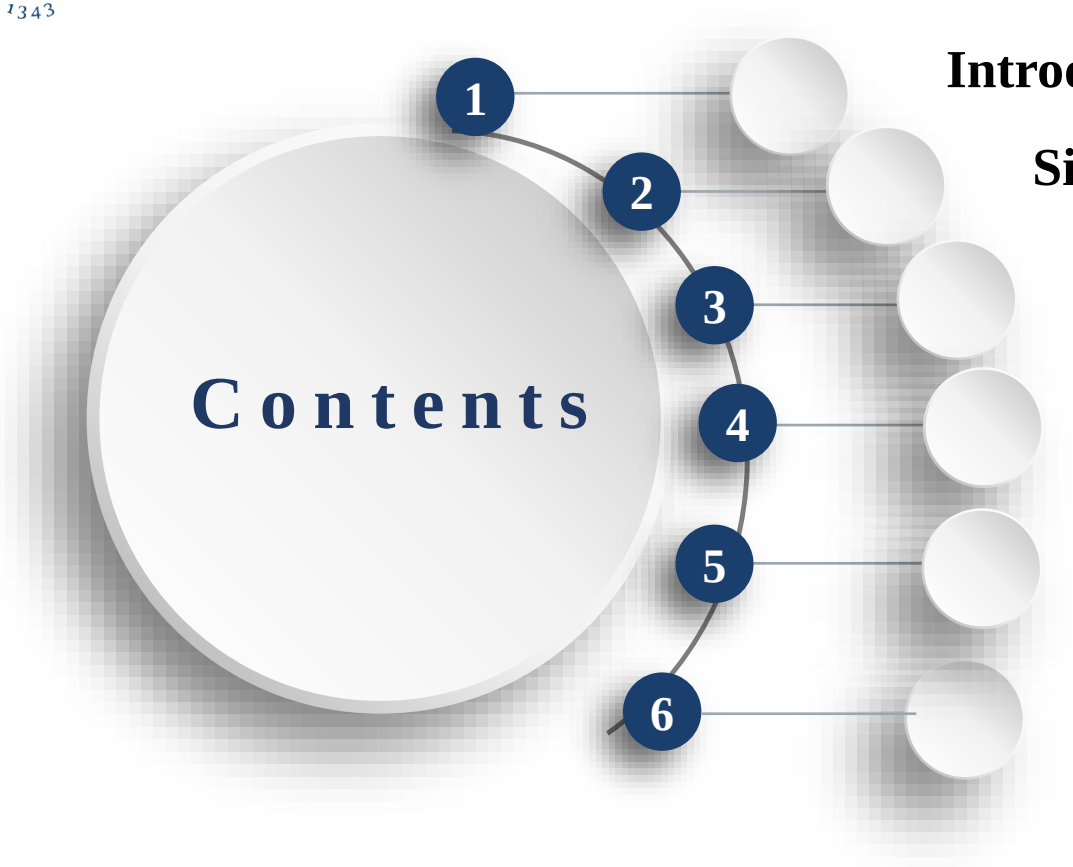
Cross-Project Defect Prediction: Scalable and Interpretable Domain Adaptation Approaches

