A Comparative Analysis of PHP Frameworks (Laravel , Codeignitor and Symfony)

Mr. Ankul Yadav[1*]

[1*]Dept. of CSE Khwaja Moinuddin Chishti Language University, Lucknow, India

e-mail: ankulyadav9001@gmail.com,

Corresponding Author: ankulyadav9001@gmail.com Tel.: +91-9336879087

Abstract— This paper is a comparative analysis of PHP programming framework. A framework defined as a structure that supports the development of dynamic websites, web applications, and services. Framework code and design are often reusable to assist resource management, customization and API related tasks. This study discusses current practices to help a developer understand PHP frameworks adoption for web application development. The systematic approach, score criteria evaluation, and PHP framework technical factors are studied to understand the features suitability of PHP Frameworks. The pure PHP programming language provides slightly better performance results than the CodeIgniter and Laravel frameworks. Hence, the use of pure PHP or frameworks can be determined from the level of needs of the application development to be created. Some popular PHP frameworks are Laravel, Symfony, CakePHP, Yii, and CodeIgniter. Performance is the most aspect to be concerned because of the market's demands entailing them to do so. The next concern that developer would likely need to decide is whether to go with Pure PHP or to go with PHP Framework.

Keywords— PHP, Programming framework, Symfony, Laravel, CodeIgniter, performance, comparison, model, scalability, Customization , API , Web Applications ,

I. INTRODUCTION

PHP was initially developed by Rasmus Lerdorf in 1994 to watch over his online resume and related personal information in which PHP initially named as" Personal Home Page". However, two programmers Zeev Suraski and Andi Gutmans rebuilt, updated, and released the PHP core in 1997 and changed the acronym PHP to" Hypertext Processor". Through time PHP has evolved and PHP has been used as a language for the World Wide Web (WWW) or so-called Internet in which developers find PHP is a language that easy to learn, community-friendly, freely available as open-source software, and easy to deploy. The current PHP environment requires developers to create interface components of the system, link to the database, and user authentication. The usage of framework could overcome the problems during development life cycle environment by reuse of code which could save times and costs to design, developing codes and tests .[1]

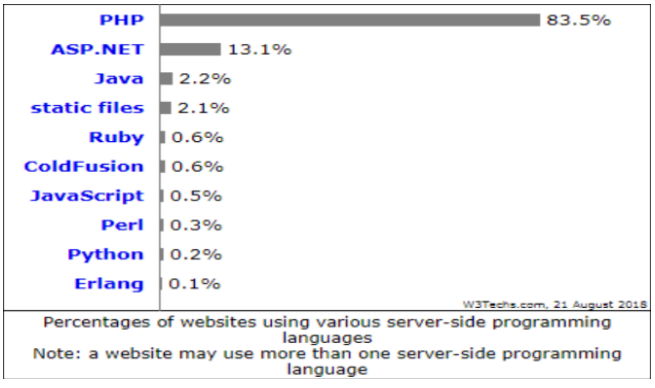


Figure 1: Server-side programming languages for websites development

Even though PHP is not as famous as any other programming language especially when creating web services, in fact, PHP is the most used server-side programming language by May 2018. PHP is used by 83.4% of all websiteswhich the server uses PHP as its programming language. Based on the percentage that we have here, it means that PHP is stillworth consideringwhen we want to start developing a project specifically web services projects.

The PHP programming language is one of the languages supported by mostservers today, so developing a web-based application using PHP will become easier for programmers. Server customization is one of the many jobs that must be completed by application developers, and by using PHP, programmers do not need to make too many adjustments to the server. By default, PHP is configured on most servers today, unlike other programming languages which require programmers to install or adjusthere and there on the server side.

PHP or core PHP is a basic programming language which could also be used to develop dynamic web pages. The core PHP works without libraries or built-in functionalities and usually developers have to make the code script using their skills and logic. On the other hand, PHP framework is a framework that consist a collection of organized source codes in a specified architecture to support the development of dynamic websites applications and services . There are a few comparison studies that have been conducted by researchers to identify which PHP frameworks would provide developers with better features regarding performance, development process, and code maintenance. These researchers have conducted several testing such as load testing and stress testing to measure the performance efficiency on the specified PHP frameworks. The discussion involved the suitability choices on various issues related to

PHP frameworks for specified web application development. Usually,the comparison made based on the researchers' interest to study in depth on certain PHP frameworks.Some Famous PHP Frameworks are given below.

