Just-in-Time Software Defect Prediction using Unsupervised Methods for Detecting and Handling Concept Drift

Aditya Pote (2020IMT069)

Supervisor: Dr. Santosh Singh Rathore
Co-Supervisor: Prof. Joydip Dhar

ABV-Indian Institute of Information Technology and Management Gwalior

Outline

1. Introduction
2. Unsupervised JIT Software Defect Prediction
3. Concept Drift in Software Defect Prediction
4. Literature Survey on JIT-SDP
5. Literature Survey on Concept Drift Detection
6. Objective
7. Dataset Overview
8. Proposed Methodology
9. Progress Summary
10. References

Introduction

- **Software Defect Prediction (SDP):** SDP predicts which parts of the software code are most likely to contain defects. By analyzing code complexity, change frequency, and other metrics, SDP helps developers focus their efforts on areas that are at higher risk for bugs, improving software quality.
- **Just-in-Time Software Defect Prediction (JIT-SDP):** JIT-SDP predicts defects immediately after code changes are made, analyzing each commit in real time. This allows developers to quickly identify and fix problems, reducing the impact of defects later in the development cycle.
- **Challenge of Concept Drift:** One key challenge in JIT-SDP is concept drift, where changes in software development practices or code over time degrade the performance of predictive models.

Unsupervised JIT Software Defect Prediction

- **Definition:** Unsupervised JIT-SDP predicts defect-prone code modules without the need for labeled data. Instead, it analyzes software metrics to group and identify potential defects.
- **Key Metrics:** These models rely on software metrics such as code complexity, churn (frequency of changes), and dependency analysis to make predictions about defects.
- **Advantages:** Unsupervised JIT-SDP models can be applied in scenarios where labeled data is scarce or unavailable, making them useful in real-world software projects.

Concept Drift in Software Defect Prediction

- **Definition:** Concept drift occurs when the underlying data distribution changes over time, leading to a decrease in the model's performance.
- **Impact on SDP Models:** In software defect prediction, concept drift can be caused by changes in development practices, tools, or frameworks, leading to inaccurate predictions over time.
- **Solution:** To maintain accuracy, models must adapt to the evolving characteristics of the data, making concept drift detection and adaptation crucial for long-term model reliability.

Literature Survey on JIT-SDP

Paper	Technique	Dataset	Results/Findings
Zhao et al. (2023) [1]	Machine learning for JIT-SDP	Comprehensive meta-analysis of 67 JIT-SDP studies	Identified that JIT-SDP performs best with high change defect ratios;Class imbalance, Concept drift, latency issue[1].
Cabral et al. (2022) [2]	Class imbalance evolution learning	10 GitHub projects	Showed that JIT-SDP suffers from class imbalance evolution, Method is Not Real Time [2].
Ni et al. (2022) [3]	Semantic + expert features (JIT-Fine model)	JIT-Defects4J dataset	JIT-Fine outperformed baselines by 10%-37% in defect prediction and Computationally complex. [3].
Kumar et al. (2022) [4]	Threshold clustering labeling (TCLP) for unsupervised defect prediction	28 datasets from five repositories of SOFTLAB, and AEEEM	Outperformed state-of-the-art methods with significant improvements in accuracy, F-measure, and MCC [4].

Literature Survey on Concept Drift Detection

Paper	Technique	Dataset	Results/Findings
Salazar et al. (2024) [5]	Group-specific distributed concept drift	MNIST, FEMNIST datasets	Improved fairness and accuracy under group-specific distributed drift in federated learning [5].
Wan et al. (2024) [6]	Maximum Concept Discrepancy (MCD)	11 datasets	Outperformed baselines in drift detection, but had Scalability Problem [6].
Wang et al. (2024) [7]	QuadCDD Framework	Various data streams	Only Analyzed drift start, end, severity, and type, enhancing model stability in evolving data [7].
Greco et al. (2024) [8]	Unsupervised drift detection with Fréchet distance	Real-time data streams	Used deep learning representations to detect and adapt to concept drift in dynamic environments [8].

Objective

- **Develop an unsupervised software defect prediction model:**
  - Create a predictive model that identifies defect-prone code modules without the need for labeled data.
- **Handle Concept Drift in Real-Time:**
  - Integrate DriftLens to continuously monitor and detect concept drift in software metrics, ensuring the model adapts over time.
- **Improve Just-In-Time Defect Prediction:**
  - Enhance JIT-SDP by making real-time predictions after code commits, helping developers address defects early in the development process.

