

DOI: doi.org/10.58797/cser.030303

Testing the Teman Ternak Website Using Black Box Testing with the Equivalence Partitioning Method

Rival Fitrah Dermawan^{*1}, Fahri Radiansyah¹, Anargya Rabbani Aslam¹,
Muhamad Mauladi Fadillah¹, Aditya Wicaksono¹,
Gema Parasti Mindara²

¹Department of Software Engineering Technology, School of Vocational Studies, IPB University, Jl. Raya Darmaga Kampus IPB, Jawa Barat 16680, Indonesia

²Department of Computer Engineering Technology, School of Vocational Studies, IPB University, Jl. Raya Darmaga Kampus IPB, Jawa Barat 16680, Indonesia

*Corresponding Author Email: rivalfitrah@apps.ipb.ac.id

Abstract

Software testing is a crucial stage in the development cycle to ensure program functionality runs as expected and maintains good quality. This process aims to identify errors so they can be fixed before the software is released to users. One of the software objects of this research is the Teman Ternak website. Teman Ternak is a website-based digital platform that functions as telemedicine, designed to overcome distance and time barriers between farmers and veterinarians, providing a platform that allows farmers to obtain remote consultation services, including initial diagnosis and treatment for diseases affecting their livestock. Testing on this website was conducted using the Black Box Testing method by applying the Equivalence Partitions technique. The Black Box method is used to test website functionality from a user perspective without needing to know the internal code structure. Meanwhile, the Equivalence Partitions technique is a testing technique that focuses on designing test cases by grouping input data on each form on the Teman Ternak website. Input data is grouped into test classes (test cases) with expected results of valid or invalid values. The purpose of testing on the Teman Ternak website is to detect and minimize functional failures during implementation, so that errors found can be fixed more quickly and the website quality improves.

Keywords: testing, teman ternak, black box testing, equivalence partitions, quality assurance

Received: 15 September 2025

Revised: 20 October 2025

Accepted: 24 November 2025

Published: 8 December 2025

Issued: 30 December 2025

**Current Steam and
Education Research**

e-ISSN: 3025-8529



INTRODUCTION

In the current digital era, software development has become a vital component in business operations and public services. However, the launch of software is never free from the risk of errors, defects, or bugs (Zhu et al., 2025). Therefore, software testing is viewed as the most crucial stage in the development lifecycle (Terragni et al., 2025). This process serves to verify the system's compliance with functional specifications and to validate whether the software meets business and end-user needs (Anwar & Kar, 2019).

Modern Software Quality Assurance (SQA) discipline has shifted from mere bug detection at the end (Quality Control) to an integrated quality assurance process throughout the entire development lifecycle (Quality Assurance) (Goericke, 2020). Software quality is ideally built from the start (shift-left approach) and maintained continuously (Goericke, 2020). In the context of software Verification and Validation (V&V), testing is identified as the most dominant and fundamental V&V activity for ensuring product quality (Rajabli et al., 2021).

One of the most fundamental testing methods is Black Box Testing. This method focuses on testing the software's functional requirements from an end-user perspective, without requiring knowledge of the internal code structure (Chazette et al., 2022; Robinson et al., 2024). Black Box Testing proves effective in uncovering a class of errors that might not be detected by structural testing (White Box) (Hassija et al., 2023).

To optimize Black Box testing, efficient test case design techniques are required. The Equivalence Partitions technique is a highly relevant method for this purpose (Hendri et al., 2020; Rahadi & Vikasari, 2020). This technique works by dividing the input data domain into equivalence classes (partitions of valid and invalid data). By testing only one representative value from each class, testers can efficiently minimize the number of test cases while ensuring that all possible execution paths have been tested (Hendri et al., 2020; Rahadi & Vikasari, 2020).

Previous research has extensively applied the Black Box Testing method with Equivalence Partitions and consistently demonstrated its effectiveness in finding functional errors in web-based systems. However, these findings also highlight a significant gap between designed functionality and field implementation.

