

Combinatorial Testing for Identifying Defect Patterns in Manufacturing

Maslita Abd Aziz¹, Kamal Z. Zamli², Zuriani Mustaffa³

¹Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka (UTeM),
Durian Tunggal, Melaka, 76100, Malaysia

^{2,3}Fakulti Komputeran, Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Pekan, 26600,
Pahang, Malaysia

DOI: <https://dx.doi.org/10.47772/IJRISS.2025.909000665>

Received: 24 September 2025; Accepted: 30 September 2025; Published: 25 October 2025

ABSTRACT

In the manufacturing industry, improving product quality and reducing defects are crucial objectives. This study investigates the use of combinatorial testing to analyse defect patterns in a manufacturing setting. We utilised a dataset containing various defects attributes on available open-source Kaggle datasets. Pairwise test cases were generated using hybrid metaheuristics to systematically explore interactions between these attributes. The proposed method significantly reduced the number of test cases while ensuring comprehensive coverage of pairwise interactions, compared to exhaustive testing. Results indicate that the combinatorial testing approach effectively identifies defect patterns, reducing the time span for defect identification. The integration of reward and penalty mechanisms with the Roulette Wheel algorithm in our hybrid metaheuristic optimisation process further enhanced the efficiency of candidate solutions for combinatorial testing. This study provides a practical framework for improving defect detection and quality control in manufacturing settings, highlighting the benefits of advanced combinatorial testing techniques.

Keywords: Combinatorial testing; Hybrid metaheuristic; Pairwise; Manufacturing defect

INTRODUCTION

The modern manufacturing industry is continually evolving, with quality control becoming more critical as products and processes grow increasingly complex [1][2]. Standard approaches to defect analysis often overlook the complex interactions among multiple variables in a manufacturing process. One promising approach to achieve this is through combinatorial testing [3], which systematically examines interactions among multiple factors to identify defect patterns. This research investigates the use of combinatorial testing paired with hybrid metaheuristics to analyse defect patterns in manufacturing settings. This combined approach aims to enhance the detection and understanding of defects, leading to improved quality control and operational efficiency [4].

Studies focused on defect pattern analysis in manufacturing have traditionally relied on statistical methods and machine learning techniques. However, these methods often require extensive historical data and may not effectively capture interactions between multiple process parameters. Statistical Quality Control (SQC) and control charts, for instance, typically assume that variables are independent and normally distributed. However, this assumption does not always hold true in complex manufacturing environments, leading to potential failures in detecting defects that arise from interactions among multiple variables [5].

Similarly, Root Cause Analysis (RCA) methods, which often rely on historical data, may not be effective in dynamic and high-dimensional settings where new types of defects can emerge from interactions between process variables [6]. Furthermore, while Pareto Analysis focuses on identifying the most significant factors

contributing to defects, it may overlook the complex interplay between less significant factors that collectively have a substantial impact [7].

Recent studies have highlighted the effectiveness of combinatorial testing in various fields, including software engineering and network security [8]. Combinatorial testing could significantly reduce the number of tests required while maintaining high defect detection rates in software systems. Applying this approach to manufacturing could yield similar benefits by identifying defects resulting from complex interactions among process variables.

Hybrid metaheuristics have gained popularity for their ability to tackle complex optimisation problems by combining different heuristic methods [9][10]. Blum and Roli [11] discussed how these algorithms leverage the strengths of individual heuristics to produce more robust and efficient solutions. In manufacturing, hybrid metaheuristics can optimise the combinatorial testing process, making it feasible to analyse many variable combinations without excessive computational costs.

Recent advancements in manufacturing, especially those aligned with Industry 4.0, emphasize the need for intelligent systems capable of real-time defect detection and analysis [12][13][14][15]. This research aligns with these advancements by proposing a methodology that integrates combinatorial testing and hybrid metaheuristics, supporting real-time and accurate defect pattern analysis.