Khadija Javed

Department of Computer Science

University of Pisa

Mauriana Pesaresi Seminars, 2025/04/11



01

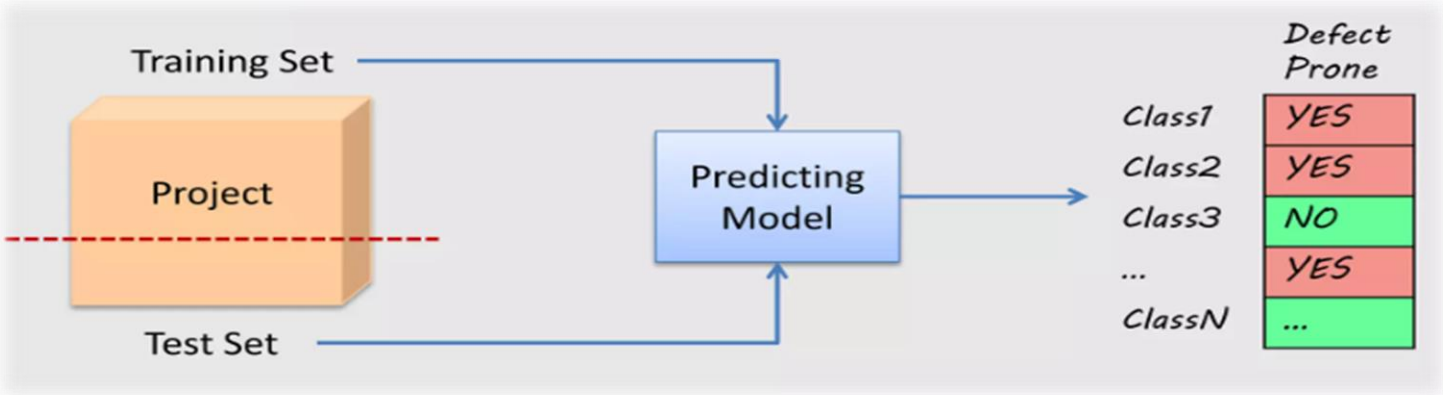
Introduction

Software Defect Prediction

- Learns a prediction model from historic data
- Predicts defect for same project
- Hundreds of prediction model exists
- Models work fairly well with precision and recall up to 80%.

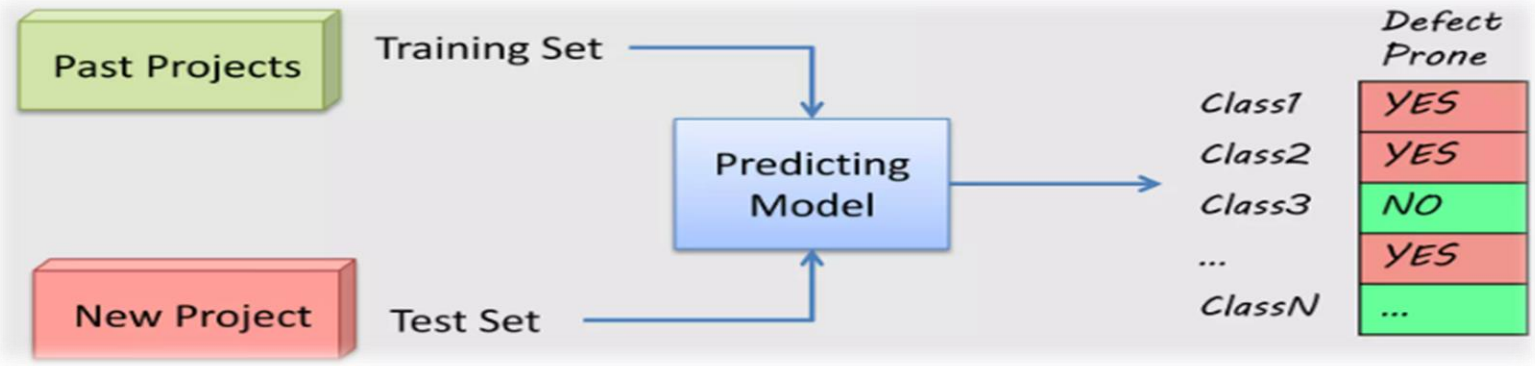
Predictor	Precision	Recall
Pre-Release Bugs	73.80%	62.90%
Test Coverage	83.80%	54.40%
Dependencies	74.40%	69.90%
Code Complexity	79.30%	66.00%
Code Churn	78.60%	79.90%
Org. Structure	86.20%	84.00%

From: N. Nagappan, B. Murphy, and V. Basili. The influence of organizational structure on software quality. ICSE 2008.



Why Cross-Project Defect Prediction?

- Some projects do have not enough data to train prediction models or the data is of poor quality
- New projects do have no data yet
- Can such projects use models from other projects?
- Cross-project defect prediction (CPDP) predict defects in a target project domain by leveraging information from different source project domains.



