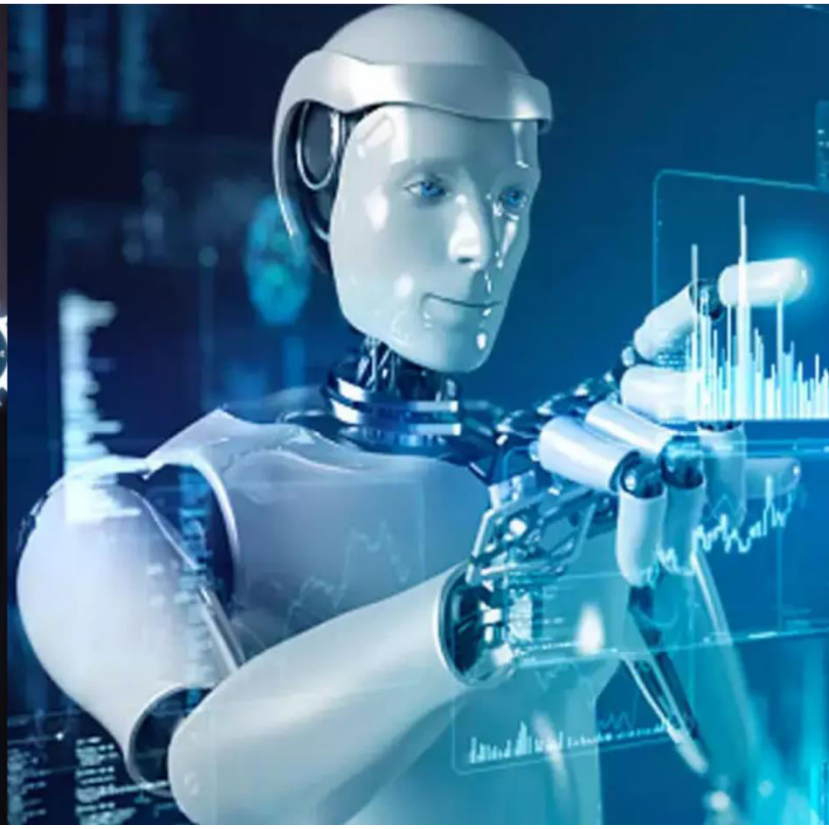
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

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: 8.699

Volume 14, Issue 9, September 2025

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

Impact Factor 8.699

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

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409052|

Web Based Vulnerability Scanning Tool

Kruthik CH, Manish N

U.G. Student, Department of Information Science and Engineering, S J C Institute of Technology, Chickaballapura, Karnataka, India

U.G. Student, Department of Information Science and Engineering, S J C Institute of Technology, Chickaballapura, Karnataka, India

ABSTRACT: Web applications are frequent targets of cyber-attacks due to their widespread use and sensitive data handling. This paper presents a web-based vulnerability scanner capable of detecting common threats such as CSRF, SQL Injection, XSS, and Broken Access Control. Built with Django for the backend and JavaScript with AJAX for an interactive interface, the system allows users to scan URLs and receive real-time results with descriptions and mitigation strategies. A proof-of-concept prototype demonstrates the tool's effectiveness in identifying vulnerabilities while ensuring secure data handling. This solution integrates modern web technologies and security practices to enhance web application security.

KEYWORDS: Web Security, Vulnerability Scanning, Cybersecurity, Django Framework, Web Application Security, SQL Injection, Cross-Site Scripting (XSS), Broken Access Control, CSRF Protection

I. INTRODUCTION

Web applications have become critical components of modern digital infrastructure, supporting essential services in sectors such as finance, healthcare, education, and e-commerce. Despite their importance, these applications are increasingly targeted by cyber-attacks due to vulnerabilities such as Cross-Site Request Forgery (CSRF), SQL Injection, Cross-Site Scripting (XSS), and Broken Access Control. These security flaws pose serious risks, including unauthorized data access, service disruption, and financial losses, thereby highlighting the urgent need for advanced protective mechanisms.