1. Laravel
2. CodeIgniter
3. Symfony
4. CakePHP
5. Yii
6. Zend
7. Phalcon
8. Zend Framework

Website technology is experiencing a fairly rapid development. The current website is not only a medium of information that is read or tends to be static, but has developed into a more dynamic and interactive medium. A website is a web page that is interconnected with one another where the web data is located on the same server and contains various collections of information that can be provided to individuals, groups, and organizations.The use of the website as one of the most popular means of communication among the public, has indeed undergone a very diverse transformation, with a very low cost factor, as well as ease of access and efficiency because it can be accessed within 24 hours. Making a website is indeed undergoing many very diverse developments, it is proven by the many choices of frameworks that can be used to build a website. Framework is a basic programming language that has been developed and made easy to use so that a website can be completed in a relatively short time. Frameworks are also often interpreted as a collection of scripts (especially classes and functions) that can assist developers/programmers in dealing with various programming problems, such as connecting to databases, calling variables, managing files, and so on. used by calling it on the program, of course how to call it depends on the framework used, then the programmer does not need to recreate these functions from scratch, while the method that is often used by the framework is Code Igniter or it can also be called Model-View Controller or it can be abbreviated as MVC.

II. PHP FRAMEWORK CONCEPT

The concept of the PHP framework is related to (OOP) and Model View Controller (MVC). MVC is a design pattern that applies the concept of software development, originally designed to provide multiple views of the same data virtually for modern interactive applications. The MVC pattern allows data interaction and methods in multiple classes and offers possible solutions to problems that could arise in the application development procedure. MVC pattern has three separate application components as follows: [5]

- **Model** —represent data structure relationship and dependencies which provides an interface to manipulate all classes corresponding to the logical object of the application.
- **View** —represent screen presentation on different devices in which the application could have multiple views of the data.
- **Controller** —represent as an information collector or input for the user and transfers the information to the model.

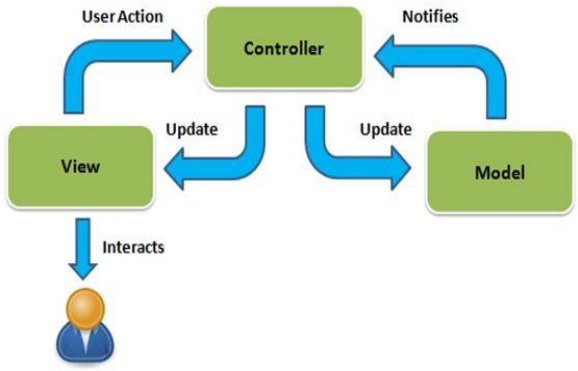


Figure 2. MVC Concept

III. DIRECTORY STRUCTURE

Laravel:

1. **app**: This directory contains the core application code, including controllers, models, and views.
2. **bootstrap**: Contains the application's bootstrap files, including the app.php file that initializes the Laravel application.
3. **config**: Contains configuration files for various aspects of the application, such as database connections and services.
4. **database**: Contains database migrations, seeds, and factories for managing database schema and seeding initial data.
5. **public**: This is the web server's document root. It contains the entry point (index.php) and assets like CSS, JavaScript, and image files.
6. **resources**: Contains views, language files, and assets like CSS and JavaScript that need to be compiled or processed before being used in the application.
7. **routes**: Contains route definitions for the application.
8. **storage**: Contains application logs, file uploads, cached views, and other files generated by the framework.
9. **tests**: Contains automated tests for the application.
10. **vendor**: Contains Composer dependencies.
11. **.env**: Environment-specific configuration file.

CodeIgniter:

1. **app**: In CodeIgniter, the application directory typically holds the controllers, models, and views, similar to Laravel's app directory.

2. **system**: Contains the core framework files and libraries.
3. **writable**: Directory for storing temporary and cache files generated by the application.
4. **public**: This directory serves as the document root for the web server. It contains the index.php file, which is the entry point for the application, along with assets and other publicly accessible files.
5. **.env**: Environment-specific configuration file, similar to Laravel.