For example, research by Hardika et al. (2024) on the Garuda Farm website found that the success rate of functional testing was only 56.25%, highlighting significant weaknesses in the validation of input dates, images, and mandatory fields that could still be saved even if empty. Similarly, research by Todingrante (2022) on the Extraordinary website also found validation weaknesses in the registration form, which still accepted input containing symbols, numbers, or blank data in critical fields (Todingrante, 2022). Research by Melani (2021) also identified an error gap where the system failed to validate the uploaded file format and allowed the form to be submitted with empty data (Melani & Mahmud, 2021).

These findings (Ningrum et al., 2019; AbuSalim et al., 2021; Amalia et al., 2021) indicate a clear gap analysis: many launched web-based systems still have critical weaknesses in functional data input validation, which can threaten data integrity and user experience.

The object of this research is the Teman Ternak website, a veterinary telemedicine platform that connects farmers with veterinarians for remote consultation services. As a platform that manages sensitive user data (farmers and veterinarians) and transactional flows (consultations and diagnoses), data input functionality on forms (such as registration, login, and livestock data entry) is crucial.

Based on gap analysis from previous research (Hardika et al., 2024; Todingrante, 2022; Melani & Mahmud, 2021) there is an urgent need to conduct systematic functional testing on the Teman Ternak website. While many studies have applied Equivalence Partitions to web-based information systems, such as book sales systems (Shaleh et al., 2021) or transactional library applications (Priyaungga et al., 2020), this research's contribution lies in its application to the specific domain of telemedicine. This testing will focus on critical functional flows related to animal health services, where the accuracy and validation of input data are vital to service reliability.

METHOD

Research Approach

This study employs a qualitative approach with an experimental method through software testing. The selected testing method is Black Box Testing, chosen because it focuses on verifying the functional requirements of the software by observing the output results based on the given input, without the need to understand the internal code structure or system design (Ningrum et al., 2019; AbuSalim et al., 2021)

The use of Black Box Testing is highly relevant to the object of this study, namely the *Teman Ternak* website, as it allows testing from an end-user perspective (Amalia et al., 2021). This approach is essential to ensure that the business processes within the website, such as telemedicine consultations and transactions, operate according to user needs and the expected specifications (Shaleh et al., 2021).

Testing Method

The specific technique used in test case design is Equivalence Partitioning. This technique works by dividing the input data domain on every form on the Teman Ternak website into equivalent data classes (Eisty et al., 2025).

These data classes are then categorized into two types of partitions Valid Partition: Data that complies with business rules and is expected to be accepted by the system **and** Invalid Partition: Data that violates the rules (e.g., empty input, incorrect format, forbidden characters) which is expected to be rejected by the system.

The selection of the Equivalence Partitions technique is based on its efficiency. This technique allows testers to significantly reduce the number of test cases while remaining representative, as they only need to test one value from each equivalence class to represent all members of that class (Hendri et al., 2020; Rahadi & Vikasari, 2020) Thus, testing can cover all possible execution paths with minimal time and resources (Yang et al., 2025).

Research Procedure

The testing procedure to be carried out on the Teman Ternak website follows the systematic stages below: (1) Functionality Identification: Identifying the main website features that require user input, such as the Login page and Consultation form; (2) Test Case Generation: Preparing testing scenarios using a table containing Test ID, Scenario Description, Input Data (based on valid/invalid partition classes), and Expected Result; (3) Execution: Running the Teman Ternak website and inputting data according to the designed scenarios and (4) Result Recording: Documenting the actual results from the system and comparing them with the expected results to determine the status of "Pass" or "Fail".

RESULTS AND DISCUSSION

This section presents the results of the Black Box testing using the Equivalence Partitions technique on the main features of the Teman Ternak website, namely Login, Add Service, Edit Service, Schedule, and Finance. The actual test results are compared with the expected results to determine the validity of the system's functionality.