METHODOLOGY

A. Hybrid Relay Algorithm

The hybrid metaheuristic optimisation algorithm uses a 4 x 100 relay concept incorporating reward and penalty mechanisms. The Hybrid Relay (HR) consists of four metaheuristic algorithms that are Jaya Algorithm, Cuckoo Search, Sine Cosine Algorithm and Flower Pollination Algorithm. The selection is generated via the Roulette Wheel algorithm that will generate a unique combination representing the four algorithms as In Figure 1.

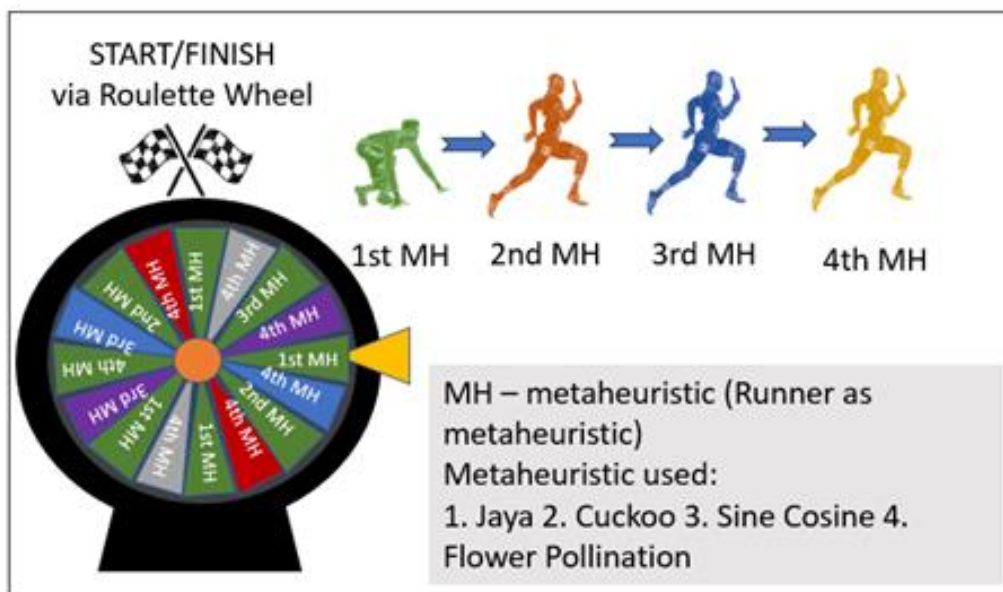


Fig. 1. Relay Hybrid Algorithm Model

Figure 2 shows the decision-making process, as well as the different possible outcomes and the probability of each outcome. The first decision point is whether the value of RelayBest is greater than or equal to XBest. If it is, the algorithm rewards the current relay sequence by using the same relay sequence. If RelayBest is less than XBest, the algorithm penalizes the current sequence by generating a new sequence with probability greater than 0.5 and selects the next sequence using the Roulette Wheel algorithm.

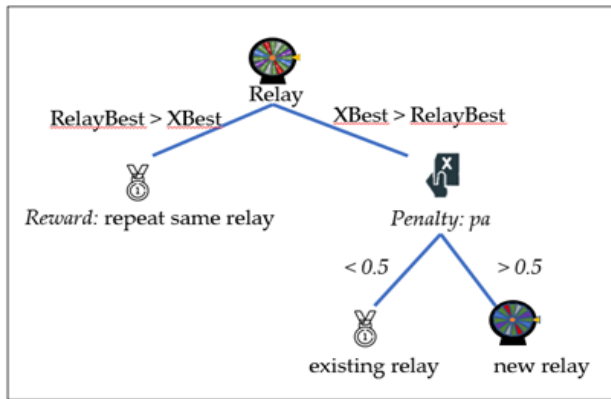


Fig. 2. Reward and Penalty implementation

B. Covering Array

There is at least one covering array (CA) for every t -way combination of parameter values. With t is the number of related parameters to be interacted known as the interaction strength. For example, when considering two parameters, it constitutes 2-way testing; with three parameters, it corresponds to 3-way testing, and so forth. CAs have proven useful in numerous industries, and researchers are exploring the best approaches to develop optimal CAs. The CA and its notation,

