by stealing their personal information and monitoring their data. Additionally, because of the open-source features associated with Android-based applications, attackers can easily upload programs running malicious code to Google Play, including programs running well-known malicious code categorized as Trojan horses, adware, file infectors, riskware, backdoors, spyware, and ransomware. Because of the current proliferation and the increasing complexity of malware, it is imperative to develop efficient malware detection mechanisms to address this crucial issue[1].

Security personnel use various technological approaches to detect and classify Android malware. Such approaches can be categorized as static, dynamic, and mixed forms of analyses, which are performed with the aid of machine learning. Over recent years, most studies on static feature extraction focused on extracting features, such as bytecode, sound, images, log records, code execution paths, control flow graphs, and data flow graphs. Such studies also focused on using machine learning algorithms, such as decision trees, logistic regression, Naive Bayes, support vector machines (SVM), and random forests (RF), or deep neural network (DNN)-based models, such as convolutional neural networks, generative adversarial networks, and recurrent neural networks, to detect malicious code[3]. On the other hand, current research on source code machine translation is extensive and accurate. To the best of our knowledge, there are no studies on malicious code detection using machine translation to enhance the information contained in the source code of malicious software. In addition to the use of classical machine learning algorithms, research on quantum machine learning has resulted in significant breakthroughs. For example, it has been shown theoretically that exponential acceleration can be achieved using a quantum support vector machine (QSVM)[5].

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

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505174|

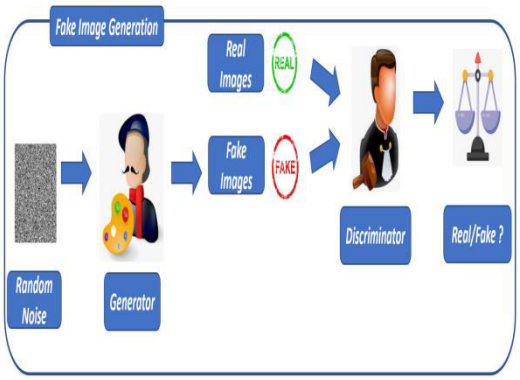


Fig 1: Deep Convolutional Generative Adversarial Networks in Image-Based Android Malware

Currently, little research has been conducted on the use of decompiled source code as a machine learning feature, especially in detecting malicious code. Cen et al. were the first to use decompiled source code, API call sequences, and permissions as features[7]. They applied a probabilistic discriminant model based on regularized logistic regression to detect Android malware. Akram et al. proposed DroidMD, which is an extensible tool based on self-improvement that detects malicious applications in the market at the source code level. DroidMD analyzes only the application code, thereby detecting malicious code, with an accuracy of 95.5%. The studies mentioned above have achieved initial results in the field of Android malicious code detection, but the features used in these previous studies are limited to common features, and the experiments' results are not satisfactory. However, there is currently no study on transferring comment information from Android source code to the malicious code domain as program features for semantic enhancement through pretrained models[11]. This is because it is currently impossible to obtain comment information from malicious code. Therefore, we attempt to establish novel features at the source code level to better describe program functions and extract semantic information. On the other hand, the malware comments obtained through transfer learning may not be of good quality, and as a result, they cannot be used to express program semantics. Therefore, we propose novel optimization methods using adversarial generation and filtering[15].

Therefore, in this study, we propose Android-SEM, an Android source code semantic enhancement model based on transfer learning. Our proposed model can be used to detect Android malware effectively. Based on the principles of natural language processing (NLP), our proposed model (Android-SEM) first extracts the semantics of Android source code, after which it generates enhanced semantic information of the source code. Android-SEM mainly comprises two parts: a transfer learning-based semantic enhancement model and a quantum classifier-based QSVM. The first part pretrains the dataset corresponding to more than two million source codes through a transformer-based deep learning model, which we then improve using a generative adversarial network. The generative network attempts to create comments samples that will be incorrectly identified as human-made, and the pretraining model learns to correctly identify such samples[14]. Our proposed model then decompiles the samples contained in the CICMalDroid2020 dataset to obtain the source code and normalize it, after which it performs semantic enhancement on a pretrained transformer to obtain the comments vector. The semantic enhancement vectors are then fused with the vectors of the decompiled source code. The feature fusion vector must pass through the newly proposed regression model-based filter[12]. The second part of our model performs quantum feature mapping on the feature vectors using QSVM to classify malicious code. Through experiments, we demonstrate that our proposed model can detect malware at an accuracy level of 99.55%. We demonstrate the effectiveness and future applications of the semantic enhancement, hybrid classical, and quantum models proposed in this study through ablation experiments[10].