02

Significance and research value

Significance and Research Value

Significance and Research Value

- Enables defect prediction in **data-scarce projects**
- **Reduces cost** by identifying defects pre-deployment
- **Improves software quality** across domains
- Leverages historical data from other projects
- Enhances prediction with **transfer learning techniques**
- Faces **challenges** due to complex structure, data disparity, and class imbalance

03

Challenges in Cross-Project Defect Prediction (CPDP)

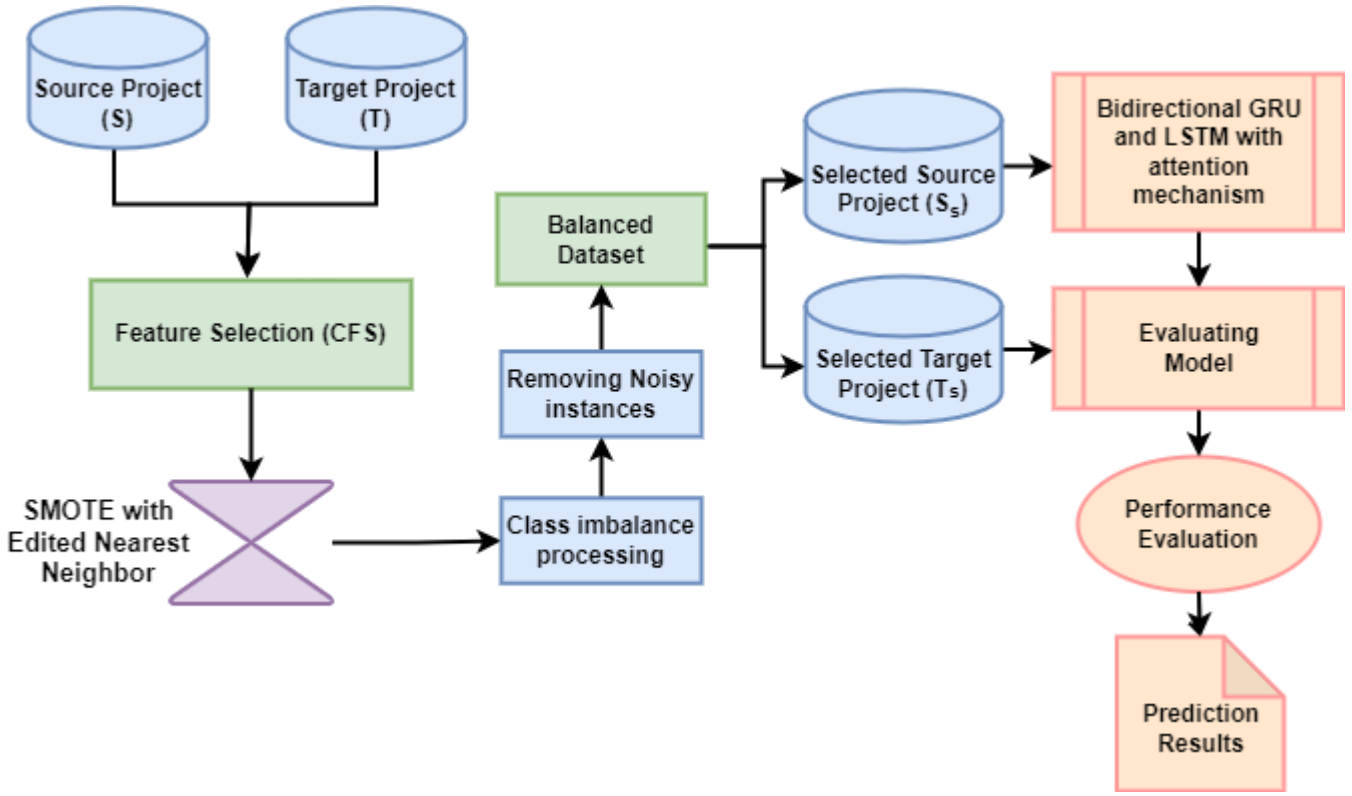
Challenges in Cross-Project Defect Prediction (CPDP)

- Models Handling **domain differences** between projects.
- Ensuring **data quality** and consistency across diverse sources.
- Overcoming the scarcity of **labeled data** in target projects.
- Balancing model **generalization** with prediction **accuracy**.

