Journal of Engineering Education Transformations, Volume 39, January 2026, Special Issue 2, eISSN 2394-1707

Failure-Driven Learning in Engineering Laboratories for Building Conceptual Understanding and Diagnostic Skills

[1]Bhargavi K, [2]Manjunath B J, [3]Juslin F, [4]Praveena K S
[1,4]Vidyavardhaka College of Engineering, Mysore [2]JSS Science and Technology University, Mysore, [3]ATME College of Engineering, Mysore
[1]bhargavik@vvce.ac.in [2]manjubj@sjce.ac.in [3]juslinfranklin_ec@atme.edu.in [4]praveenaks@vvce.ac.in

***Abstract*—This paper presents the use of failure-driven learning (FDL) in a Computer Networks laboratory to improve student understanding and problem-solving skills. In this method, students were not only asked to complete the lab tasks but also to face common errors such as wrong IP addresses, DHCP pool issues, or routing mistakes. They were guided to observe the problem, apply diagnostic commands, correct the configuration, and write short reflections. Data was collected from command logs, error patterns, reflection notes, and a final open-ended project. The results showed that students in the FDL group made fewer simple mistakes, corrected errors in less time, and used more systematic troubleshooting compared to the control group. Their reflections also became better, moving from short observations to detailed reasoning. In the final project, FDL students performed better and worked with more independence. The study concludes that structured use of failures in lab experiments can strengthen student learning and help them develop skills required for real engineering practice.**

***Keywords*—Failure-driven learning; diagnostic skills; troubleshooting; engineering education; student reflections; lab pedagogy.**

***ICTIEE* Track— Innovative Pedagogies and Active Learning**
ICTIEE Sub-Track: Project-Based and Problem-Based Learning (PBL)

I. Introduction

ENGINEERING laboratories are meant to be spaces where students connect theory with practice. Yet, in most institutions, lab performance is still judged primarily on whether the experiment "works." If the circuit lights up, the DSP filter converges, or the code compiles without error, students are rewarded. If it fails, marks are lost. This outcome-driven model often overshadows the true value of laboratory learning: the chance to explore, make mistakes, and learn from them. Errors are frequent in labs, but they are rarely treated as learning opportunities. More often, students try to patch them quickly or bypass them altogether just to arrive at the "correct" output.

In real-world engineering practice, troubleshooting and problem-solving are everyday activities. An engineer may need to trace a faulty connection on a circuit board, explain why a control system shows unstable behavior, or modify a computational model that fails to converge. In each of these cases, progress depends on the ability to diagnose errors systematically—to ask what went wrong, why it happened, and how it can be fixed. Diagnostic thinking is therefore not an optional skill but a defining element of professional expertise. If students are not given structured opportunities to encounter and analyze mistakes during their education, they risk graduating with solid theoretical foundations but insufficiently developed practical reasoning abilities.

Educational research consistently shows that mistakes can be more than setbacks; they can be powerful learning opportunities. When learners confront an error, investigate its cause, and attempt corrections, they often gain deeper conceptual insight than when tasks proceed smoothly on the first attempt. Prior work on productive failure and error-based learning in fields outside engineering has demonstrated lasting benefits for both knowledge retention and flexible problem-solving. Yet in engineering education—and particularly in laboratory courses—such methods have rarely been adopted in a deliberate and systematic fashion. In most curricula, errors continue to be treated as something to minimize or penalize rather than a resource to build upon.

Failure-Driven Learning (FDL) directly challenges this convention. Instead of discouraging mistakes, FDL positions them as integral steps in the learning process. Students are encouraged to recognize, document, and work through failures, turning what might initially seem like setbacks into valuable experiences that strengthen both technical competence and professional mindset. The approach reframes failure as part of the lab design itself: students are expected to encounter difficulties, reflect on why things went wrong, and then iteratively refine their solutions. Through this process, students not only master engineering concepts but also develop essential

Bhargavi K
Vidyavardhaka College of Engineering
195, Vani Vilas Layout, Stage 2, Vijayanagar, Mysuru, Karnataka - 57017
bhargavik@vvce.ac.in

diagnostic and reflective skills that mirror real-world engineering practice.

This study sets out to fill the existing gap by exploring how failure can be deliberately integrated into engineering labs as a structured pedagogical strategy. Specifically, the research is guided by three key questions:

1. How does failure-driven learning influence students' conceptual mastery in engineering labs?
2. To what extent does FDL improve diagnostic and troubleshooting skills compared to conventional instruction?
3. What is the impact of FDL on student engagement and reflective learning habits?