Login Form Testing

Table 1. Test Case Login

No	Input	Equivalence Class	Expected Result
Email			
1	usertternak	Not an email	Input not accepted because it is not an email
2	blablabla@email.com	Unregistered email	Return error because the email is not registered
3	gibrananggalana@gmail.com	Registered email	Return success because the email is in the system data
Password			
1	“ ”	No Input	Return error "Password must be filled"
2	walalawalala	Wrong Password	Return error "Email or password is

No	Input	Equivalence Class	Expected Result
			incorrect"
3	passwordbenar	Correct Password	Login successful, redirected to dashboard

Table 2. Testing Results and Analysis for Login

No	Input	Equivalence Class	Expected Result	Actual Result	Conclusions
Email					
1	usertternak	Not an email	Input not accepted because it is not an email	Input not accepted	<i>Valid</i>
2	blablabla@email.com	Unregistered email	Return error because the email is not registered	Return error because its not registered	<i>Valid</i>
3	gibrananggalana@gmail.com	Registered email	Return success because the email is in the system data	Return success because the email is registered	<i>Valid</i>
Password					
1	" "	No Input	Return error	Return error	<i>Valid</i>
2	walalawalala	Password salah	Return error "Incorrect email or password"	Return error "Incorrect email or password"	<i>Valid</i>
3	passwordbenar	Password benar	Successful login, redirected to dashboard	Successful login, redirected to dashboard	<i>Valid</i>

The Login test results indicate that the system has good validation for email format and user authentication.

Table 3. Test Case Add Services

No	Input	Equivalence Class	Expected Result
Consultation Name			
1	-=,./[\	Special Symbol	Returns error because it is not a character
2	(no input)	Char = 0	Returns error because there is no character
3	55555	Number Integer	Returns error because the input is not a character
Nominal			
1	-420	< 0	Return Error because the nominal limit cannot be negative/minus.
2	100000000000	0 > Infinite	Return success, data added.
3	0	0	Return Error because the nominal limit cannot be exactly 0.
Duration			
1	-550	Number Integer	Return Error because the nominal limit cannot be negative/minus.
2	10000000	Number Integer	Return success, data added.
3	0	Number Integer	Return Error because the Duration limit cannot be exactly 0.
Description			
1	-=,./[\	Unique Symbol	Return error because it isn't a character
2	(no input)	Char = 0	Return error because there is no input
3	55555	Integer	Return success, data added successfully

Table 4. Testing Results and Analysis for Add Services

No	Input	Equivalence Class	Expected Result	Actual Result	Conclusions
Consultation Name					
1	-=,./[\	Special Symbol	Returns error because it is not a character	Successfully change the Consultation Name	<i>Invalid</i>
2	(no input)	Char = 0	Returns error because there is no character	Returns error because there is no character	<i>Valid</i>
3	55555	Number Integer	Returns error because the input is not a character	Enable to change the name	<i>Invalid</i>
Nominal					
1	-420	< 0	Return Error because the nominal limit cannot be negative/minutes.	Return Success, the data that are negative/minutes can be added.	<i>Invalid</i>
2	100000000000	0 > Infinite	Return success, data added.	Return success, data added.	<i>Valid</i>
3	0	0	Return Error because the Duration limit cannot be exactly 0.	Return success, data added.	<i>Invalid</i>
Durasi					
1	-550	Number Integer	Return Error because the nominal limit cannot be negative/minutes	Return Success, the data that are negative/minutes can be	<i>Invalid</i>

No	Input	Equivalence Class	Expected Result	Actual Result	Conclusions
			s.	added.	
2	10000000	Number Integer	Return success, data added.	Return success, data added.	<i>Valid</i>
3	0	Number Integer	Return Error because the Duration limit cannot be exactly 0.	Return success, data added.	<i>Invalid</i>
Description					
1	-=,./[]\	Unique Symbol	Return error because it isn't a character	Successfully change it with Unique Symbol	<i>Invalid</i>
2	(no input)	Char = 0	Return error because there is no input	Return error because there is no input	<i>Valid</i>
3	55555	Integer	Return success, data added successfully	Return success, data added successfully	<i>Valid</i>

The table above indicates that the system failed to reject invalid input, such as special symbols in the service name and negative values for both the nominal amount and duration. Similar findings were also observed during the testing of the Edit Service feature.

Schedule Form Testing

In the Schedule Form, testing focused on validating the start and end times. Table 4 shows that the system successfully validated empty input, but provided a response different from expectations for past time input.