05

Methodology

Methodology

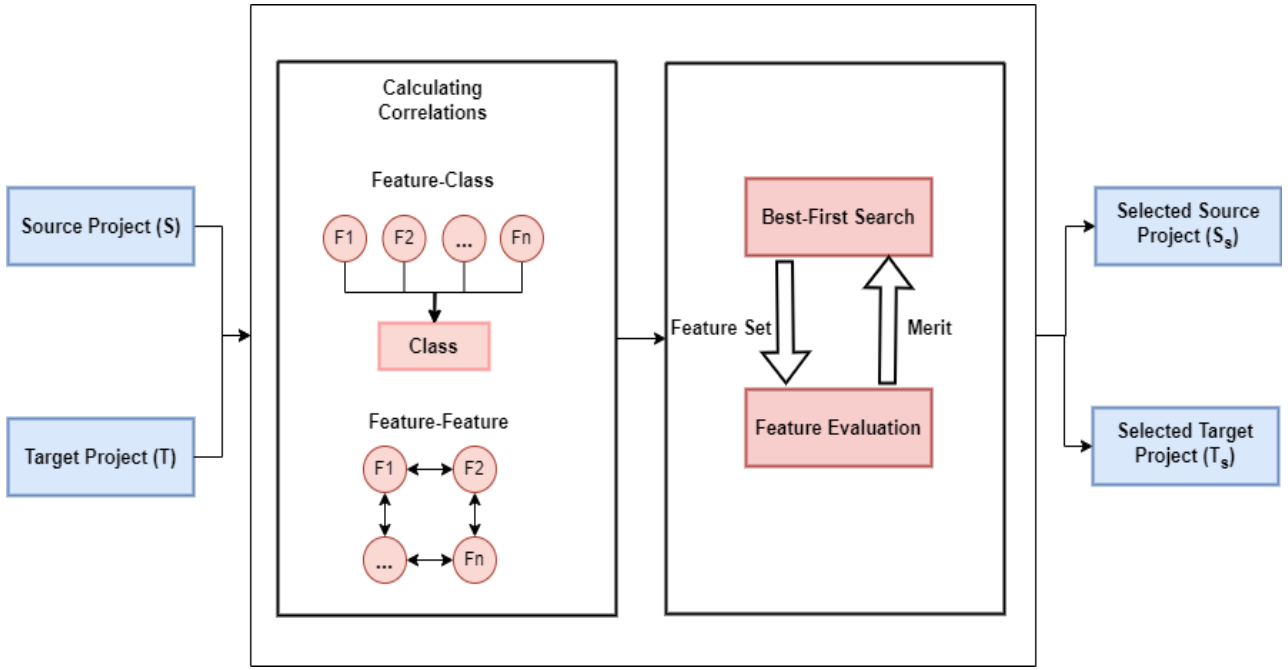


Framework of Proposed Methodology for Cross-Project Defect Prediction.

Methodology

Model Building: Correlation-based Feature Selection (CFS)

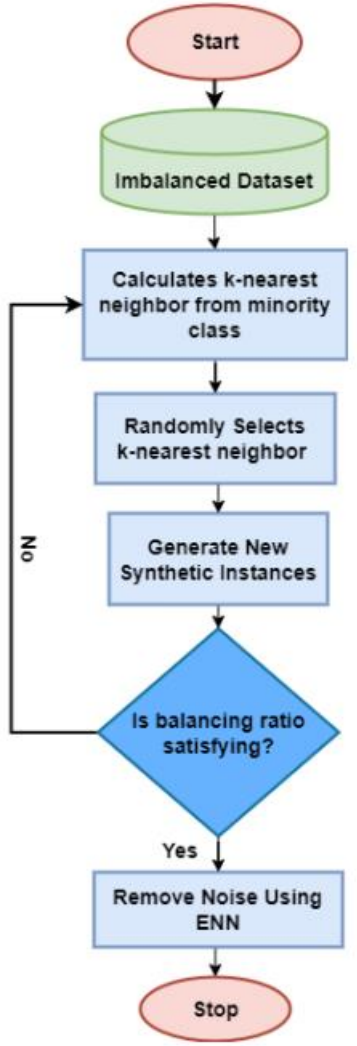
- Correlation-based Feature Selection (CFS) selects features based on predictive capacity and redundancy.
- Uses best-first search to find features with high correlation to target and low internal correlation