Symfony:

1. **app**: This directory contains configuration files and other application-specific settings.
    - **config**: Configuration files for different environments (dev, prod, test).
    - **Resource**s: Additional configuration resources such as routing, translations, and templates.
    - **cache**: Cache files generated by the framework.
    - **logs**: Log files generated by the application.
2. **bin**: Contains executable files, mainly the Symfony Console binary used for running console commands.
3. **src**: This directory holds the source code of the application.
    - **AppBundle**: An example directory for holding application-specific code. The actual directory name may vary depending on the project.
4. **Tests**:Directory for storing automated tests for the application
5. **var**: Directory for storing variable data generated by the application.
    - **Cache**: Cache files generated by the application
    - **logs**: Log files generated by the application.
    - **sessions**: Session files.
    - **test**: Temporary files used during testing.
6. **vendor**: Contains dependencies installed via Composer.
7. **web**: This directory serves as the document root for the web server.
8. **assets**: Static assets like CSS, JavaScript, and images.
9. **app.php**: The entry point for the application.
10. **favicon.ico**: The favicon icon.
11. .**env**: Environment-specific configuration file.
12. **composer.json**: Configuration file for Composer dependencies.
13. **composer.lock**: Lock file generated by Composer to lock dependencies to specific versions.

IV. LITERATURE REVIEW

Laravel vs. CodeIgniter vs. Symfony – A Detailed Comparison

Now, let's go more in-depth and compare all these PHP frameworks on the basis of key factors that you should consider while determining which one to pick for your projec

Have a look at common uses cases of Laravel, CodeIgniter and Symfony as PHP framework for web development and some of the widely known apps built with these frameworks.

- ***Laravel***

  Laravel is often praised for its elegant syntax and developer-friendly features. It is a PHP web application framework that follows the Model-View-Controller (MVC) architectural pattern. Here are some key characteristics of Laravel:
  - **Eloquent ORM**: Laravel's Eloquent ORM simplifies database interactions, making it easy to work with databases using expressive syntax.
  - **Blade Templating**: Blade, Laravel's templating engine, offers a clean and efficient way to create dynamic templates.
  - **Artisan Console**: Laravel includes a command-line tool called Artisan, which simplifies common tasks like migrations, seeding, and more.
  - **Lively Community**: Laravel has a vibrant community that continually produces packages and extensions, enhancing the framework's capabilities.

- ***CodeIgniter***

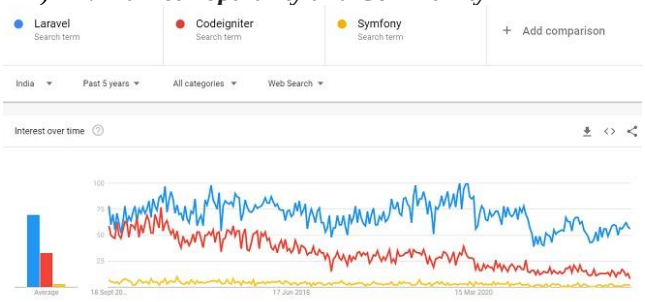
  CodeIgniter is known for its simplicity and lightweight nature. It's designed to be easy to learn and use, making it a great choice for developers new to PHP frameworks. Here are some key characteristics of CodeIgniter:
  - **Small Footprint**: CodeIgniter has a small footprint, making it suitable for projects where resource efficiency is a priority.
  - **Clear Documentation**: It boasts clear and comprehensive documentation, making it easy for developers to get started.
  - **Simplicity**: CodeIgniter promotes simplicity and allows you to write code quickly without extensive configurations.
  - **Community Support**: While the community is smaller than Laravel or Symfony, CodeIgniter still has an active user base.

❖ ***Symfony***

Symfony is known for its flexibility and ability to adapt to various project requirements. It is a set of reusable PHP components and a full-stack web application framework. Here are some key characteristics of Symfony:

- **Components**: Symfony's components are used not only in Symfony projects but also in other PHP applications. They provide a solid foundation for building web applications.
- **Twig Templating**: Symfony uses Twig as its templating engine, offering a secure and flexible way to create templates.
- **Doctrine ORM**: Symfony integrates with Doctrine, a powerful ORM, for managing database interactions.
- **Extensibility**: Symfony's modular architecture allows you to use only the components you need, making it highly customizable.

1) 2. Market Popularity and Community



As per Google Trends, the interest over time for Laravel has remained high over other PHP frameworks for the last five years.