$$CA(N; t, k, v) \quad (1)$$

is a mathematical object that ensures all possible combinations of a specific number of input parameters (factors) are tested [16][17]. It helps to identify and eliminate potential interactions among these parameters. In the given example, we have a covering array with:

t (number of rows or experiments): 2

k (number of factors or input parameters): 7

v (number of levels or possible values for each factor): 2

If all possible interactions are tested, it will be $2^7 = 128$ test cases to execute. Instead of testing all combination, CA with $t=2$, $k=7$, and $v=2$ where each row represents an experiment, and each column represents a factor can produce a test suite to include every 2-way combination. The array is constructed in such a way that every combination of the k factors appears at least once in the array.

The HR will find the minimum test case by eliminating the tuple based on weightage until the covering array table is empty. The covering array or interaction table will be generated to be compared with random generated test case. For example, extending from example $t=2$, $k=7$, and $v=2$, the strength is $t=2$, therefore the interaction table will be generated for pairing the factors. Let assume factor 1 as A, factor 2 as B and factor 3 as C until factor 7 as G. Therefore, the interaction table for $t=2$ is AB, AC and BC and so forth as in Table 1.

TABLE I. PAIR OF FACTORS OF $t=2$

2-way interactions					
AB	AC	AD	AE	AF	AG
BC	BD	BE	BF	BG	
CD	CE	CF	CG		
DE	DF	DG			
EF	EG				
FG					

For simplicity we denote the first value as 0 and the second as 1. Each pair of factors can have the following combinations as in Table 2 show example for AB and CD.

TABLE II. ALL POSSIBLE PAIRWISE COMBINATIONS

A	B		C	D
0	0		0	0
0	1		0	1
1	0		1	0
1	1		1	1

With these generated tuples, a set of test case are randomly generated, for example 00000000. With this test case, AB = 00, AC = 00, BC = 00 etc are removed and the weight is 7. The process is repeated by generating the next test case and keep on removing the tuple until the table is empty.

The outcome of the process yields the optimal number of test cases. These test cases play a pivotal role in revealing interactions and dependencies among the parameters (denoted by 'k'). With this example the minimum test cases produce is 8 (Table 3). Furthermore, they ensure comprehensive coverage by testing each combination at least once [18]. As illustrated in Figure 3, HR will be running until the t-tuple table is empty incorporating the steps in Figure 1 and Figure 2.

TABLE III. INTERACTION TABLE

Test Case	A	B	C	D	E	F	G
1	0	0	0	0	0	0	0
2	0	0	0	1	1	1	1
3	0	1	1	0	0	1	1
4	1	0	1	0	1	0	1
5	1	1	0	1	0	1	0
6	1	0	0	1	1	0	0
7	0	1	1	1	0	0	1
8	1	1	1	1	1	1	0