II. RELATED WORK

This section offers an overview of previous research related to intrusion detection systems, Android malware detection, and standard datasets of Android malicious attacks. Furthermore, it provides an overview of the machine learning and deep learning approaches applied to the design of cybersecurity systems[2]. The regular improvement of sophisticated Android malware families, e.g., Chamois malware, has made the task of detecting malicious attacks daunting. To tackle this, researchers developed machine learning techniques that improved the available systems. Recently, many studies

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

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505174|

have applied machine learning models for Android botnet detection, such as linear regression, KNN, SVM, and decision trees (DT) algorithms. Some of these recent studies used deep learning algorithms, although they do not provide a thorough understanding of their effectiveness. Therefore, the current study compares with deep learning models to examine their effectiveness in Android botnet detection with the use of the available installation support center of expertise (ISCX) botnet dataset[11].

Kadir et al. used deep learning models to analyze Android botnet attacks in an attempt to understand the latter's hidden features. The system was evaluated using the ISCX Android botnet dataset, which contained 1929 samples. Anwar et al. proposed an Android botnet detection approach based on static functions. The features of permissions, MD5 signatures, and broadcast receivers were combined and processed with machine learning algorithms. The input data collected from the ISCX dataset were 1400 from different botnet applications, with the system achieving an accuracy of 95.1% in distinguishing Android botnet attacks[10].

Several machine learning algorithms were proposed to classify normal and abnormal botnet attacks. In one study, the results indicated that the random forest approach had 0.972% precision and 0.96% recall. In machine learning approaches were proposed for detecting Android botnets. The ISCX dataset consisted of 1635 benign and 1635 attacks. The random forest tree model achieved 97%. In another study, the DT, Naive Bayes, and random forest machine learning algorithms were used to detect Android attacks. The information gain method was used to select the significant features. The random forest algorithm achieved a 94.6% accuracy[11]. Karim et al. proposed the static analysis approach to explore the pattern of the features of Android botnet attacks. The features were compared with the intrusion application using the Drebin dataset. Artificial intelligence (AI) approaches using a knowledge-based system were used to secure Android mobiles against malicious attacks. Inspired by a meta-heuristic rule and based on fuzzy logic, intrusion detection and data mining systems were developed, while machine learning approaches were applied in the development of IDS applications. The design of IDS systems employed the artificial bee colony, particle swarm optimization, grey wolf optimization, and artificial fish swarm algorithms[15].

III. METHODS

We propose a transfer learning-based scoring device for generated comments to avoid generating bad comment vectors that do not represent program semantics effectively. Because similar source code and comment vectors must have similar semantics, the Euclidean distance between the two is small. Therefore, during the pretraining stage, we used the function vector obtained from the encoder and the Euclidean distance between the comments vector obtained from the decoder and the BLUE value calculated from the generated comments and its label value, both of which were trained using a linear regression model. After the model is saved, the bad comments are filtered by generating the Euclidean distance between the comments vector and the function vector obtained during the fine-tuning stage. Therefore, we could eliminate bad comments in the feature fusion stage without the need for annotated labels[14].

First, we obtained more than 16,000 APK files of five types from the CICMalDroid2020 dataset. After complete feature engineering and extraction, 14,873 samples comprising 1500 types of adware, 2204 types of banking malware, 4622 types of SMS malware, 2534 types of riskware, and 4013 types of benign software were obtained. We used these samples as the dataset for classifying Android malware. To balance the number of benign and malicious samples, we obtained 6847 types of benign software from Google Play. To further ensure the effectiveness of the experiment, we obtained software from multiple categories, including business, games, and education, to prevent the emergence of a single type of benign software. We balanced the numbers of benign and malicious software samples in preparation for the malware detection tasks.

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

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505174|

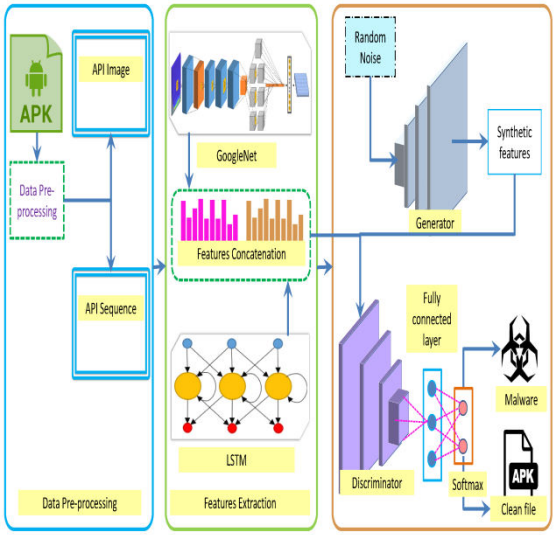


Fig 2: A Multifaceted Deep Generative Adversarial Networks Model for Mobile Malware