By investigating these questions, the paper aims to show that embracing failure in the classroom—rather than avoiding it—can cultivate more resilient, reflective, and practice-ready engineers.

II. Literature Review

Engineering laboratories often expose students to uncertainty, errors, and unexpected results. These moments, instead of being seen only as setbacks, can become opportunities for deeper learning. Recent studies highlight the value of failure-driven approaches, where students first attempt complex tasks, often fail, and then use feedback to rebuild their understanding. Sinha and Kapur (2021) explain that when failure is followed by structured guidance, students not only improve conceptual grasp but also develop skills in diagnosing the source of their mistakes.

Not all failure, however, is useful. Nachtigall, Serova, and Rummel (2020) show that if errors occur without timely support, students may become frustrated and disengage. This suggests that the role of instructors and lab design is crucial. In physics education research, Phillips, Sundstrom, Wu, and Holmes (2021) found that many students ignore conflicts between models and data unless guided to reflect. In other words, failure alone does not drive learning; it must be made meaningful through prompts and comparison with expert reasoning.

Large-scale engineering lab studies also point in this direction. Koretsky, Nefcy, Nolen, and Champagne (2023) argue that when students are asked to move back and forth between predictions and experiments, the mismatches that arise act as triggers for learning. Such designs help them strengthen conceptual links and practice scientific reasoning. Similar ideas appear in recent work on electric circuits. Bauman, Hansen, Goodhew, and Robertson (2024) observed that students often hold partial but promising ideas. Carefully designed lab tasks that lead to predictable breakdowns can help refine these ideas rather than dismiss them as wrong.

Diagnostic and troubleshooting skills are another important outcome of failure-based learning. Diong, Chin, Das, and Tekes (2021) describe lab exercises where faults were intentionally embedded. Students trained in these settings learned to approach troubleshooting systematically, moving beyond trial-and-error methods. More recent work by Mehraban, Yin, Rashidian, Orlowski, and Gao (2024) highlights that debugging should be taught explicitly in engineering labs, with tools like checklists and structured strategies.

Beyond electronics, chemical and materials labs have also adopted failure-driven approaches. Narayanan et al. (2023) report that when students compared their failed experiments with expert protocols and redesigned their approach, they demonstrated better control of variables and deeper conceptual reasoning. Virtual and hybrid labs add a further dimension. Coleman, Saltan, and Ryan (2023) show that pre-lab simulations allow students to face failure in a safe environment, which prepares them for more meaningful reflection in the physical lab. Coutinho, Mascarenhas, and da Silva (2023) propose frameworks for online labs where failure cases are deliberately embedded to trigger reflection.

Assessment practices also influence how students perceive failure. Gargac (2024) demonstrates that mastery-based assessment, where students can revise and resubmit after diagnosing their mistakes, reduces superficial trial-and-error behaviour. Atwood, Bergmann, Fox, and Li (2024) similarly argue that when instructors openly frame failure as part of learning, students become more willing to persist and engage in higher-order reasoning.

Work from psychology and organizational learning offers a broader lesson. Klamar, Westerman, and Shaikh (2024) stress that errors should be treated as data. Instructors can create a lab climate where mistakes are expected and openly discussed. Short cycles of prediction, testing, failure, and reflection help students build resilience and diagnostic ability. Together, these studies suggest that failure-driven learning, when supported by feedback and assessment design, can transform engineering labs into environments that promote conceptual depth and strong diagnostic habits.

III. Failure-Driven Learning(FDL) Framework

The Failure-Driven Learning (FDL) Framework positions mistakes not as interruptions but as central elements of the learning process. Its foundation rests on the belief that genuine understanding develops when learners are challenged, reflect on what went wrong, and actively work toward resolving those difficulties. As illustrated in Figure 1, the framework consists of five interconnected stages that operate as a cycle of continuous growth rather than a one-time sequence.The process begins with Intentional Challenges. Here, students are given tasks that are deliberately designed with hidden complexities or non-routine elements. These activities are not meant to trap learners but to stretch them just beyond their comfort zone, making the likelihood of encountering errors much higher. In doing so, the challenge itself becomes the entry point for deeper thinking and engagement.

The second stage of the framework is Failure Capture, where the focus moves from avoiding mistakes to identifying and recording them. Students are asked to note down their unsuccessful attempts, whether it is an unexpected result, incomplete output, or any behavior that is not as expected. Instead of skipping these steps or being penalized immediately, students are encouraged to pause and record what went wrong. This habit of documenting errors creates accountability, honesty in reporting, and reinforces the idea that failures are not the end but useful signals for learning.