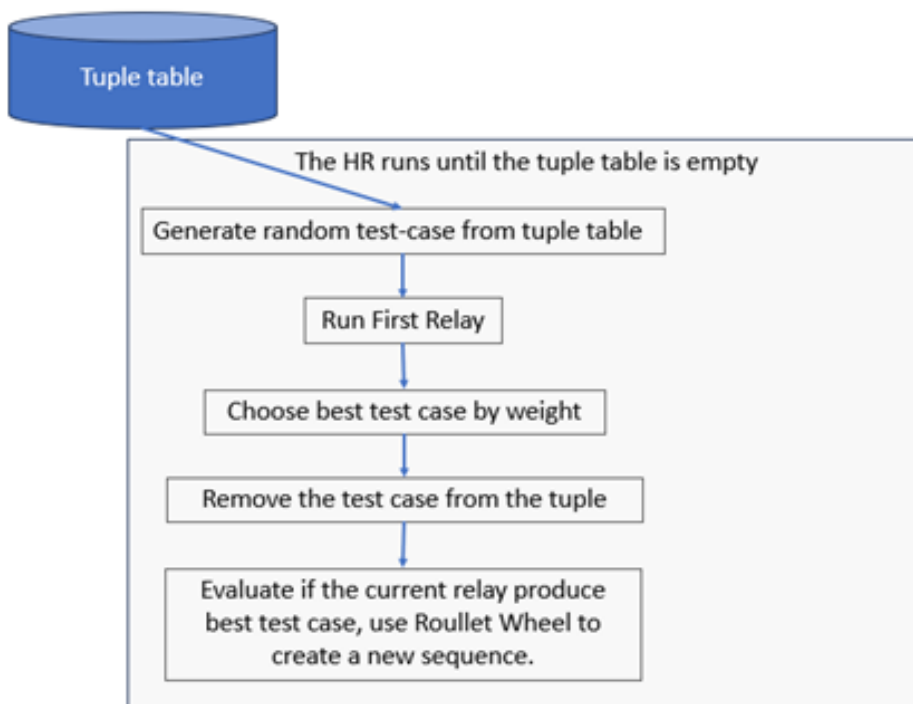


Fig. 3. Outline of the proposed Hybrid Relay Algorithm

C. Dataset

In this proposed case study, the Manufacturing Defect Dataset sourced from Kaggle was utilized. The simulated dataset is related to manufacturing defects observed during quality control process. The dataset encompasses 100 products, featuring parameters such as defect type, detection date, location within the product, severity level, inspection method used, and repair costs., as illustrated in Figure 4.

defect_id	product_id	defect_type	defect_date	defect_location	severity	inspection_method	repair_cost
1	15	Structural	6/6/2024	Component	Minor	Visual Inspection	245.47
2	6	Functional	4/26/2024	Component	Minor	Visual Inspection	26.87
3	84	Structural	2/15/2024	Internal	Minor	Automated Testing	835.81
4	10	Functional	3/28/2024	Internal	Critical	Automated Testing	444.47
...	...	...	...	...	...	...	...
998	17	Structural	1/16/2024	Component	Minor	Automated Testing	401.12
999	96	Cosmetic	6/21/2024	Internal	Moderate	Manual Testing	775.63
1000	1	Cosmetic	3/23/2024	Component	Minor	Visual Inspection	963.4

Fig. 4. Samples of manufacturing defect with varying features

RESULT AND DISCUSSION

The dataset examined in this study comprises attributes associated with manufacturing defects, including defect ID, product ID, defect type, defect date, defect location, severity, inspection method, and repair cost. For the combinatorial testing application, three primary factors were selected: defect type (Cosmetic, Structural, Functional), defect location (Component, Surface, Internal), and inspection method (Automated Testing, Visual Inspection, Manual Testing). These factors were prioritized due to their direct relevance to defect detection and analysis in manufacturing processes. Figure 5 illustrates the input interface for these factors and their corresponding values, mapped to levels 0, 1, and 2 for computational purposes.