By contrast, we conducted fine-tuning training on the API sequences extracted from the malicious code dataset. We also used the concept of NLP to map the API sequences through the vocabulary, PyTorch.nn.embedding for word embedding, and the bidirectional gated recurrent unit (Bi-GRU) to learn the API semantics and context to obtain the vector representation of the API sequence. Finally, the feature fusion vector formed using the method, comments, and feature vector of the API sequence were spliced in preparation for malicious code detection and classification tasks. The tasks were performed using two methods: quantum feature mapping and MLP.

In contrast, this study does not rely on analyzing images. Instead, it features the first semantic enhancement model that uses source code and transfer learning to generate comments automatically. Large-scale data were used for pretraining through a transformer-based deep learning model, and comments were generated from the source code to enhance program semantics. After the vector generated by the encoder and decoder parts during the pretraining process was filtered, feature fusion was performed through bilinear pooling. Finally, the context relationship was obtained through Bi-GRU training using the API sequence. This study is the first to use a pretraining model to generate comments automatically, which were then used as semantic enhancement features after feature fusion. It is also the first to use a generative adversarial network to improve the quality of generated semantics and migration learning to significantly save training time and achieve significantly high levels of accuracy.

IV. RESULT ANALYSIS

To prove that QSVM was comparable to classical SVMs in detecting and classifying malicious Android code, or better in certain circumstances, the hyperparameters were varied, and the classification performance and decision boundary were observed to select the best data fitting method and make slight adjustments. The objective was for the SVM kernel to produce better classification results than those of the QSVM.

Subsequently, we constructed the classification accuracy and confusion matrices of different types of quantum cores in the QSVM. The function of the SVM kernel is to transform data into a more suitable modeling space. Different algorithms use different types of kernel functions. It is well known that the most commonly used functions are linear and RBF, which we used as our comparison methods. Specifically, we used the linear and RBF methods in the classical SVM, after which we compared the scores of different quantum kernel functions in the QSVM.

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

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505174|

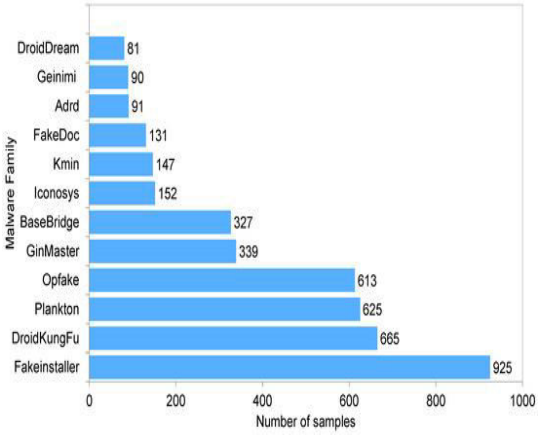


Fig 3: A Multifaceted Deep Generative Adversarial Networks Model for Mobile Malware

We removed the semantic enhancement module of the part that generates source code comments, and we retained only the pretrained encoder for position information, word embedding, and the multi-head attention mechanism in the encoder. Simultaneously, the semantic vector obtained using the API sequence through word embedding and Bi-GRU was retained. The two were vectorized and classified. The accuracy obtained in this experiment was approximately 2% lower than that obtained in the previous experiment. However, compared with the enhanced semantic module, the difference was 6%. This shows that the proposed semantic enhancement model can obtain more semantic information and improve classification accuracy. Finally, when we eliminated the entire pretraining module and used word embedding and a long short-term memory (LSTM) model for semantic extraction and feature classification, the accuracy dropped by approximately 5% compared with that of the former. This shows that the transformer-based pretraining model can obtain additional semantic information from a large Android program source code corpus in advance to complete the malicious code detection task, achieve a satisfactory classification result, and significantly reduce training time.