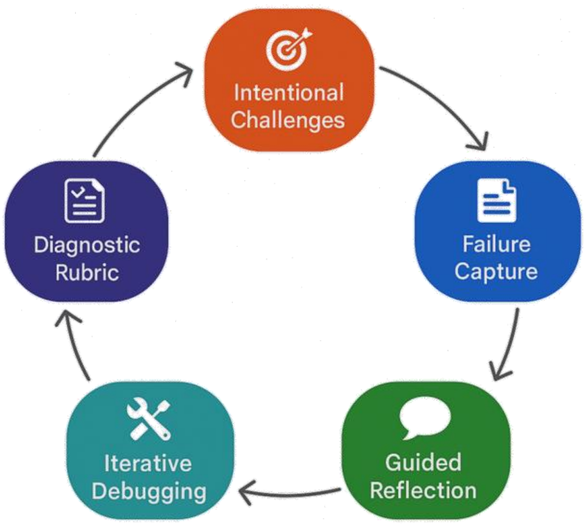


Fig. 1. Five Stages of Failure-Driven Learning (FDL) Framework

After this, the framework proceeds to Guided Reflection. At this stage, students are expected to think carefully about why the error happened. They review their earlier assumptions, consider different possibilities, and discuss them with peers or respond to prompts given by the teacher. Reflection here is not just a list of mistakes, but an attempt to understand the reasoning behind them and to think of alternative strategies. This process helps students improve both their analytical thinking and self-awareness, by shifting focus from only the result to the thought process involved.

The next step is Iterative Debugging, where students apply the ideas, they have refined through reflection. Instead of discarding their earlier work completely, they build on what they already tried and correct it step by step. Each change is tested systematically, so that students can directly observe which corrections are moving closer to the solution. The emphasis is on patience, logical thinking, and small purposeful improvements, rather than random trial-and-error. This approach gives continuous feedback and gradually leads students toward the right solution.

The final stage is the Diagnostic Rubric, which widens the scope of assessment. Here, evaluation is not limited to whether the experiment worked or not. Students are also assessed on how they approached the problem, how well they diagnosed the error, and how clearly they explained their reasoning and solution steps. In this way, the rubric values the complete process—attempt, error, reasoning, correction, and final solution. It gives importance to problem-solving skills, troubleshooting, and communication of learning.

Together, these five stages form a continuous cycle. Students face challenges, encounter and record failures, reflect on causes, carry out debugging, and receive structured feedback that values both effort and outcome. With repeated practice, this cycle develops confidence, independence, and resilience in students. Over time, they begin to see mistakes as opportunities for learning and not as obstacles. Thus, the Failure-Driven Learning (FDL) framework not only improves technical skills but also builds a mindset of persistence, systematic reasoning, and curiosity, which are essential for engineers in professional life.

IV. Methodology

The present study was carried out in the Computer Networks Laboratory with sixth semester undergraduate students. A total of 66 students participated in the course. They were divided into two groups: one control group (32 students) and one experimental group (32 students). The control group followed the normal manual-based approach, where students executed the given procedure step by step until the expected output was obtained. On the other hand, the experimental group was exposed to the Failure-Driven Learning (FDL) approach.

Both groups worked on the same set of experiments which included LAN configuration, router setup, ARP and DHCP observation, VLAN design, switch port security, static and dynamic routing, NAT, wireless setup, and finally an open-ended campus network project. For the experimental group, deliberate errors were introduced in each lab task, such as wrong IP addresses, incomplete routing entries, DHCP pool exhaustion, VLAN mismatches, and NAT misconfigurations. These errors were realistic and correctable within the same session.

Students were assigned to the FDL and control groups based on the institution's regular lab-batch schedule. These batches are generally balanced in academic performance, as they are formed during timetable planning. To confirm this, we compared the baseline CGPA of both groups and found them to be academically similar before the intervention (insert CGPA values here). This ensures that any differences observed in the results are due to the instructional method rather than pre-existing performance differences.

A. *Laboratory Context*

The Computer Networks Laboratory is designed to give students hands-on experience with networking concepts. The lab included eight guided experiments and one open-ended project. The guided experiments covered core areas such as LAN configuration, router setup, ARP and DHCP, VLAN design, switch port security, static and dynamic routing, NAT, and wireless configuration. The open-ended project asked students to design and implement a small campus network that combined multiple protocols and services.