Traditional security approaches often fail to provide adequate protection against evolving threats, as they are either limited in scope or require deep technical expertise for deployment and monitoring. This leaves organizations exposed to potential breaches and undermines user trust. In this context, automated vulnerability scanning tools have emerged as essential solutions that can proactively detect, monitor, and report security flaws in web applications.

This paper introduces a web-based vulnerability scanner developed using Django as the backend framework and JavaScript with AJAX for an interactive and responsive interface. The system enables real-time URL scanning, identifies potential vulnerabilities, and provides detailed reports along with suggested mitigation strategies. To ensure secure operations, the scanner incorporates safe input handling and data integrity mechanisms, making it both effective and reliable.

The goal of the proposed system is to provide a practical, accessible, and efficient security solution that empowers developers and organizations to safeguard their web applications. By integrating modern web technologies with advanced scanning techniques, the system strengthens application resilience, reduces the risk of cyber-attacks, and contributes to building a secure and trustworthy digital environment.

II. PROBLEM STATEMENT

Web applications, which handle sensitive data and deliver essential services, are increasingly vulnerable to cyber-attacks. Developers and organizations often face challenges in identifying and mitigating threats such as Cross-Site equest Forgery (CSRF), SQL Injection, Cross-Site Scripting (XSS), and Broken Access Control. Existing security measures and manual testing methods are time-consuming, require advanced technical expertise, and often fail to

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

Impact Factor 8.699

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

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409052|

provide continuous protection. This creates a critical need for an automated solution that can accurately detect vulnerabilities in real time and support timely remediation.

One of the major pain points in current security practices is the lack of user-friendly and automated tools that can be operated without deep cybersecurity expertise. Many scanners provide complex outputs that are difficult for non-experts to interpret, while others lack real-time reporting, delaying the detection and mitigation process. Without streamlined and responsive tools, organizations remain exposed to risks such as unauthorized data access, financial loss, and reputational damage. A solution is needed that not only detects vulnerabilities effectively but also simplifies interpretation through clear reporting and actionable mitigation strategies.

The cybersecurity landscape continues to evolve with increasingly sophisticated attacks, but many web applications—especially in small and medium enterprises—still lack access to affordable, comprehensive, and scalable vulnerability scanning tools. While commercial solutions exist, they are often costly, resource-intensive, and unsuitable for diverse deployment environments. There is a pressing need for a unified, web-based vulnerability scanner that integrates automated detection, real-time feedback, secure data handling, and accessibility features, ultimately empowering organizations to strengthen their web application security and build resilient digital infrastructures.

III. LITERATURE REVIEW

1. Author : D. Azhawanth & G. Sujatha
a. Title: A Novel Automated Method to Detect XSS Vulnerability in Webpages
b. Outcome: The study proposes an automated method for detecting Cross-Site Scripting (XSS) vulnerabilities in webpages, improving accuracy and efficiency in identifying injection-based threats.
c. Disadvantage: The approach is limited to XSS vulnerabilities and does not comprehensively address other major web threats such as SQL Injection or CSRF.

2. Author : Giuseppe Beltrano, Claudia Greco, Michele Ianni & Giancarlo Fortino
a. Title: Deep Learning-Based Detection of CSRF Vulnerabilities in Web Applications
b. Outcome: This research applies deep learning models to effectively detect CSRF vulnerabilities, demonstrating how AI techniques can enhance web security automation.
c. Disadvantage: The system's dependency on training datasets limits adaptability to new or evolving attack patterns without continuous model updates.

3. Author : Nilima D. Bobade, Vinit A. Sinha & Swati S. Sherekar
a. Title: A Diligent Survey of SQL Injection Attacks, Detection and Evaluation of Mitigation Techniques
b. Outcome: The paper provides a comprehensive survey of SQL Injection attacks and evaluates multiple detection and prevention strategies, highlighting their strengths and weaknesses.
c. Disadvantage: While insightful, the work is limited to theoretical survey and does not provide a practical implementation framework.