Table 5. Test Case for Schedule

No	Input	Equivalence Class	Expected Result
Start Time			
1	No Input ()	Date & Time	Return Error because the data cannot be empty
2	02/05/04 & No input()	Date & Time	Return Error because the data cannot be empty
3	02/05/04 & 01.00	Date & Time	Return Error because a data that below a certain year is not possible
4	02/09/25 23:01	Date & Time	Return Success because the data is valid
End Time			
1	No Input ()	Date & Time	Return Error because data cannot be empty
2	02/05/04 & No input()	Date & Time	Return Error because data cannot be empty
3	02/05/04 & 01.00	Date & Time	Return Error because a data that below a certain year is not possible
4	02/09/25 23:01	Date & Time	Return Success because the data is valid and compliant

Table 6. Testing Results and Analysis for Schedule

No	Input	Equivalence Class	Expected Result	Actual Result	Conclusions
Starting Time					
1	No Input ()	Date & Time	Return Error because the data cannot be empty	Return Error karena data tidak boleh kosong	<i>Valid</i>
2	02/05/04 & No input()	Date & Time	Return Error because the data cannot be empty	Return Error karena data tidak boleh kosong	<i>Valid</i>

No	Input	Equivalence Class	Expected Result	Actual Result	Conclusions
3	02/05/04 & 01.00	Date & Time	Return Error because a data that below a certain year is not possible	Return Error karena overlapping dengan data yg ada	<i>Invalid</i>
4	02/09/25 23:01	Date & Time	Return Success because the data is valid	Return Success karena data sesuai	<i>Valid</i>
End Time					
1	No Input ()	Date & Time	Return Error because data cannot be empty	Return Error because data cannot be empty	<i>Valid</i>
2	02/05/04 & No input()	Date & Time	Return Error because data cannot be empty	Return Error because data cannot be empty	<i>Valid</i>
3	02/05/04 & 01.00	Date & Time	Return Error because a data that below a certain year is not possible	Return Error because the data is overlapping with eachothers	<i>Invalid</i>
4	02/09/25 23:01	Date & Time	Return Success because the data is valid and compliant	Return is success	<i>Valid</i>

Although the system rejected the past date input, the resulting error message ("overlapping") did not align with the initial expectation ("data below a certain year is not allowed"), thus it is categorized as Invalid in the context of error message appropriateness.

Keuangan Form Testing

The Keuangan Form testing concluded with highly successful results. All established test scenarios, which covered account numbers, bank selection, and the withdrawal amount fields, were found to be Valid.

Table 7. Test Case for Keuangan Form

No	Input	Equivalence Class	Expected Result
Account Number			
1	(no input)	Int = 0	Return error because no data is filled
2	1234567890123	10 < x < 16 number	Return success, because data is valid
3	-123123123123	< 0	Return error because data cannot be less than 0
4	123456789012345678	> 16 number	Return error because the data cannot be more than 16 number
Bank			
1	No Input	Char	Return error because there is no data is filled.
2	BRI	Char	Return success because BRI is listed
3	VISA	Char	Return error because VISA is not listed
Withdrawal Amount			
1	(no input)	Int = 0	Return error because there is no data filled.
2	5000	Int >12000	Return error because the data is lower than the minimum amount.
3	12000	Int >12000	Return success because the data is in range.
4	1.000.000	Int 0 >12000 > n	Return error because the inputed data exceeds the existing savings amount

Table 8. Testing Result and Analysis for Keuangan Form

No	Input	Equivalence Class	Expected Result	Actual Result	Conclusions
Account Number					
1	(no input)	Int = 0	Return error because no data is filled	Return error because no data is filled	Valid
2	1234567890123	10 < x < 16 number	Return success, because data is valid	Return success, because data is valid	Valid
3	-123123123123	< 0	Return error because data cannot be less than 0	Return error because data cannot be less than 0	Valid
4	123456789012345678	> 16 number	Return error because the data cannot be more than 16 number	Return error	Valid
Bank					
1	No Input	Char	Return error because there is no data is filled.	Return error because there is no data is filled.	Valid
2	BRI	Char	Return success because BRI is listed	Return success because BRI is listed	Valid
3	VISA	Char	Return error because VISA is not listed	Return error because VISA is not listed	Valid
Withdrawal Amount					
1	(no input)	Int = 0	Return error because there is no data filled.	Return error because there is no data filled.	Valid
2	5000	Int >12000	Return error because the data is lower than the	Return error because the data is lower than the	Valid

No	Input	Equivalence Class	Expected Result	Actual Result	Conclusions
			minimum amount.	minimum amount.	
3	12000	Int >12000	Return success because the data is in range.	Return success because the data is in range.	Valid
4	1.000.000	Int 0 >12000 > n	Return error because the inputed data exceeds the existing savings amount	Return error because the inputed data exceeds the existing savings amount	Valid

The system successfully validated the minimum withdrawal limit and the availability of bank options in accordance with business rules.