Dataset Overview

- **Dataset Name:** ApacheJIT
- **Commits:** 106,674 total commits from 14 Apache projects, including 28,239 bug-inducing commits and 78,435 clean commits.
- **Metrics:** Includes various change metrics like:
    - Lines added (la), lines deleted (ld)
    - Files and directories touched (nf, nd)
    - Subsystems involved (ns), change entropy (ent)
    - Developer experience metrics: ndev, aexp, arexp, asexp
- **Time Period:** Covers commits from 2003 to 2019 across multiple versions.
- **Use Case:** Designed for Just-In-Time defect prediction, particularly suited for deep learning models.

Proposed Methodology

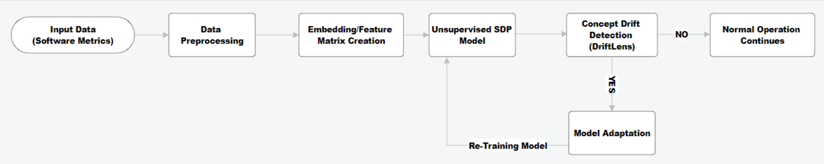
Input Data (Software Metrics)
Data Preprocessing
Embedding/Feature Matrix Creation
Unsupervised SDP Model
Concept Drift Detection (DriftLens)
NO
Normal Operation Continues
YES
Model Adaptation
Re-Training Model

Figure 1: Workflow of the Proposed Methodology

Workflow Explanation:

- The input data (software metrics) undergoes preprocessing and is transformed into a feature matrix.
- The unsupervised SDP model processes the feature matrix to predict defect-prone modules.
- DriftLens monitors for concept drift using deep learning representations. If drift is detected, the model undergoes retraining and adaptation.
- If no drift is detected, the system continues normal operation without intervention.

Progress Summary

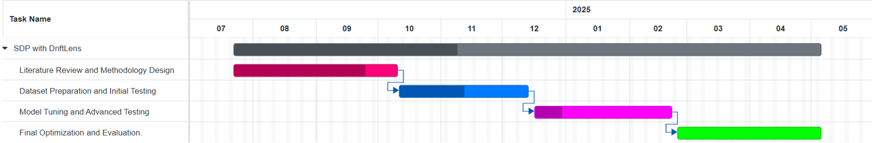

Figure 2: Project Progress and Timeline

Progress Overview:

- **Literature Review and Methodology Design:** Completed during Q3 2024, covering in-depth research on concept drift, unsupervised SDP, and JIT-SDP.
- **Dataset Preparation and Initial Testing:** In progress, with initial testing on the ApacheJIT dataset for unsupervised defect prediction.
- **Model Tuning and Advanced Testing:** Planned for Q4 2024, focusing on tuning the DriftLens framework for concept drift detection.
- **Final Optimization and Evaluation:** Set for Q1-Q2 2025, where the final model will be optimized and evaluated.

References I

[1] Y. Zhao, K. Damevski, and H. Chen, “A systematic survey of just-in-time software defect prediction,” *ACM Comput. Surv.*, vol. 55, Feb. 2023.

[2] G. G. Cabral, L. L. Minku, E. Shihab, and S. Mujahid, “Class imbalance evolution and verification latency in just-in-time software defect prediction,” in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pp. 666–676, 2019.

[3] C. Ni, W. Wang, K. Yang, X. Xia, K. Liu, and D. Lo, “The best of both worlds: integrating semantic features with expert features for defect prediction and localization,” in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2022, (New York, NY, USA), p. 672–683, Association for Computing Machinery, 2022.

[4] R. Kumar, A. Chaturvedi, and L. Kailasam, “An unsupervised software fault prediction approach using threshold derivation,” *IEEE Transactions on Reliability*, vol. 71, no. 2, pp. 911–932, 2022.

[5] T. Salazar, J. Gama, H. Araújo, and P. H. Abreu, “Unveiling group-specific distributed concept drift: A fairness imperative in federated learning,” 2024.

[6] K. Wan, Y. Liang, and S. Yoon, “Online drift detection with maximum concept discrepancy,” in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD ’24, (New York, NY, USA), p. 2924–2935, Association for Computing Machinery, 2024.

[7] P. Wang, H. Yu, N. Jin, D. Davies, and W. L. Woo, “Quadcdd: A quadruple-based approach for understanding concept drift in data streams,” *Expert Systems with Applications*, vol. 238, p. 122114, 2024.

[8] S. Greco, B. Vacchetti, D. Apiletti, and T. Cerquitelli, “Unsupervised concept drift detection from deep learning representations in real-time,” 06 2024.

Thank you!

2020IMT-069