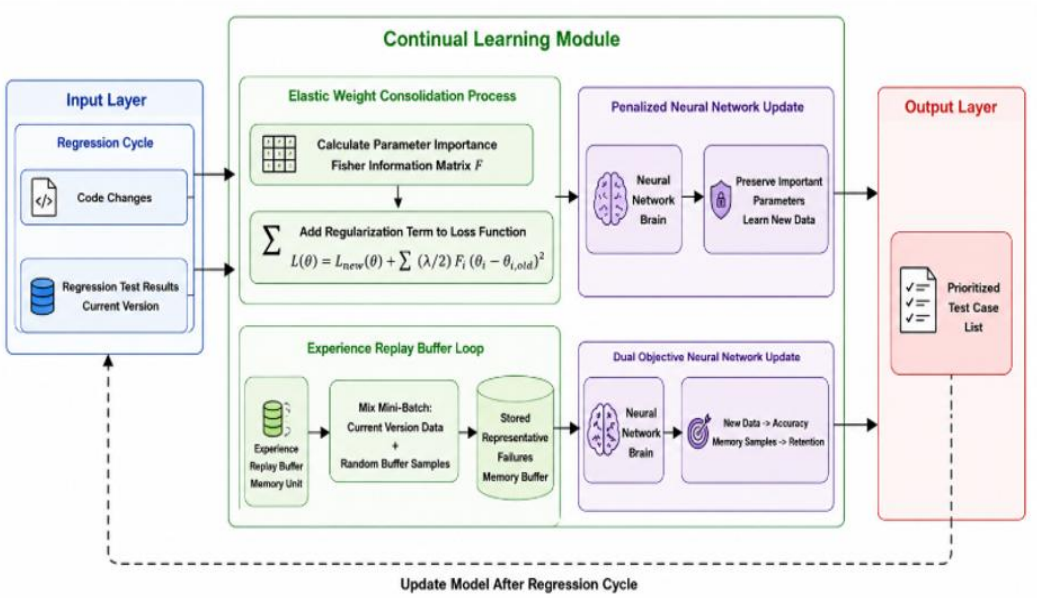
Masters Thesis Information Showcase
Spring/Fall Semester-2026
Year 2026

	Department: Software Engineering Programme: ME Software Engineering Specialization: Software Engineering
1	**Title of the Thesis** Test case prioritization using continual learning
2	**Abstract** (min 300 to 500 words) Software testing phase is very important in software development lifecycle because for a successful and reliable product it is important to ensure that the quality of the product is not compromised. There are different levels of testing such as unit, integration, system, and user acceptance testing but, on every level, the basic thing is to prioritize the test cases that how prior test case will going to be tested to get the desired result. Regression testing plays a critical role in ensuring software reliability as systems evolve so it ensures that recent code changes and bug fixes haven't unintentionally broken previously working features or introduced new defects. Hence, the efficient model is developed in this research which sets the priority of test cases to get early failure in regression of whole version. This thesis evaluates the techniques of continual learning (CL) to preserved past knowledge of test cases to use in test case prioritization in every regression of new version without need of retraining the model. Two continual learning techniques, Elastic Weight Consolidation (EWC) and Experience Replay (ER), are used as a unified, dual-defense framework and compared against traditional coverage-based and history-based approaches. Experimental evaluation is conducted using the Westermo regression testing dataset and performance is measured using the Average Percentage of Faults Detected (APFD) metric, Backward Transfer (BWT) to measure performance retention, Recall at Top-k tracks percentage of total fault in initial tests, Computational Overhead to calculate execution time. The proposed framework achieved a high, stable APFD score of 0.89 and a strong fault recall rate of 92% within the Top-20% of the test suite. Moreover, the framework's empirical results when trained with terminal Backward Transfer (-0.0125) and a flat execution time profile (13 seconds per cycle), indicate its capability to overcome catastrophic forgetting and decrease computational burden from the traditional retraining procedure. The results show promising progress with continual learning in evolving software systems. In future this work can be enhanced by using a larger data set and more software versions to evaluate the proposed approach and look at long-term learning more completely.
3	**Impact on SDG#___"** (min 500 words) (Please mention the impact of the relevant attained SDG) The intersection of software engineering and artificial intelligence holds immense potential for driving global sustainability. This thesis directly aligns with the United Nations Sustainable Development Goals (SDGs), specifically contributing to SDG 9 (Industry, Innovation, and Infrastructure), SDG 12 (Responsible Consumption and Production), and SDG 7 (Affordable and Clean Energy). By optimizing Test Case Prioritization (TCP) through a Continual Learning (CL) framework, this research transforms software testing

	from a resource-heavy bottleneck into an efficient, sustainable process applicable across diverse software domains. **Driving Digital Infrastructure via SDG 9** Software serves as the foundational infrastructure for modern global industries. This research significantly advances SDG 9 by upgrading technological capabilities and software testing pipelines to make them resilient and clean. Traditional testing regimes suffer from extreme inefficiencies when managing regression suites. By utilizing the Recall@Top-K and APFD metrics, this framework ensures that critical faults are detected early in the test cycle. Maximizing APFD means software bugs are caught with minimal test execution, accelerating continuous integration and continuous deployment (CI/CD) pipelines. This high-precision filtering fosters robust technological innovation, allowing global industries to deploy reliable software faster while minimizing industrial downtime caused by software failures. **Enabling Responsible Computing via SDG 12** AI-driven testing often creates an invisible environmental tax due to the resource-intensive nature of retraining models from scratch. This thesis addresses SDG 12 by achieving sustainable management and efficient use of computational resources. The integration of Continual Learning relies heavily on optimizing Backward Transfer (BWT), which measures the model's ability to retain historical knowledge without suffering from catastrophic forgetting. By maintaining a stable BWT, the proposed model eliminates the wasteful cycle of complete model retraining upon receiving new data. Instead, it builds incrementally on past knowledge. This paradigm shift directly reduces the hardware wear, memory overhead, and digital waste typically generated in data centers, pioneering a more responsible approach to life-cycle software maintenance. **Mitigating the Carbon Footprint via SDG 7** Data centers are among the fastest-growing consumers of electricity globally. This research advances SDG 7 by doubling the global rate of energy efficiency within software verification pipelines. The direct optimization of Execution Time achieved by this TCP framework translates immediately into reduced kilowatt-hour (kWh) consumption on cloud computing servers. Rather than running exhaustive, hours-long test suites that strain power grids, the prioritized ordering allows teams to halt testing once the Recall@Top-K threshold guarantees sufficient fault coverage. Shortening the active runtime of central and graphics processing units (CPUs/GPUs) directly curtails the carbon dioxide emissions associated with cloud-scale testing. Consequently, this methodology offers a concrete pathway toward green computing, proving that algorithmic efficiency can actively mitigate the carbon footprint of the digital ecosystem.
4	**Scholar Name** (along with contact details) Name: Syed Faraz Ali Zaidi Email: Zaidi.pg4201205@cloud.neduet.edu.pk

5	**Supervisor & Co-supervisor Name** (along with contact details) Name: Dr. Mustafa Latif
6	**Graphical Abstract** 

In this architecture, two continual learning strategies, Elastic Weight Consolidation and Experience Replay, are integrated in a single regression testing process. The process begins with the Input Layer, where regression cycle information, and current regression test results are provided to the continual learning module. These inputs are related to the behaviour of the software system, as it changes over several regression cycles. The central element is the Continual Learning Module, parallel learning path. Elastic Weight Consolidation (EWC) and (ER) which helps to prioritized test case list.