The Black Box Testing, utilizing the Equivalence Partitions technique, proved effective in identifying functional gaps on the Teman Ternak website, a finding consistent with prior research on its capability to uncover latent errors. The primary issues identified centered on validation failures within the "Add/Edit Service" feature. Specifically, the system allows for the input of negative nominal and duration values, as well as service names containing special symbols. This suggests a significant absence of proper interface or server-side validation for both data types and value constraints. This vulnerability mirrors a common weak point observed in other application studies: inadequate input validation. Such failures in a system meant to be a precise telemedicine platform pose a risk of data inconsistency.

On the other hand, the Login and Finance features demonstrated a high level of functional maturity. Successful validation of sensitive features like finance is crucial for maintaining user trust. For the Schedule feature, although invalid input was rejected, the discrepancy between the actual and expected error messages indicates the need for improvements to the error handling mechanism to provide more accurate information to users. Overall, while some key features are working well, a critical bug discovery in the service input validation requires immediate attention to ensure the quality and reliability of the Teman Ternak website before full release.

.CONCLUSION

Based on the results of the testing and analysis conducted, this study successfully answered the research objectives with the following findings this study successfully designed and implemented test scenarios (test cases) using the Black Box Testing method with the Equivalence Partitions technique. Testing was conducted on crucial functionality of the Teman Ternak website, including

the Login, Service Management (Add and Edit), Schedule, and Finance features. Testing results for functional errors indicated that the Teman Ternak website's functionality was largely functioning well, but critical bugs were still identified. The Login and Finance features performed Valid across all test scenarios, demonstrating the maturity of the validation and authentication systems in those areas. However, significant functional errors were found in the Add Service and Edit Service features. The system failed to validate invalid input, such as still accepting special symbols in service names and negative and zero (0) values in the nominal and duration fields. These validation failures have the potential to cause data inconsistencies and errors in the system. In the Schedule feature, a discrepancy was found in the error handling mechanism, where the error message displayed to the user (related to overlapping) did not match the expected test scenario (related to past date input). Overall, this study concludes that the Equivalence Partitions technique is very effective in revealing input validation weaknesses on the Teman Ternak website.

Future research should focus on immediately patching the validation mechanism in Service Management and fixing error messages in the Schedule feature. Additionally, research could implement complementary black-box techniques like Boundary Value Analysis (BVA) or non-functional testing like usability testing to assess the website's usability.