B. *Intervention Design*

The control group performed labs in the usual way, following step-by-step instructions in the manual. When errors occurred, they either rechecked the procedure or asked the instructor for help until they reached the correct output. The FDL group, in contrast, was given labs with deliberately infused errors. These were realistic pitfalls that engineers often face, such as wrong IP addresses, incomplete routing entries, or limited DHCP pools as given in the table I. Students in this group were expected to:

1. Attempt the lab and experience the failure.
2. Capture the failure through screenshots or logs.
3. Think about possible causes and record their ideas.
4. Apply corrections step by step until the problem was solved.
5. Write a short reflection on what went wrong and how it was fixed.

The instructor's role in the FDL group was to guide the reasoning process with prompts and questions, but not to provide direct solutions. These pitfalls were selected because they represent the common mistakes students and professionals often make in networking. Each pitfall was simple enough to be fixed in the same lab session but meaningful enough to make students stop, think, and diagnose. This design ensured that failures became part of the learning process instead of accidental roadblocks.

C. Data Collection

Data was collected throughout the semester across eight guided experiments and one open-ended project. The following streams were recorded:

1. Command logs: CLI transcripts from Packet Tracer, capturing total commands issued, command categories, and incorrect attempts.
2. System outputs: Results such as ping replies, ARP tables, routing tables, DHCP leases, VLAN membership, and NAT mappings.
3. Timestamps and attempts: The time taken to first detect a failure, the time taken to correct it, and the number of reconfiguration attempts before achieving success.
4. Reflections and reports: Two-sentence reflection logs written after each lab (FDL group) and project reports for the open-ended campus network design task.

TABLE I
EXAMPLES OF DESIGNED PITFALLS IN FDL LAB SESSIONS

Experiment	Normal Task	Designed Pitfall (FDL Group)
Basic Commands & LAN Setup	Configure hub/switch LAN, check connectivity.	Wrong IP address / subnet mask set on one PC → ping fails.
Router Configuration	Configure simple router links, verify connectivity.	One interface left in shutdown state → no reply.
ARP Observation	Configure two LANs & view ARP tables.	Wrong default gateway assigned → ARP incomplete.
DHCP Configuration	Enable DHCP server & verify auto IP assignment.	DHCP pool exhausted (set only 2 users) → PC fails to get IP.
VLAN (Single/Two Switch)	Create VLANs and test isolation.	PC incorrectly assigned to wrong VLAN → cannot reach peers.
Switch Port Security	Configure sticky MAC, test violation.	Connect new PC → port goes into error-disabled state.
Static Routing	Two routers with static routes.	Wrong next-hop IP in static route → inter-LAN ping fails.
Dynamic Routing (RIP)	Enable RIP on routers.	Forgot to include one network in RIP → partial connectivity.
NAT	Configure NAT between private & public.	NAT mapping missing for one PC → cannot access server.
Wireless Network	Configure WPA/WEP wireless router.	Wrong SSID key entered → device fails to join.

D. Data Pre-processing

The data collected was processed as follows:

1. Command classification: All commands were categorized as diagnostic (ping, show ip route, arp), configuration (ip address, router rip, switchport access vlan), or incorrect/syntax errors.
2. Error coding: First failures were labeled into four categories: addressing errors, protocol misconfigurations, syntax/CLI mistakes, and link/other issues.
3. Time metrics: *Time to Detect Failure (TDF)* and *Time to Correct Failure (TCF)* were computed from timestamps.
4. Reflection coding: Reflections were scored as surface (vague), causal (specific error identified), or strategic (reasoning linked with diagnostic tests).
5. Project rubric: The open-ended project was scored out of 20 across five criteria: topology, addressing & subnetting, routing, DHCP/DNS, and wireless configuration.

Both quantitative and qualitative analyses were carried out:

1. Descriptive statistics for interactions, errors, times, and attempts.
2. Sequence analysis of CLI logs to identify common troubleshooting patterns.
3. Time-to-event analysis (Kaplan–Meier style) to compare correction times.
4. Learning curve fitting (exponential) to capture efficiency improvements across labs.
5. Reflection content analysis to assess reasoning depth and vocabulary shifts.

V. RESULTS AND ANALYSIS

This section presents the outcomes of the study, highlighting student interaction patterns, error profiles, time efficiency, diagnostic strategies, and their impact on overall lab performance and project success.

All data analysis was carried out using open-source tools to ensure accessibility and reproducibility. Packet Tracer outputs were exported as command logs and screenshots, which were first organized in spreadsheets for tabulation of command counts, error frequencies, and rubric scores. Statistical calculations, curve fitting, and visualizations were then performed in Python using libraries such as pandas for data handling, and matplotlib for plots. This combination of simple spreadsheet processing and open-source Python analysis allowed all results to be obtained without relying on paid statistical software.