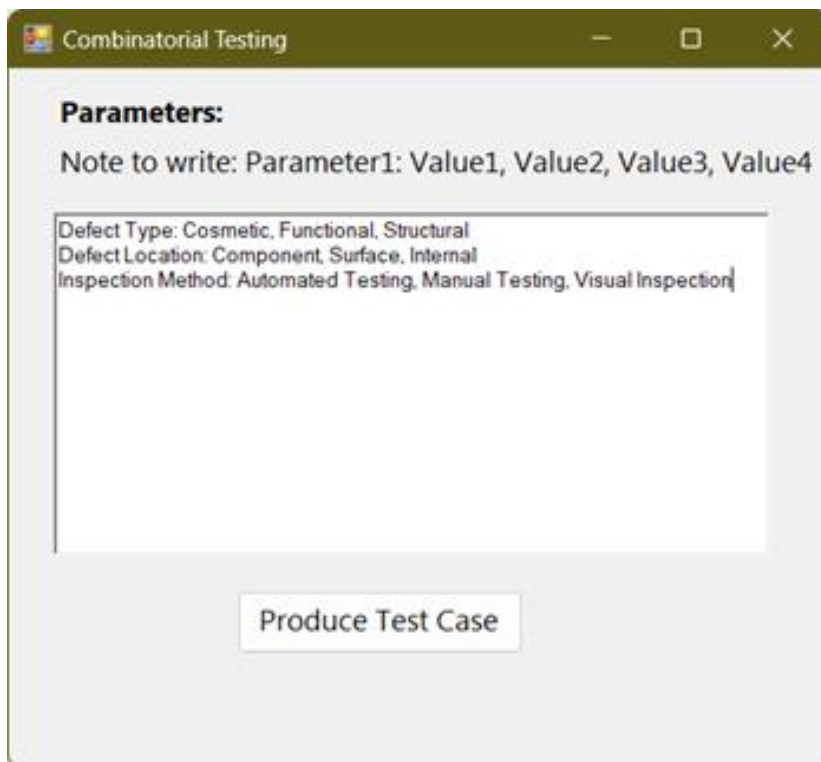


Fig. 5. Interface to input the factors and values

This configuration corresponds to a covering array denoted as $CA(N; 2, 3, 3)$, where the interaction strength (t) is 2, the number of factors (k) is 3, and the number of levels per factor (v) is 3. In contrast to exhaustive

testing, which would necessitate $3^3 = 27$ test cases to cover all possible combinations, the hybrid relay algorithm generated an optimized set of 9 test cases, as depicted in Figure 6.

This reduction ensures full coverage of all pairwise interactions while minimizing the testing effort. Figure 7 further presents the distribution of tests conducted per product in the original dataset, revealing that defect identification for a single product could involve up to 20 tests, highlighting inefficiencies in conventional approaches.