But before coming to any conclusion, let's have a look at how vast is the community around each PHP framework. After all, a well-connected and strong community of users helps troubleshoot or ease the implementation of the technology.

a) *Laravel*

Laravel has extensive community support with more than 45,000 active users and 66.4k GitHub stars. You can find the Laravel community in several places
– Laracasts,Laravel.io, Github – to name a few.

b) *CodeIgniter*

CodeIgniter has a decent community with an 18.2k star rating and more than 1.8k users on Github. Other places where you can find community around this technology are Codeigniter forums and Stackoverflow.

c) *Symfony*

Like other PHP frameworks, Symfony also has great community support with a 25.8k star rating and more than 69k users on Github. Other places where here you can find community around Symfony include SymfonyConnect, Slack and Reddit.

1. Architecture

When selecting a backend framework, it's important to look at what type of architecture does framework support as it affects the overall performance and scalability of the web application. so, let's have a peek!

a) *Laravel*

Laravel is based on the Model-View-Controller (MVC) pattern that makes it easy for developers to work together due to a tight separation between logic and presentation layers. As a result, the development of large applications using Laravel becomes easy.

b) *CodeIgniter*

Unlike other PHP frameworks out there, CodeIgniter doesn't force developers to implement the MVC model but encourages them to fast-track and simplify the development of complex web applications.

c) *Symfony*

Like other PHP frameworks, Symfony also follows Model-View-Controller (MVC) pattern that separates the business logic from the presentation layer.

V. MOST POPULAR

In PHP Framework there are different framework are popular in but we check the three php framework (Laravel , Codeignitor , And Symfony) . In the Google Trend the Most search framework is Laravel in World wide and India as well.

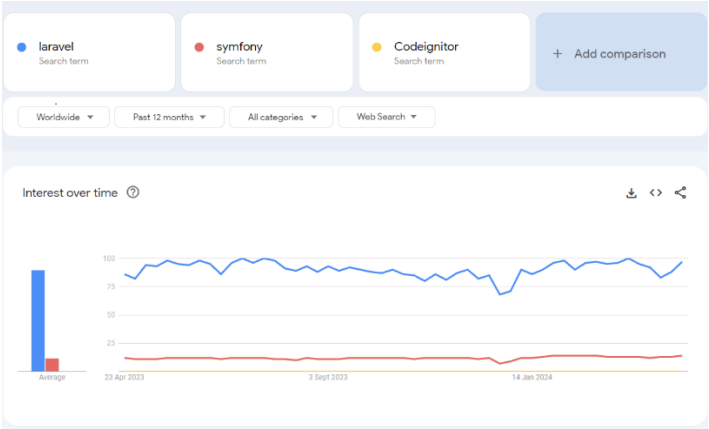


Figure 4. Worldwide Search Result

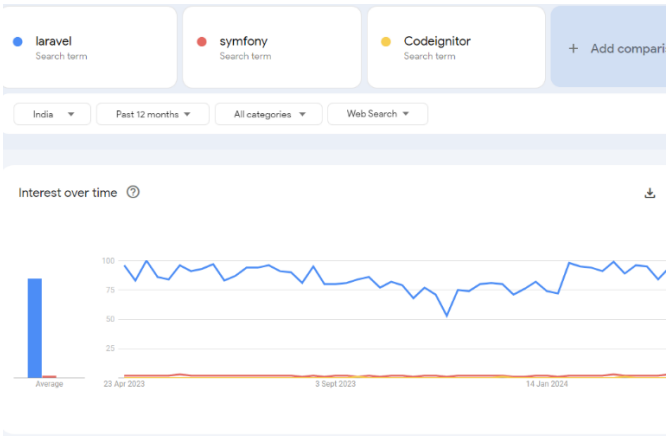


Figure 5 . India Search Result

VI. BEST SECURITY PRACTICE

PHP frameworks are popular tools for web development, but they also come with security risks. If you use a PHP framework, you need to follow some best practices to protect your code, data, and users from hackers, malware, and other threats. In this article, we will cover six essential security practices for PHP frameworks, such as Laravel, Symfony, and CodeIgniter.