Methodology

Model Building: SMOTE-ENN

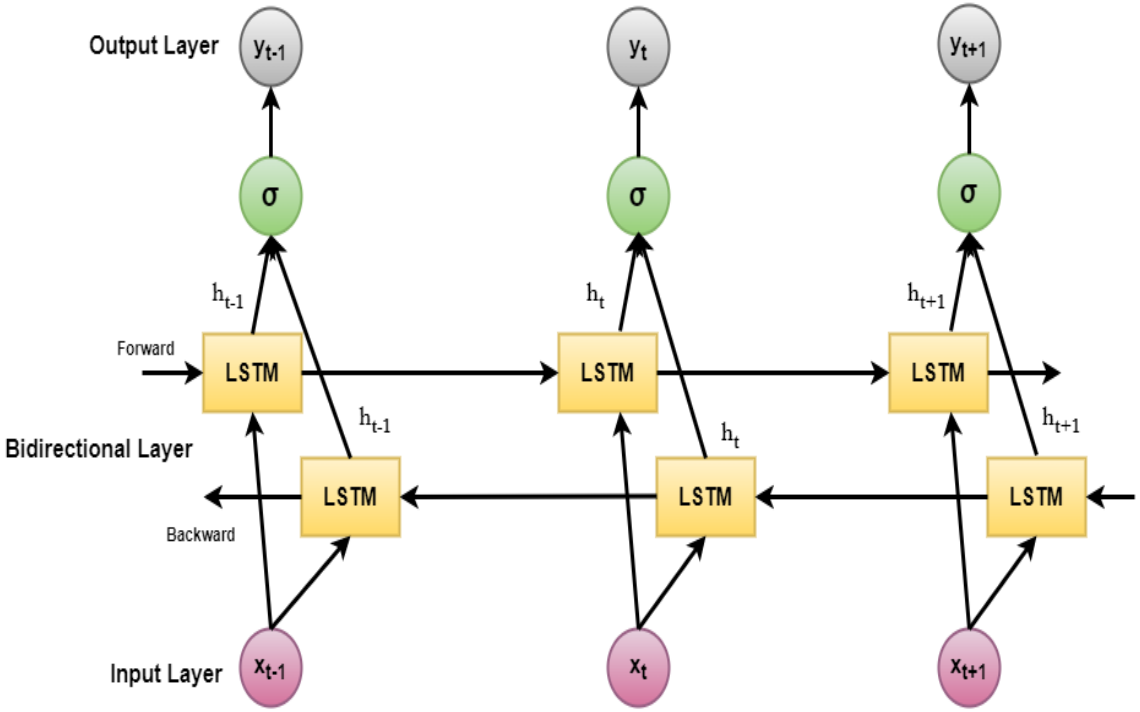
- SMOTE: **Oversample minority class** by creating synthetic examples with **k-nearest neighbors**.
- Repeat until target oversampling is achieved.
- ENN: Determine k-nearest neighbors and assign majority class.
- Delete instances with differing classes between observation and neighbors and repeat until class balance is reached