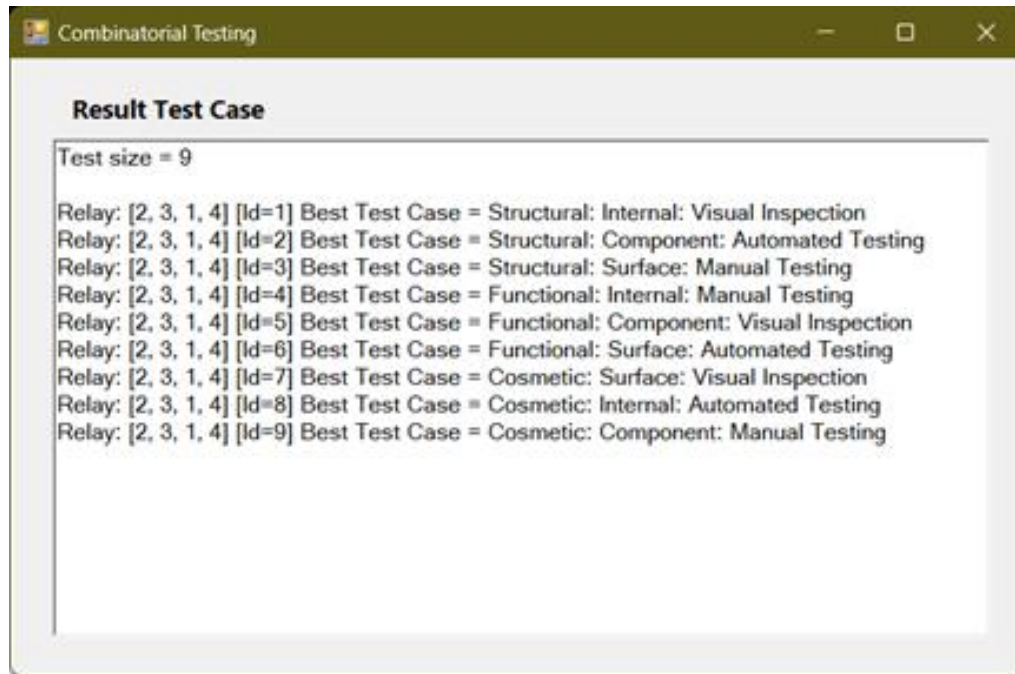


Fig. 6. Optimized test cases

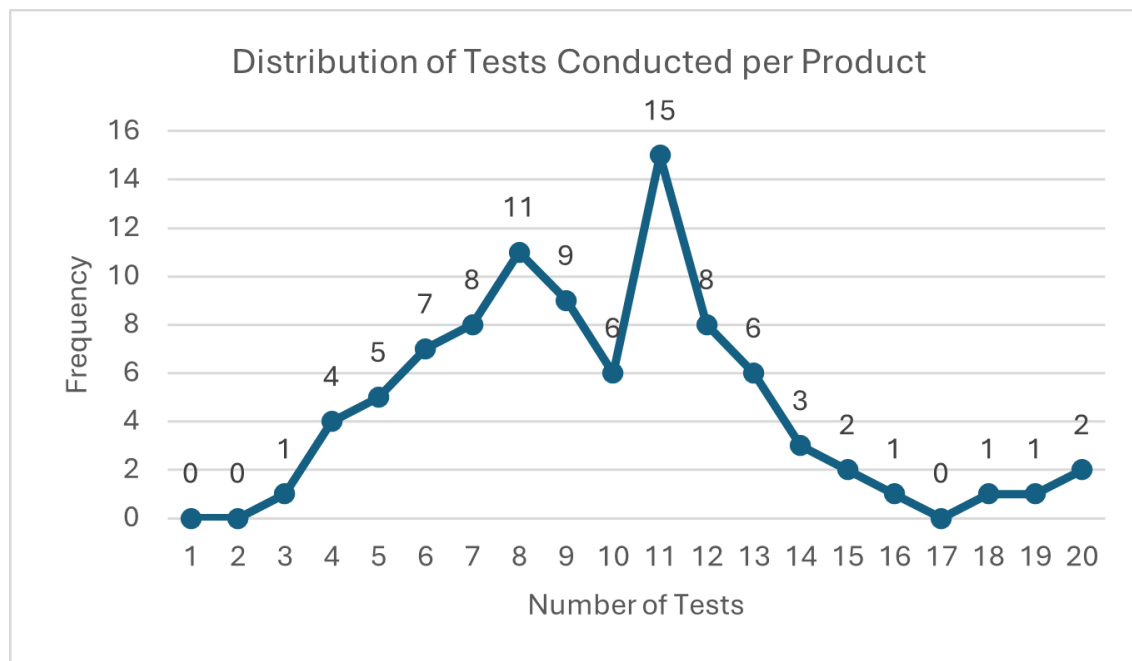


Fig. 7. Distribution of tests conducted per product

Table 4 exemplifies the extended time span for defect identification in one product (Product ID 1), where defects were recorded from February 22, 2024, to November 1, 2024 denote a period exceeding eight months. The optimized test cases from the combinatorial approach could potentially compress this timeline by systematically targeting interactions among the selected factors, thereby accelerating pattern recognition.

TABLE IV. TIME SPAN TO IDENTIFY A DEFECT

Defect id	Product id	Defect type	Defect date	Defect location	Inspection method	Repair cost
109	1	Cosmetic	22/2/2024	Component	Automated Testing	978.5
194	1	Structural	3/3/2024	Component	Visual Inspection	192.21
1000	1	Cosmetic	23/3/2024	Component	Visual Inspection	963.4
699	1	Cosmetic	4/4/2024	Component	Automated Testing	940.9
874	1	Cosmetic	19/4/2024	Surface	Visual Inspection	98.45
577	1	Functional	21/4/2024	Surface	Visual Inspection	404.22
538	1	Structural	11/5/2024	Component	Automated Testing	856.33
390	1	Structural	19/5/2024	Surface	Automated Testing	564.46
872	1	Functional	20/5/2024	Internal	Automated Testing	148.15
772	1	Cosmetic	11/6/2024	Surface	Manual Testing	514.52
104	1	Structural	25/6/2024	Surface	Manual Testing	338.48
979	1	Functional	1/11/2024	Internal	Manual Testing	652.97

The results demonstrate that the combinatorial testing method, through the CA(N; 2, 3, 3) configuration, achieves a substantial 67% reduction in test cases compared to exhaustive testing, while maintaining comprehensive pairwise coverage. This efficiency is particularly evident in the ability to pinpoint defect patterns, such as recurring combinations of defect type and location under specific inspection methods, which may contribute to prolonged identification periods as shown in Table 4 and Figure 7. By reducing the number of required tests, the approach not only streamlines the process but also minimizes resource allocation in quality control.