1. Validate Input

One of the most common ways that attackers exploit web applications is by injecting malicious code or data into user input. This can lead to SQL injection, cross-site scripting, remote code execution, and other attacks. To prevent this, you should always validate and sanitize user input before processing it. You can use built-in functions or libraries provided by your framework to filter and escape input, or use custom validation rules to ensure that only valid and safe data is accepted

2. Use HTTPS

Hypertext Transfer Protocol Secure (HTTPS) is a protocol that encrypts the communication between your web server and your clients. This prevents hackers from intercepting, modifying, or stealing sensitive information, such as passwords, credit card numbers, or personal details. You should use HTTPS for all your web pages, especially those that handle user authentication or transactions. You can use SSL certificates to enable HTTPS on your web server, and configure your framework to force HTTPS redirection and set secure cookies.

3. Update Regularly

Another way that hackers can exploit your web application is by taking advantage of known vulnerabilities in your framework, libraries, or dependencies. These vulnerabilities can allow attackers to bypass security measures, execute arbitrary code, or access restricted resources. To avoid this, you should always keep your framework and its components updated to the latest versions. You can use tools like Composer or npm to manage your dependencies and check for updates. You should also follow the official documentation and changelogs of your framework to learn about any security patches or fixes.

4. Implement authentication and authorization

Authentication and authorization are two key aspects of web security that deal with verifying the identity and permissions of your users. Authentication is the process of verifying that a user is who they claim to be, usually by asking for a username and password. Authorization is the process of verifying that a user has the right to access or perform certain actions on your web application, usually by checking their roles or privileges. You should implement both authentication and authorization on your web application to control who can access what. You can use features or plugins provided by your framework to handle authentication and authorization, or use third-party services like OAuth or JWT.

5. Prevent CSRF Attacks

Cross-site request forgery (CSRF) is a type of attack that tricks a user into performing an unwanted action on your web application, such as changing their password, transferring money, or deleting their account. This can happen when a user visits a malicious website that sends a forged request to your web application using the user's cookies or session. To prevent this, you should use a CSRF token, which is a random and unique value that is attached to each request and verified by your server. You can use methods or helpers provided by your framework to generate and validate CSRF tokens.

6. Secure File Upload

File uploads are a useful feature for web applications, but they can also pose a security risk. If you allow users to upload files to your web server, you should make sure that they are not malicious or harmful. You should limit the size, type, and number of files that can be uploaded, and scan them for viruses or malware. You should also store the files in a separate and protected directory, and use random and unique names for the files. You can use functions or libraries provided by your framework to handle file uploads securely.[3]

VII. CONCLUSION

The PHP framework requirement decision is also important to be put into consideration whether to use a framework or develop from scratch. There are a few requirements that vital to be considered in the development of web application using framework as shown in Table 2 on the advantages and disadvantages using frameworks [25]. Besides the advantages discussed in Table 2, PHP frameworks exploitation is adequate for a large and complex web application, however, for the small and simple web applications the benefits of frameworks could not be fully utilized, and justification should be made whether the framework usage is over sufficient and thorough consideration or evaluation is required [27]. On the other hand, the core PHP works without any extra library and developers involved should have strong programming skill and logic. Consequently, developers with a strong foundation of core PHP would find PHP frameworks are much easier to understand especially if the basic knowledge of the core PHP such as functions, classes, and methods is deeply well-versed. Looking at this situation for developers who wished to start using frameworks, understanding core PHP to build web application would amazingly help to improves the web development as a whole and developers will not limit themselves to a particular PHP framework only [19]. The beginner developer should also consider learning the easiest framework as discussed by Chao et al [4] to build their knowledge and skills confidence level. Another important thing that should be considered is the framework selection criteria such as built-in feature actively developed or improved, continuous supported from service provider and follow the software engineering practices. Lastly, the conclusion is the choice of the PHP framework is dependent on the web application project in which every project has [1], [4]different requirement or features and sometimes the choice might not quite suitable for the next new project. The PHP framework requirement decision is also important to be put into consideration whether to use a framework or develop from scratch. There are a few requirements that vital to be considered in the development of web application using framework as shown in Table 2 on the advantages and disadvantages using frameworks [25]. Besides the advantages discussed in Table 2, PHP frameworks exploitation is adequate for a large and complex web application, however, for the small and simple web applications the benefits of frameworks could not be fully utilized, and justification should be made whether the framework usage is over sufficient and thorough consideration or evaluation is required [27]. On the other hand, the core PHP works without any extra library and developers involved should have strong programming skill and logic. Consequently, developers with a strong foundation of core PHP would find PHP frameworks are much easier to understand especially if the basic knowledge of the core PHP such as functions, classes, and methods is deeply well-versed. Therefore, developers who wished to start using frameworks are require to understand the core PHP to build web application. The knowledge would amazingly help to improves the web development as a whole and developers will not limit themselves to a particular PHP framework only [19]. The beginner developer should also consider learning the easiest framework as discussed by Chao et al [4] to build their knowledge and skills confidence level. Another important thing that should be considered is the framework selection criteria such as built-in feature actively developed or improved, continuous supported from service provider and follow the software engineering practices. Lastly, the conclusion is the choice of the PHP framework is dependent on the web application project in which every project has different requirement or features and sometimes the choice might not quite suitable for the next new project.
[3], [4], [5]