Methodology

Model Building: Bidirectional LSTM

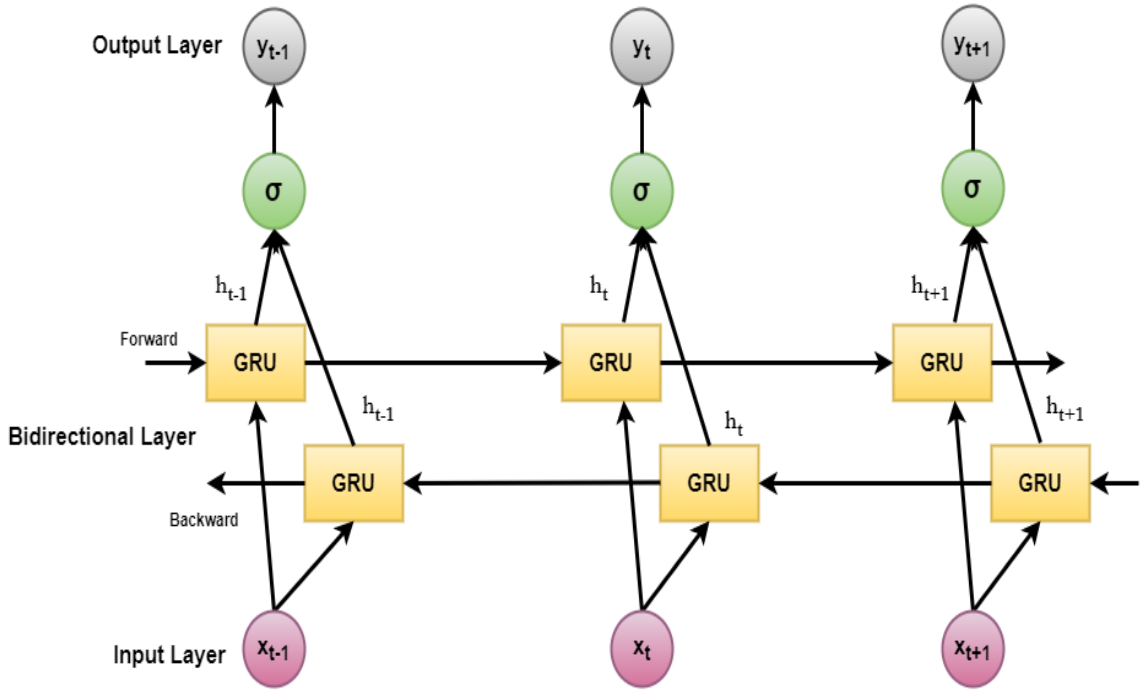
- Bidirectional LSTM handles context dependencies with two layers
- One layer processes the sequence forward, the other backward
- Final output is the concatenation of both layers' hidden states



Methodology

Model Building: Bidirectional GRU

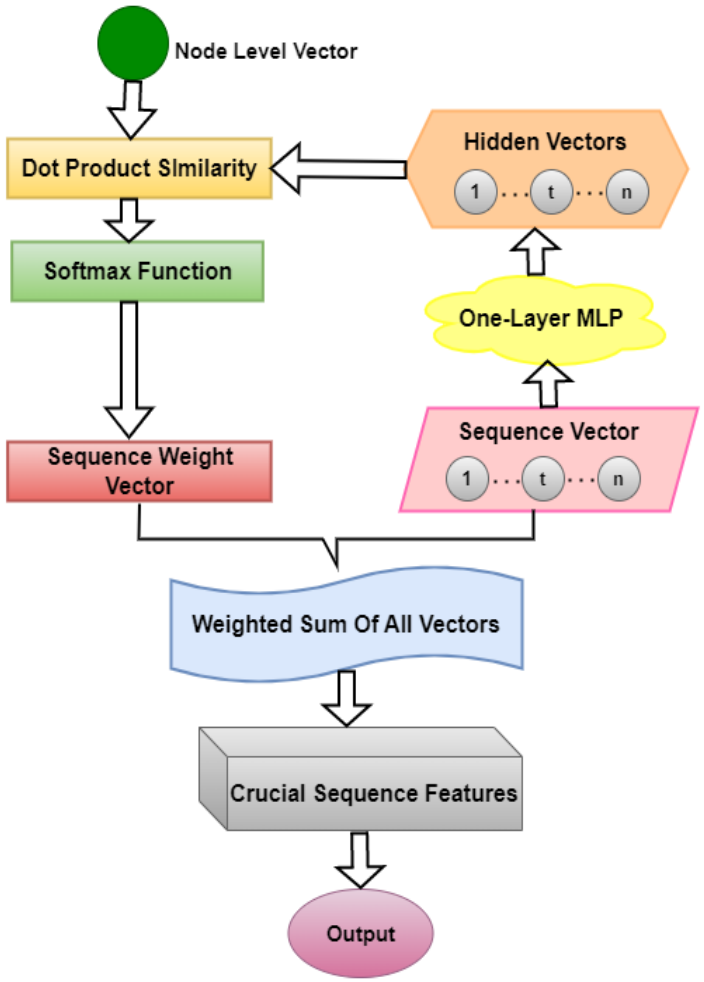
- Bi-GRU uses two unidirectional GRUs, one forward and one backward
- Combines past and future information to impact current states