Beyond this pairwise analysis, other covering arrays are feasible with the dataset to explore deeper or broader interactions. For instance, increasing the interaction strength to $t=3$ with the same three factors yields CA(N; 3, 3, 3), which would require up to 27 test cases for full triple coverage but could be optimized via the hybrid algorithm to approximately 20 to 25 cases, capturing more complex interdependencies like the joint effects of defect type, location, and inspection method on severity. Alternatively, incorporating an additional factor, such as severity (discretized into Low, Medium, High levels), results in CA(N; 2, 4, 3), expanding exhaustive requirements to 81 test cases but optimizable to around 15 to 18, enabling analysis of pairwise interactions including cost implications. Extending further to five factors (adding binned repair cost) at $t=2$ could produce CA(N; 2, 5, 3) with an optimized N of 20 to 30, facilitating a more holistic view of defect dynamics.

The significance of combinatorial testing in this case study lies in its capacity to address limitations of traditional methods, such as SQC or RCA, by explicitly accounting for variable interactions that drive defects in manufacturing environments [3][5][6]. In similar cases, where processes involve multiple interdependent parameters, this technique promotes zero-defect manufacturing by enabling rapid, cost-effective pattern detection, reducing downtime, and supporting predictive maintenance [1][2][12]. Ultimately, it offers manufacturers a scalable framework for quality enhancement, adaptable to Industry 4.0 demands for real-time analytics [13][14].

CONCLUSION

In conclusion, the use of combinatorial testing with hybrid metaheuristics has proven to be an effective approach for analysing defect patterns in manufacturing settings. By focusing on key factors such as defect type, defect location, and inspection method, we were able to identify critical defect patterns with a reduced number of test cases. This method offers significant advantages over traditional testing approaches, providing a valuable tool for manufacturers to enhance their quality control processes.

ACKNOWLEDGEMENT

The authors would like to express gratitude to Centre of Advanced Communication Technology (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi (FTMK), Universiti Teknikal Malaysia Melaka (UTeM) and Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah for their invaluable support and resources provided throughout this research.