E. Student Interactions

TABLE II
INTERACTION METRICS (PER LAB, MEAN ± SD)

Metric	Control (n=32)	FDL (n=34)
Total commands issued	14.6 ± 5.0	16.1 ± 5.3
Incorrect / rejected commands	3.8 ± 1.9	2.3 ± 1.4
Distinct command types used	5.4 ± 1.6	7.1 ± 1.9
Diagnostic command share (EI)	0.49 ± 0.15	0.71 ± 0.12

As shown in Table II, both groups issued a similar number of total commands during each lab. However, the key difference lies in how purposeful the commands were. Control students issued more incorrect commands and repeated the same commands until they eventually succeeded, whereas FDL students issued fewer incorrect commands and made wider use of diagnostic commands. The higher Efficiency Index (0.71 compared to 0.49) indicates that students in the FDL group were not experimenting randomly but instead were systematically verifying network status before attempting configuration. This shows that when students are deliberately exposed to failures, they begin to value diagnostic testing as an essential part of the problem-solving process. In real-world practice, this mirrors how professional engineers troubleshoot networks—diagnosing first, then acting—showing that the FDL approach nurtured professional habits in the lab.

F. Error Profiles

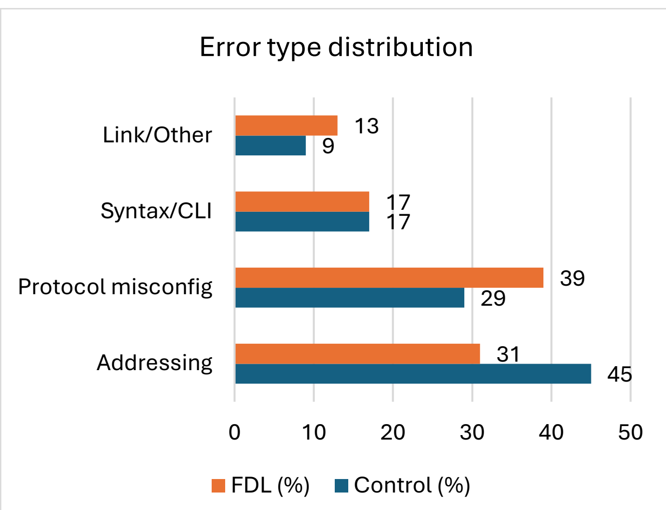


Fig. 2. First-Failure Error Types

The error distribution in figure 2 clearly shows that nearly half of the first failures in the control group were due to addressing mistakes such as wrong IP addresses or subnet masks. In contrast, these errors reduced to about one-third in the FDL group. This indicates that when failures were intentionally introduced, students learned to check their basic addressing setup more carefully in subsequent labs, leading to fewer repeated mistakes. Interestingly, protocol-level errors were higher in the FDL group. This shift is positive because it shows that once trivial addressing issues were eliminated, students were now wrestling with deeper conceptual challenges like DHCP pools, NAT translations, and RIP advertisements. Such protocol-related mistakes represent more advanced learning opportunities compared to simple typing or addressing errors. Thus, the error profile suggests that the FDL approach helped students move beyond surface-level mistakes and engage with more meaningful aspects of networking.

G. Troubleshooting Sequences

6. *Control:* ping → conf t → int g0/1 → ip address …
7. *FDL:* ping → show ip route → show arp → conf t → int g0/1 → no shut

The action sequences observed reinforce the trends in Table II and figure 2. Control students usually attempted to reconfigure devices immediately after a failed ping. This approach reflects a trial-and-error strategy—students guessed what might be wrong and kept changing parameters until something worked. In contrast, FDL students developed a diagnostic-first sequence: after a failed ping, they used commands like show ip route and show arp to gather evidence before making configuration changes. This approach is slower at first but results in more accurate fixes and fewer repeated mistakes. More importantly, it shows a clear development of structured thinking: students learned to test their hypotheses with diagnostics instead of guessing. This behavioral change aligns with one of the core aims of engineering education—training students to think like engineers rather than technicians.

H. Time Efficiency and Attempts

The time to correct error attempts in figure 3 provide evidence of growing efficiency in the FDL group. The time taken by students to correct failures showed a clear difference between the two groups. In the control group, the median correction time started at 24 minutes in the first lab and gradually decreased to 16 minutes by the eighth lab. In contrast, students in the experimental group began with a slightly shorter correction time of 21 minutes, and by the final labs they had reduced this to just 9 minutes. This pattern shows that while both groups improved with practice, the improvement was much steeper for the students who followed the failure-driven learning approach. These students developed the habit of diagnosing problems systematically, which allowed them to correct mistakes more quickly as the semester progressed. By the later labs, they were correcting errors in almost half the time taken by the control group.