4. Author : Abheenesh Kejiou & Girish Bekaroo
a. Title: A Review and Comparative Analysis of Vulnerability Scanning Tools for Wireless LANs
b. Outcome: This study reviews existing vulnerability scanning tools for wireless LAN environments, offering insights into their effectiveness and limitations.
c. Disadvantage: The focus on wireless LANs narrows applicability, leaving broader web-based vulnerabilities outside its scope.

5. Author : Fahad A. Alshaya, Sultan S. Alqahtani & Yasser A. Alsamel
a. Title: Machine Learning Driven Cybersecurity Tool for Vulnerability Classification
b. Outcome: he research introduces a machine learning–driven system that classifies vulnerabilities based on the CWE (Common Weakness Enumeration) framework, enhancing reporting and categorization.
c. Disadvantage: The classification system relies heavily on labeled datasets and may struggle with zero-day or novel vulnerabilities.

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

Impact Factor 8.699

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

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409052|

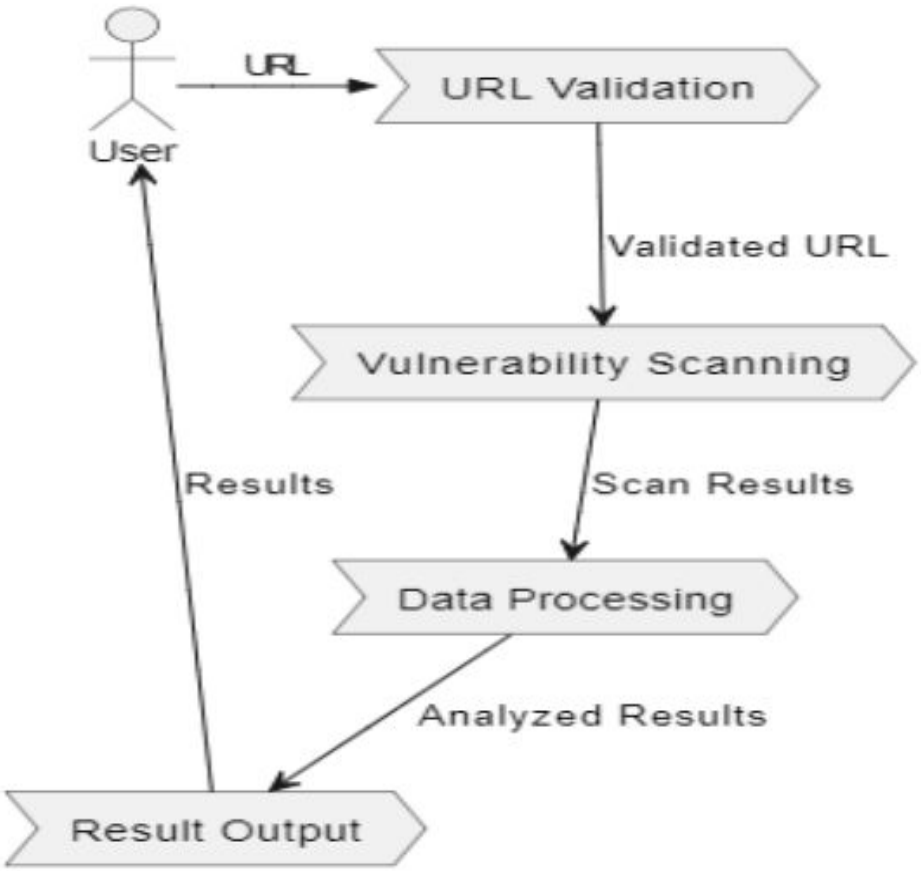
6. Author : Veneta Yosifova, Antoniya Tasheva & Roumen Trifonov

a. Title: Predicting Vulnerability Type in Common Vulnerabilities and Exposures (CVE) Database with Machine Learning Classifiers

b. Outcome: The study applies machine learning classifiers to predict and categorize vulnerability types in the CVE database, showcasing the role of AI in vulnerability management.

c. Disadvantage: The reliance on existing CVE datasets reduces effectiveness against unknown vulnerabilities and limits real-time adaptability.