With the standard framework design methods to style web, leading to large limitations, time-consuming and other issues, this study presents the planning and implementation method of an internet supported Laravel framework, Laravel make the event process is standardized, processing some non-business logic relationship automatically. This paper designs and implements a straightforward Laravel model, which achieved automated processing for a part of the look.

The experimental and simulation proved, web design supported Laravel framework, has scalability and robust scalability, so on improve the developing efficiency. The result analysis for the Laravel and Phalcon shows that the differences between them are relatively small to Laravel's advantage. The Laravel is just a dozen milliseconds faster and consumes some bytes less memory than Phalcon. The results also show that the Laravel application has relatively shorter stack trace in its function calls compared to Phalcon. If the Laravel PHP application was larger or had a stack trace tree of the identical size because the Phalcon version, it might mean that the Laravel wouldn't are performing better than the Phalcon. that's because the expansion of the stack trace within the Laravel application would have caused the next execution time and memory consumpt ion because it has been proven in Laravel is usually fitted to smaller projects whether or not there are not any obvious reasons to not use Phalcon, but Phalcon would be the popular choice when developing larger projects. The result analysis also showed that Phalcon is that the only light weight framework among all the tested frameworks during this study. Followed by Codelgniter, which within the terms of performance has showed mediocre values during this study. The remaining framework Symfony are the heavy weight frameworks performing best both in terms of execution time and memory usage. this can be because of the actual fact that these frameworks have deep stack trace trees in their implementations.

we conclude that CodeIgniter outperforms Pure PHP in this study case (Academic Information System) regarding the results that it achieved along the experiment time even though Pure PHP in some parts outperforms CodeIgniter, but it is because CodeIgniter has more complex system than Pure PHP and despite being more complex, the gap between Pure PHP and CodeIgniter is very narrow. The distance between the results obtained by PHP and the two

frameworks which is not too far apart gives the implication that the use of the framework is still highly recommended with several features embedded in it. Pure PHP can be used if the programmer plans to develop small to medium scale applications or creates his own framework. Meanwhile, the use of the framework can be used on a medium to high level application scale. Using a framework will also help programmers to be able to synergize with each other in developing applications because it is wrapped in the framework ecosystem.

REFERENCES

[1] M. Doroodchi and S. Dastgheib, "A framework for web application development," in *Proceedings of the 2008 International Conference on Software Engineering Research and Practice, SERP 2008*, 2008, pp. 311–316. doi: 10.17148/iarjset.2022.9218.

[2] H. Abutaleb, A. Tamimi, and T. Alrawashdeh, "Empirical Study of Most Popular PHP Framework," in *2021 International Conference on Information Technology, ICIT 2021 - Proceedings*, 2021. doi: 10.1109/ICIT52682.2021.9491679.

[3] A. Niarman, Iswandi, and A. K. Candri, "Comparative Analysis of PHP Frameworks for Development of Academic Information System Using Load and Stress Testing," *International Journal Software Engineering and Computer Science (IJSECS)*, vol. 3, no. 3, pp. 424–436, Dec. 2023, doi: 10.35870/ijsecs.v3i3.1850.

[4] A. M. Joshi and W. Kirti, "PHP Frameworks in Web Application Development," 2021. [Online]. Available: www.ijcrt.org

[5] M. Laaziri, K. Benmoussa, S. Khoulji, K. M. Larbi, and A. El Yamami, "A comparative study of laravel and symfony PHP frameworks," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, pp. 704–712, Feb. 2019, doi: 10.11591/ijece.v9i1.pp704-712.