REFERENCES

1. F. Psarommatis, M. Vuichard, and D. Kiritsis, "Improved heuristics algorithms for re-scheduling flexible job shops in the era of Zero Defect manufacturing," *Procedia Manufacturing*, vol. 51, pp. 1485–1490, 2020.
2. F. Psarommatis and D. Kiritsis, "Identification of the Inspection Specifications for Achieving Zero Defect Manufacturing," in *IFIP International Conference on Advances in Production Management Systems (APMS)*, F. Ameri, K. E. Stecke, G. von Cieminski, and D. Kiritsis, Eds., Austin, TX, United State: Springer International Publishing, pp. 267–273, 2019.
3. B. S. Ahmed, E. Enoiu, W. Afzal, and K. Z. Zamli, "An evaluation of Monte Carlo-based hyper-heuristic for interaction testing of industrial embedded software applications," *Soft Computing*, vol. 24, no. 18, pp. 13929–13954, 2020.
4. A. Hassan and N. Pillay, "Hybrid metaheuristics: An automated approach," *Expert Systems with Applications*, vol. 130, pp. 132–144, 2019.
5. N. Heigl, B. Schmelzer, F. Innerbichler, and M. Shivhare, "Statistical Quality and Process Control in Biopharmaceutical Manufacturing - Practical Issues and Remedies," *PDA Journal of Pharmaceutical Science and Technology*, p. pdajpst.2020.011676, Jan. 2021.
6. C. Hagedorn, J. Huegle, and R. Schlosser, "Understanding unforeseen production downtimes in manufacturing processes using log data-driven causal reasoning," *Journal of Intelligent Manufacturing*, vol. 33, no. 7, pp. 2027–2043, 2022.
7. M. M. Potomkin, A. A. Sedliar, O. V Deineha, and A. O. Zvarych, "Comprehensive Use of the Pareto Principle and the Analytic Hierarchy Process to Increase the Substantiation of Alternative Ranking Results," *Cybernetics and Systems Analysis*, vol. 57, no. 3, pp. 422–428, 2021.
8. H. M. Fadhil, M. Najm, and M. Younis, "Combinatorial Testing Approaches: A Systematic Review," *Iraqi Journal of Computer, Communication, Control and System Engineering*, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:259599940>
9. A. A. Muazu, A. S. Hashim, and A. Sarlan, "Review of Nature Inspired Metaheuristic Algorithm Selection for Combinatorial t-Way Testing," *IEEE Access*, vol. 10, pp. 27404 – 27431, 2022.
10. S. T. Milan, L. Rajabion, H. Ranjbar, and N. J. Navimipour, "Nature inspired meta-heuristic algorithms for solving the load-balancing problem in cloud environments," *Computers and Operations Research*, vol. 110, pp. 159–187, 2019.
11. G. R. Blum, Christian and Raidl, *Hybrid metaheuristics: powerful tools for optimization*. Springer, 2016.
12. A. Bousdekis, K. Lepenioti, D. Apostolou, and G. Mentzas, "Data analytics in quality 4.0: literature review and future research directions," *International Journal of Computer Integrated Manufacturing*, vol. 36, no. 5, pp. 678–701, May 2023.
13. A. Rana, G. Gupta, P. Vaidya, W. Salehi, S. Basheer, and M. Bhatia, "Techniques Based on Metaheuristics Combined with an Adaptive Neurofuzzy System and Seismic Sensors for the Prediction of Earthquakes," *Journal of Sensors*, vol. 2023, no. 1, p. 5063981, 2023.
14. C. Zhang et al., "A review on learning to solve combinatorial optimisation problems in manufacturing," *IET Collaborative Intelligent Manufacturing*, vol. 5, no. 1, p. e12072, 2023.
15. L. Baty, K. Jungel, P. S. Klein, A. Parmentier, and M. Schiffer, "Combinatorial Optimization-Enriched Machine Learning to Solve the Dynamic Vehicle Routing Problem with Time Windows," *Transportation Science*, vol. 58, no. 4, pp. 708–725, Feb. 2024.
16. R. N. Kacker, D. R. Kuhn, Y. Lei, and D. E. Simos, "Correction to: Factorials Experiments, Covering Arrays, and Combinatorial Testing," *Mathematics in Computer Science*, vol. 15, no. 4, p. 741, 2021.

17. C. Wu, Huayao and Xu, Lixin and Niu, Xintao and Nie, “Combinatorial testing of restful apis,” in Proceedings of the 44th International Conference on Software Engineering, pp. 426--437, 2022.
18. F. Klück, Y. Li, J. Tao, and F. Wotawa, “An empirical comparison of combinatorial testing and search-based testing in the context of automated and autonomous driving systems,” Information and Software Technology, vol. 160, 2023.
19. M. Sánchez, J. M. Cruz-Duarte, J. c. Ortiz-Bayliss, H. Ceballos, H. Terashima-Marin, and I. Amaya, “A Systematic Review of Hyper-Heuristics on Combinatorial Optimization Problems,” IEEE Access, vol. 8, pp. 128068–128095, 2020.
20. Y. Hanaka, T., Kiyomi, M., Kobayashi, Y., Kobayashi, Y., Kurita, K., & Otachi, “A Framework to Design Approximation Algorithms for Finding Diverse Solutions in Combinatorial Problems,” in Proceedings of the AAAI Conference on Artificial Intelligence, pp. 37(4), 3968-3976, 2023.