IV. DESIGN AND IMPLEMENTATION



The data flow diagram (DFD) for the **Web Application Vulnerability Scanning Tool** illustrates how input data (URLs) flows through the system to identify and report security issues. The process starts when the user submits a **URL** to the system. This URL first undergoes **validation**, ensuring that it is in the correct format and safe to process. Once validated, the system moves to the **vulnerability scanning** stage, where the submitted URL is tested for potential threats such as XSS, SQL Injection, CSRF, and other common security weaknesses. The scanning process generates **scan results**, which are then sent to the **data processing** module. This module analyzes the results, categorizes detected vulnerabilities, and prioritizes them based on severity. Finally, the **result output** stage delivers the analyzed findings back to the user in a clear, structured format.

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

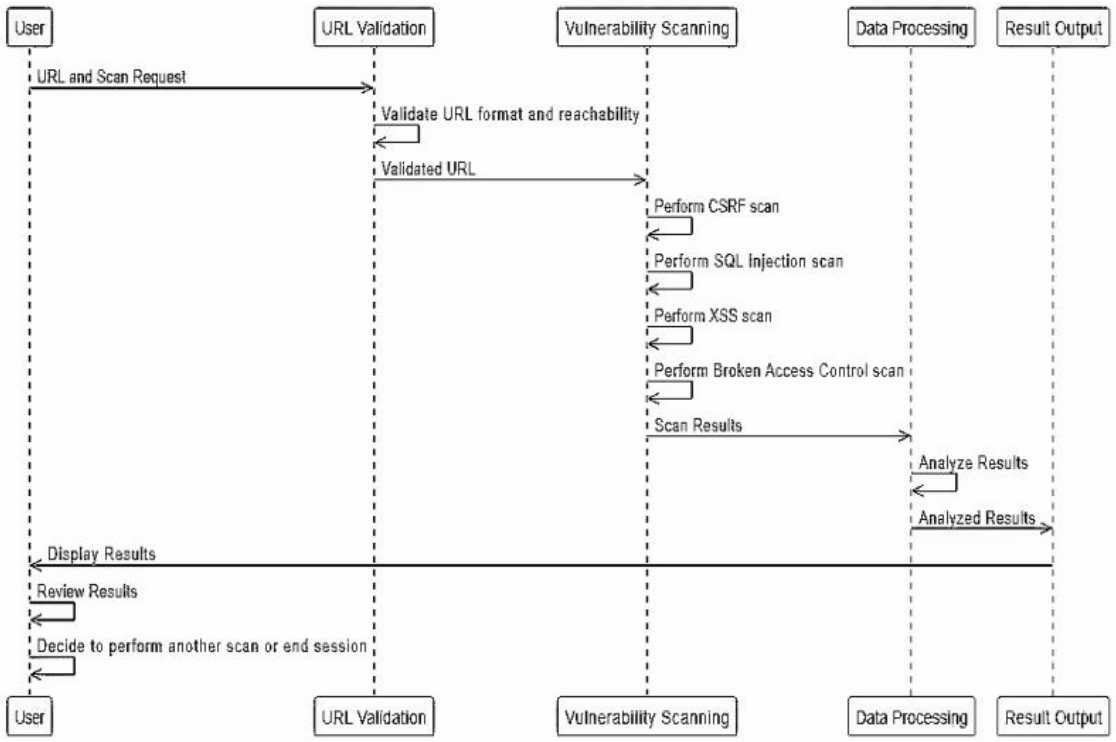
Impact Factor 8.699

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

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409052|

Sequence Diagram



The sequence diagram illustrates the flow of interactions between the user and different modules of the web-based vulnerability scanning system. The process begins when the user submits a URL and scan request to the system. The request is first handled by the URL Validation module, which checks whether the submitted URL is in the correct format and reachable. Once validation is successful, the validated URL is passed on to the next stage.