REFERENCES

- AbuSalim, S. W. G., Ibrahim, R., & Wahab, J. A. (2021). Comparative Analysis of Software Testing Techniques for Mobile Applications. *Journal of Physics: Conference Series*, 1793(1), 012036. <https://doi.org/10.1088/1742-6596/1793/1/012036>
- Amalia, A., Putri, W., & Kristanto, T. (2021). Pengujian Black Box Menggunakan Teknik Equivalence Partitions Pada Aplikasi E-Learning Berbasis Web. *Building of Informatics, Technology and Science (BITS)*, 3(3), 269–274. <https://doi.org/10.47065/bits.v3i3.1062>
- Anwar, N., & Kar, S. (2019). Review Paper on Various Software Testing Techniques & Strategies. *Global Journal of Computer Science and Technology*, 19(2), 43–49. <https://doi.org/10.34257/gjcstcvol19is2pg43>
- Chazette, L., Brunotte, W., & Speith, T. (2022). Explainable software systems: from requirements analysis to system evaluation. *Requirements Engineering*, 27(4). <https://doi.org/10.1007/s00766-022-00393-5>
- Eisty, N. U., Kanewala, U., & Carver, J. C. (2025). Testing research software: an in-depth survey of practices, methods, and tools. *Empirical Software Engineering*, 30(3). <https://doi.org/10.1007/s10664-025-10620-6>
- Goericke, S. (2020). *The future of software quality assurance*. Springeropen.
- Hardika, B., Kurniawan, M. D., Adzka, M., Prastowiyono, D., Banyubasa, A., Wicaksono, A., & Nasir, M. (2024). Pengujian Blackbox Testing Website Garuda Farm Menggunakan Teknik Equivalence Partitioning. *Jurnal Kridatama Sains Dan Teknologi*, 6(02), 740–753. <https://doi.org/10.53863/kst.v6i02.1420>
- Hassija, V., Chamola, V., Mahapatra, A., Singal, A., Goel, D., Huang, K., Scardapane, S., Spinelli, I., Mahmud, M., & Hussain, A. (2023). Interpreting Black-Box Models: a Review on Explainable Artificial Intelligence. *Cognitive Computation*, 16(1), 45–74. <https://doi.org/10.1007/s12559-023-10179-8>
- Hendri, H., Hasiholan Manurung, J. W., Ferian, R. A., Hanaatmoko, W. F., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Pengelolaan Masjid Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 3(2), 107. <https://doi.org/10.32493/jtsi.v3i2.4694>
- Melani, Y. I., & Mahmud, N. (2021). Black Box Testing Using Equivalence Partition Method in Sintana Application. *Atlantis Highlights in Engineering/Atlantis Highlights in Engineering*. <https://doi.org/10.2991/ahe.k.210205.089>

- Ningrum, F. C., Suherman, D., Aryanti, S., Prasetya, H. A., & Saifudin, A. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 4(4), 125–130. <https://doi.org/10.32493/informatika.v4i4.3782>
- Priyaungga, B. A., Aji, D. B., Syahroni, M., Aji, N. T. S., & Saifudin, A. (2020). Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 3(3), 150. <https://doi.org/10.32493/jtsi.v3i3.5343>
- Rahadi, N. W., & Vikasari, C. (2020). Pengujian Software Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions. *Infotekmesin*, 11(1), 57–61. <https://doi.org/10.35970/infotekmesin.v11i1.124>
- Rajabli, N., Flammini, F., Nardone, R., & Vittorini, V. (2021). Software Verification and Validation of Safe Autonomous Cars: A Systematic Literature Review. *IEEE Access*, 9(1), 4797–4819. <https://doi.org/10.1109/ACCESS.2020.3048047>
- Robinson, D., Cabrera, C., Gordon, A. D., Lawrence, N. D., & Mennen, L. (2024). Requirements are All You Need: The Final Frontier for End-User Software Engineering. *ACM Transactions on Software Engineering and Methodology*, 34(5). <https://doi.org/10.1145/3708524>
- Shaleh, I. A., Yogi, J. P., Pirdaus, P., Syawal, R., & Saifudin, A. (2021). Pengujian Black Box pada Sistem Informasi Penjualan Buku Berbasis Web dengan Teknik Equivalent Partitions. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 4(1), 38. <https://doi.org/10.32493/jtsi.v4i1.8960>
- Terragni, V., Vella, A., Roop, P., & Blincoe, K. (2025). The Future of AI-Driven Software Engineering. *ACM Transactions on Software Engineering and Methodology*, 34(5). <https://doi.org/10.1145/3715003>
- Todingrante, G. (2022). Implementasi Black Box Testing Pada Website Extraordinary. *KHARISMA Tech*, 17(1), 135–148.
- Yang, L., Wei, C., Yang, J., Xia, W., Yang, Y., Luo, Y., Niyato, D., Sun, L., & Liu, Z. (2025). FuzzCoder: Code Large Language Model-Based Fuzz Testing for Industrial IoT Programs. *IEEE Internet of Things Journal*, 12(18), 36842–36851. <https://doi.org/10.1109/jiot.2025.3577602>
- Zhu, X., Zhou, W., Han, Q.-L., Ma, W., Wen, S., & Xiang, Y. (2025). When Software Security Meets Large Language Models: A Survey. *IEEE/CAA Journal of Automatica Sinica*, 12(2), 317–334. <https://doi.org/10.1109/jas.2024.124971>