V. CONCLUSIONS

Based on the concept of transfer learning, in this study, we focused on obtaining comment information not initially present in Android malware through migration. We integrated conventional deep learning models with quantum machine learning and proposed Android-SEM, which is a transfer learning-based model for the semantic enhancement of Android malicious code. To improve the effectiveness of our proposed model in detecting malicious code, we optimized the performance of the comment vectors generated by the semantic enhancement module through a generative adversarial approach. We also proposed a comment filter based on a linear regression model that retains high-quality comment vectors and combines them with other features to detect and classify Android malicious code. Experiments on the CICmalDroid2020 dataset showed that our proposed method can detect malicious code in 13,694 benign and malicious samples with an accuracy level of 99.55%. Moreover, 14,873 samples of different types of malicious code in the dataset can be classified with 99.01% accuracy. We conducted ablation experiments for the proposed model, which illustrated the necessity of each module, demonstrated the effectiveness of our proposed method, and showed that our proposed method achieved enhanced accuracy in the detection and classification of malicious code. In addition, this study combined quantum and classical machine learning methods. Simulation experiments showed that quantum and classical machine learning models achieved exceptional accuracy levels in malicious code detection tasks, with QSVM resulting in a theoretical enhancement.

REFERENCES

1. Liu, Z.; Wang, R.; Japkowicz, N.; Tang, D.; Zhang, W.; Zhao, J. Research on unsupervised feature learning for Android malware detection based on Restricted Boltzmann Machines. Future Gener. Comput. Syst. 2021, 120, 91–108. [Google Scholar] [CrossRef]
2. Chen, D.; Wawrzynski, P.; Lv, Z. Cyber security in smart cities: A review of deep learning-based applications and case studies. Sustain. Cities Soc. 2021, 66, 102655. [Google Scholar] [CrossRef]

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

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505174|

3. Shang, Y. Consensus of Hybrid Multi-Agent Systems With Malicious Nodes. IEEE Trans. Circuits Syst. II Express Briefs 2020, 67, 685–689. [Google Scholar] [CrossRef]
4. Fragkos, G.; Minwalla, C.; Plusquellic, J.; Tsiropoulou, E.E. Artificially Intelligent Electronic Money. IEEE Consum. Electron. Mag. 2021, 10, 81–89. [Google Scholar] [CrossRef]
5. Wu, Q.; Zhu, X.; Liu, B. A Survey of Android Malware Static Detection Technology Based on Machine Learning. Mob. Inf. Syst. 2021, 2021, 8896013. [Google Scholar] [CrossRef]
6. Kouliaridis, V.; Kambourakis, G. A Comprehensive Survey on Machine Learning Techniques for Android Malware Detection. Information 2021, 12, 185. [Google Scholar] [CrossRef]
7. LeClair, A.; Jiang, S.; McMillan, C. A neural model for generating natural language summaries of program subroutines. In Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), Montreal, QC, Canada, 25–31 May 2019; pp. 795–806. [Google Scholar]
8. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. Phys. Rev. Lett. 2014, 113, 130503. [Google Scholar] [CrossRef]
9. Cen, L.; Gates, C.S.; Si, L.; Li, N. A probabilistic discriminative model for android malware detection with decompiled source code. IEEE Trans. Dependable Secur. Comput. 2014, 12, 400–412. [Google Scholar] [CrossRef]
10. Akram, J.; Mumtaz, M.; Jabeen, G.; Luo, P. DroidMD: An efficient and scalable android malware detection approach at source code level. Int. J. Inf. Comput. Secur. 2021, 15, 299–321. [Google Scholar] [CrossRef]
11. LeClair, A.; McMillan, C. Recommendations for datasets for source code summarization. arXiv 2019, arXiv:1904.02660. [Google Scholar]
12. Mahdavifar, S.; Kadir, A.F.A.; Fatemi, R.; Alhadidi, D.; Ghorbani, A.A. Dynamic Android Malware Category Classification using Semi-Supervised Deep Learning. In Proceedings of the 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, 17–22 August 2020; pp. 515–522. [Google Scholar]
13. Zhang, X.; Zhang, Y.; Zhong, M.; Ding, D.; Cao, Y.; Zhang, Y.; Zhang, M.; Yang, M. Enhancing state-of-the-art classifiers with API semantics to detect evolved android malware. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 9–13 November 2020; pp. 757–770. [Google Scholar]
14. Onwuzurike, L.; Mariconti, E.; Andriotis, P.; Cristofaro, E.D.; Ross, G.; Stringhini, G. Mamadroid: Detecting android malware by building markov chains of behavioral models (extended version). ACM Trans. Priv. Secur. 2019, 22, 1–34. [Google Scholar] [CrossRef] [Green Version]
15. Xu, K.; Li, Y.; Deng, R.; Chen, K.; Xu, J. DroidEvolver: Self-evolving Android malware detection system. In Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P), Stockholm, Sweden, 17–19 June 2019; pp. 47–62. [Google Scholar]
16. Arp, D.; Spreitzenbarth, M.; Hubner, M.; Gascon, H.; Rieck, K.; Siemens, C. Drebin: Effective and explainable detection of android malware in your pocket. NDSS 2014, 14, 23–26. [Google Scholar]

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details