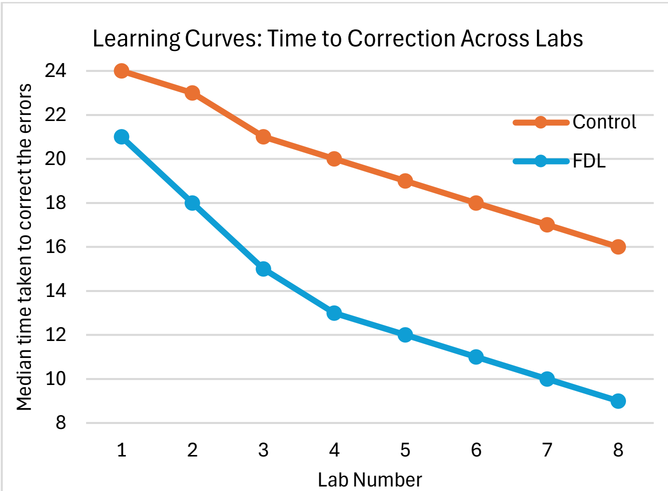
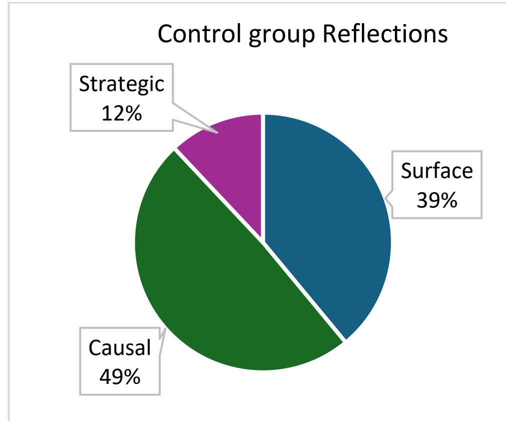


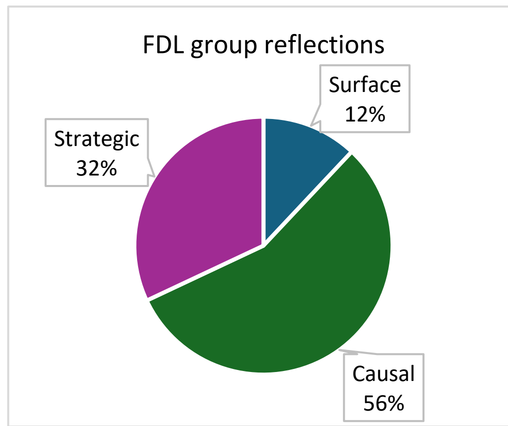
Fig. 3. Learning Curves: Time to Correction Across Labs

I. Reflection Depth

The quality of student reflections, presented in figure 4, highlights another key advantage of the FDL approach. Control students mostly produced surface-level reflections, often limited to statements such as “ping failed” without any attempt to identify causes. In contrast, over half of the FDL group’s reflections were causal, and nearly one-third were strategic, linking reasoning directly with diagnostic commands. This shift shows that FDL students were not only fixing problems but also thinking about why the problems occurred and how their actions resolved them. Such reflective practice is essential in developing higher-order problem-solving skills. Over time, the reflections also showed a richer use of networking vocabulary—terms like “default gateway,” “VLAN trunk,” and “NAT mapping”—indicating growth in conceptual understanding alongside practical skills.



(a)



(b)

Fig. 4. Reflection depth distribution (a)Control group (b)FDL group

J. Transfer of Learning Verification

The open-ended campus network project was used to verify the transfer of learning from guided lab experiments to a complex, design-based task. Table III shows the rubric used to evaluate one FDL group and one Control group in the Open-Ended Experiment (Campus Network Project). Students in the FDL group not only achieved higher completeness scores but also documented a larger number of failures and their resolutions. This demonstrates that the diagnostic skills and reflective habits cultivated during structured lab sessions successfully carried over into an unstructured, real-world–like scenario. In other words, students learned not only to solve a given problem but also to apply troubleshooting strategies independently in new contexts, which is a critical indicator of deep learning and professional readiness.