The Vulnerability Scanning module then performs a series of security checks on the validated URL. These include scanning for Cross-Site Request Forgery (CSRF), SQL Injection, Cross-Site Scripting (XSS), and Broken Access Control. Each vulnerability test is executed sequentially, and the results are collected after the scanning process.

Next, the Data Processing module analyzes the scan results. This involves categorizing vulnerabilities by severity, filtering out false positives, and preparing structured results that are easy for users to interpret. The processed results are then forwarded to the Result Output module, where they are displayed back to the user.

The user can then review the results, understand the vulnerabilities detected, and decide whether to perform another scan or end the session. This sequence ensures a systematic flow from input validation to vulnerability detection, analysis, and user feedback, while maintaining real-time interaction and accuracy throughout the process.

V. SYSTEM DESIGN

The **User Interaction Module** is responsible for providing a simple and user-friendly interface where users can enter the URL of the web application they want to scan. Once the URL is submitted, this module ensures smooth interaction by forwarding the request to the backend and later displaying the scan results to the user. It acts as the bridge between the user and the system, allowing even non-technical users to initiate scans and understand the results with ease.

The **URL Input and Validation Module** ensures that the URL entered by the user is well-formed, valid, and reachable before scanning begins.

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

Impact Factor 8.699

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

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409052|

The **Vulnerability Detection Module** forms the core of the system, responsible for identifying common web application vulnerabilities such as Cross-Site Request Forgery (CSRF), SQL Injection, Cross-Site Scripting (XSS), and Broken Access Control. It uses automated testing scripts and payload injection techniques to detect flaws in the target application. This module ensures that potential threats are flagged for further analysis and reporting.

The **Data Processing and Analysis Module** manages the results generated by the vulnerability detection process. It categorizes vulnerabilities based on severity levels such as Critical, High, Medium, and Low. In addition, it provides clear interpretations of the results by highlighting affected URLs, parameters, and recommended mitigation strategies. This ensures that the output is not just raw technical data but actionable insights for the user.

The **Real-Time Monitoring Module** enhances the system's usability by showing live progress of the scan. It provides indicators such as percentage completion, vulnerabilities currently being tested, and intermediate results if available. Once the scan is completed, this module updates the final results dynamically on the user interface, ensuring users receive immediate feedback on the security status of their web application.

The **Security Module** ensures that all user inputs and outputs are handled securely throughout the scanning process. It follows best practices by avoiding the storage of sensitive user data and ensuring data integrity. Secure communication is maintained between the user and the system to prevent unauthorized access, thereby building trust and maintaining confidentiality.

IMPLEMENTATION

CSRF Detection: The user interface of the application includes a mechanism to check for Cross-Site Request Forgery (CSRF) vulnerabilities. The application retrieves all forms present on the web page and examines each form for the presence of a hidden input field named 'csrf'. If a form does not contain a hidden input field with a name that includes 'csrf', the page is marked as vulnerable to CSRF attacks.

SQL injection Detection: The application includes a feature to detect SQL injection vulnerabilities by leveraging a list of common SQL injection payloads. For each payload in the list, the application sends a GET request to the target URL, appending the payload as a query parameter. The response is then analyzed for common SQL error messages. If any error message indicative of an SQL injection flaw is detected, the page is marked as vulnerable to SQL injection.