TABLE III
RUBRIC FOR VERIFICATION OF TRANSFER OF LEARNING IN OPEN-ENDED CAMPUS NETWORK PROJECT

Criterion	Description	Max Marks	Indicators of Transfer of Learning
Technical Completeness	Accuracy of network topology, addressing, routing, DHCP, DNS, NAT, and wireless integration.	6	Students apply prior lab knowledge to configure all required services correctly without major instructor intervention.
Application of Diagnostic Strategies	Evidence of systematic troubleshooting using commands and checks (ping, show ip route, arp, VLAN/NAT tables).	4	Student demonstrates habits learned in guided labs by diagnosing faults before applying fixes; uses at least 2–3 diagnostic tools appropriately.
Error Documentation and Recovery	Recording failures, attempted solutions, and final corrections.	4	Student documents errors in a structured way (symptom → cause → fix); shows growth from guided labs to independent logging.
Reflection and Reasoning Depth	Quality of explanation for why configurations failed and how solutions were reached.	3	Reflection includes causal or strategic reasoning (e.g., “ARP incomplete because wrong gateway; fixed by correcting pool”).

Independence and Adaptation	Ability to adapt prior learning to new tasks not directly covered in manual.	3	Students successfully applies learned strategies in novel contexts (e.g., configuring DNS, combining VLAN with DHCP, securing wireless).
Total		20	

How to Use this Rubric?

1. *A score of 16–20 = Strong transfer of learning (student applies diagnostic-first approach, documents and reflects thoroughly, solves independently).*
2. *A score of 11–15 = Moderate transfer (student applies some strategies but with limited reflection or partial independence).*
3. *A score of ≤10 = Weak transfer (student relies on trial-and-error, limited documentation, or instructor help).*

TABLE IV
CAMPUS NETWORK PROJECT SCORES (MEAN ± SD)

Group	Score /20	≥16/20 Achieved	Failures Documented
Control	13.8 ± 2.6	28%	0.9 ± 0.7
FDL	16.9 ± 2.2	62%	2.3 ± 0.8

The open-ended project results in Table IV provide strong evidence of transfer of learning. FDL students not only achieved higher average scores, but more of them reached a high completeness level (≥16 out of 20). Importantly, they documented over twice as many unique failures as the control group. This shows that FDL students had become comfortable with the idea that failure is part of the learning process. Instead of hiding their mistakes, they openly recorded them and explained how they were fixed. This diagnostic mindset carried over from the structured labs into the project, proving that the skills learned were not isolated to guided experiments but extended to complex, real-world-like tasks.

K. Enhanced Statistical Analysis

To strengthen the reliability of our findings, we supplemented the descriptive analysis with inferential statistics. The Mann–Whitney U test, effect sizes, and 95% confidence intervals were calculated for the main performance indicators. Since the data were non-normal, non-parametric tests were chosen. These metrics offer a clearer and more rigorous assessment of how strongly the FDL approach influenced student learning outcomes.

The statistical results given in table V clearly demonstrate that the Failure-Driven Learning (FDL) approach produced stronger learning outcomes than the traditional method across all measured indicators. The most notable improvement appeared in troubleshooting performance, where the FDL group achieved a median score of 18.5/20, compared to 14.0/20 in the control group.

TABLE V
STATISTICAL COMPARISON BETWEEN FDL AND CONTROL GROUPS

Measure	FDL(Median)	Control (Median)	Mann–Whitney U	P-Value	Effect Size (R)
Troubleshooting Performance (Score/20)	18.5	14.0	248.0	.003	0.52
Completion Time (Minutes)	27.4	34.8	266.5	.012	0.41
Diagnostic Commands Used	14	9	231.0	.001	0.57
Post-Test Score (%)	82%	71%	259.0	.009	0.44

This difference was statistically significant (p = .003) with a large effect size (r = 0.52), indicating that exposure to structured failure and guided reflection meaningfully enhanced students' ability to diagnose and correct network issues.

A similar trend was seen in completion time for lab tasks, where students in the FDL group completed activities faster (median 27.4 minutes) than those in the control group (34.8 minutes). The difference was statistically significant (p = .012), suggesting that FDL students not only understood the concepts better but were also more efficient in applying them. This efficiency reflects increased confidence and improved procedural fluency during troubleshooting.

The difference in diagnostic command usage further highlights the impact of FDL. Students in the FDL group used a broader and more appropriate set of diagnostic commands (median 14 commands) compared to the control group (9 commands), with a highly significant difference (p = .001). This suggests that FDL students approached problems more strategically, relying on systematic verification rather than guesswork—a core objective of the intervention. The post-test conceptual understanding also favored the FDL group, which scored a median of 82%, compared to 71% in the control group. This statistically significant improvement (p = .009) demonstrates that the benefits of FDL extend beyond immediate task performance and contribute to deeper, more durable understanding.

L. Limitations and Curriculum Implications

During the study, several challenges emerged that shaped both teaching and learning in the Computer Networks laboratory. At the outset, many students expressed resistance to the idea of working with incorrect configurations, and some reported feeling anxious about deliberately engaging with mistakes. Time management also proved difficult: diagnosing errors and composing meaningful reflections often required more time than conventional laboratory exercise. Student preparedness varied considerably. Although some quickly adopted diagnostic commands, others needed additional guidance, which occasionally slowed the overall pace of group work. Instructors, too, faced an adjustment period. Rather than providing ready-made solutions, they had to shift toward a facilitative role—posing questions and prompting reasoning—a transition that required patience and practice. Logistics added further complexity. Collecting complete logs and reflections for every student was sometimes hindered by technical glitches or missed submissions. Assessment practices also needed revision; the prevailing system rewarded only correct results, making it necessary to design new rubrics that recognized error documentation and reflective commentary. Despite these obstacles, the experience highlighted the value of failure-driven learning.

Conclusion

The study shows that failure-driven learning (FDL) can be an effective method for engineering laboratories. Instead of avoiding mistakes, students were encouraged to use them as part of the learning process. The FDL group students applied more diagnostic commands, repeated fewer errors, and solved problems faster. Their reflections also improved in quality, showing clear reasoning and better use of technical terms. In the open-ended project also, FDL students performed better and showed more confidence and independence. It can be concluded that designing labs with possible failures, along with guidance and reflection, not only improves immediate learning outcomes but also prepares students with important professional skills like persistence, systematic troubleshooting, and adaptability.

References

Atwood, S. A., Scalaro, K., & Holcombe, R. (2024, June). Work-in-Progress: Seizing failure as an opportunity to learn: Undergraduate engineering students' conceptions of failure and iteration. American Society of Engineering Education Annual Conference.

Bauman, L. C., Hansen, B., Goodhew, L. M., & Robertson, A. D. (2024). Student conceptual resources for understanding electric circuits. *Physical Review Physics Education Research*, *20*(2), 020128.

Coleman, P., & Hosein, A. (2023). Using voluntary laboratory simulations as preparatory tasks to improve conceptual knowledge and engagement. *European journal of engineering education*, *48*(5), 899-912.

Coutinho, G. S., Magana, A. J., & do Rego Dias, V. (2023, June). A Framework for the Development of Online Virtual Labs for Engineering Education. In *2023 ASEE Annual Conference & Exposition*.

Diong, B. M., Chin, C. A., Das, S., Tekes, A., & Thain, W. (2021, July). Enhancing engineering students' troubleshooting skills. In *2021 ASEE Virtual Annual Conference Content Access*.

Gargac, J. (2024, June). Failing Forward: A Mastery-based Learning Approach in a Theory of Machine Kinematics and Dynamics Course. In *2024 ASEE Annual Conference & Exposition*.

Klamar, A., Horvath, D., Frese, M., & Keith, N. (2024). Different approaches to learning from errors: Comparing the effectiveness of high reliability and error management approaches. *Safety Science*, *177*, 106578.

Koretsky, M. D., Nefcy, E. J., Nolen, S. B., & Champagne, A. B. (2023). Connected epistemic practices in laboratory-based engineering design projects for large-course instruction. *Science Education*, *107*(2), 510-550.

Mehraban, H., & Hu, J. (2024, June). Board 293: How to Teach Debugging? The Next Million-Dollar Question in Microelectronics Education. ASEE Annual Conferences & Exposition.

Nachtigall, V., Serova, K., & Rummel, N. (2020). When failure fails to be productive: Probing the effectiveness of productive failure for learning beyond STEM domains. *Instructional Science*, *48*(6), 651-697.

Narayanan, S., Sarin, P., Pawar, N., & Murthy, S. (2023). Teaching research skills for experimental physics in an undergraduate electronics lab. *Physical Review Physics Education Research*, *19*(2), 020103.

Phillips, A. M., Sundstrom, M., Wu, D. G., & Holmes, N. G. (2021). Not engaging with problems in the lab: Students' navigation of conflicting data and models. *Physical Review Physics Education Research*, *17*(2), 020112.

Sinha, T., & Kapur, M. (2021). When problem solving followed by instruction works: Evidence for productive failure. *Review of Educational Research*, *91*(5), 761-798.