XSS Detection Detection: The application includes a feature to detect Cross-Site Scripting (XSS) vulnerabilities by leveraging a list of common XSS payloads. For each payload in the list, the application sends a GET request to the target URL, appending the payload as a query parameter. The response is then analyzed to check if the payload appears in the response content. If the payload is found in the response, the page is marked as vulnerable to XSS. This approach helps in identifying potential security issues related to XSS attacks.

Broken Access Control Detection: The application includes a feature to detect Broken Access Control vulnerabilities by attempting to access a common admin URL. By appending '/admin' to the base URL, the application sends a GET request to this potentially sensitive endpoint. If the HTTP response status code is 200, indicating that access is allowed without proper authorization, the page is marked as vulnerable to Broken Access Control. This approach helps in identifying potential security issues related to unauthorized access to administrative functions.

Scanning: The application includes a feature to start a scanning process when the user clicks the "Start Scanning" button. This process retrieves the selected vulnerability type from the dropdown menu, starts a progress indicator, and performs the scanning in a new thread to keep the user interface responsive.

VI. CONCLUSION

This project successfully developed a web-based vulnerability scanning tool designed to enhance the security of modern web applications by detecting common vulnerabilities such as Cross-Site Request Forgery (CSRF), SQL Injection, Cross-Site Scripting (XSS), and Broken Access Control. By automating the scanning process and providing real-time results, the system reduces reliance on manual methods, enabling quicker identification and mitigation of potential threats.

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

Impact Factor 8.699

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

Volume 14, Issue 9, September 2025

|DOI: 10.15680/IJIRSET.2025.1409052|

Built using Django for the backend and JavaScript with AJAX for real-time interactivity, the tool ensures a secure, lightweight, and responsive user experience. The system effectively validates user inputs, processes scan data, and delivers analyzed results without storing sensitive information, thereby maintaining confidentiality and integrity throughout the scanning process.

Through systematic analysis, design, implementation, and rigorous testing, the project achieved its primary objectives of building an efficient, accurate, and user-friendly vulnerability detection system. Its modular architecture ensures scalability, allowing the tool to handle multiple scans while maintaining performance and reliability.

Looking ahead, potential improvements include integrating machine learning models to predict and classify emerging vulnerabilities, expanding the scope of detection to cover more advanced attack vectors, and offering integration with existing security tools for enterprise-level use. Enhancements in the reporting interface, along with customizable scanning options, could further increase usability and adoption.

Overall, this project provides a strong foundation for a secure, real-time, and accessible vulnerability scanning solution that contributes to building safer web applications and fostering trust in the digital ecosystem.

REFERENCES

[1]. D Azhawanth, G. Sujatha(2022) “A novel automated method to detect XSS vulnerability in webpages” Published in International Conference on Computer Communication and Informatics (ICCCI).
[2]. Giuseppe Beltrano, Claudia Greco, Michele Ianni, Giancarlo Fortino(2023) “Deep LearningBased Detection of CSRF Vulnerabilities in Web Applications” Published in 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).
[3]. Nilima D. Bobade, Vinit A. Sinha, Swati S. Sherekar(2024) “A diligent survey of SQL injection attacks, detection and evaluation of mitigation techniques” Published in IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS).
[4]. Abheenesh Kejiou, Girish Bekaroo(2022) “A Review and Comparative Analysis of Vulnerability Scanning Tools for Wireless LANs” Published in International Conference on Next Generation Computing Applications (NextComp).
[5]. Fahad A. Alshaya, Sultan S. Alqahtani, Yasser A. Alsamel(2023) “A CWE-Based Vulnerability Report Tagger Machine Learning Driven Cybersecurity Tool for Vulnerability Classification” Published in IEEE/ACM 1st International Workshop on Software Vulnerability (SVM).
[6]. Veneta Yosifova, Antoniya Tasheva, Roumen Trifonov(2021) “Predicting Vulnerability Type in Common Vulnerabilities and Exposures (CVE) Database with Machine Learning Classifiers” Published in12th National Conference with International Participation (ELECTRONICA).

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