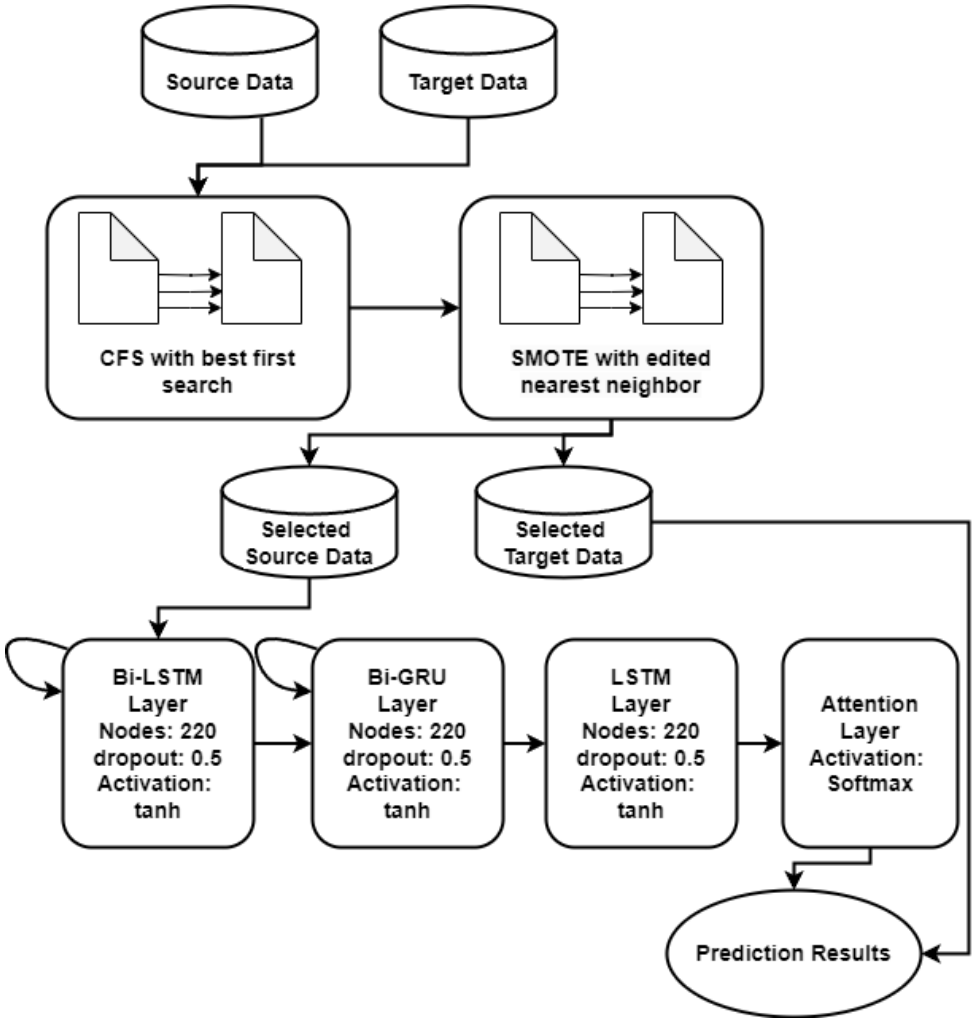
Methodology

Model Building: Attention mechanism

- Attention mechanism weights and focuses on important nodes in a sequence
- Aggregates meaningful nodes to build a sequence vector



Methodology



Architecture of Proposed Model

06

Results

Results

Description of Datasets

Dataset	Project	Number of instances	Defective instances%
AEEEM	EQ	325	39.692
	JDT	997	20.662
	LC	399	16.040
	ML	1862	13.158
	PDE	1492	14.008
PROMISE	Ivy2.0	352	11.36
	Poi3.0	442	64.09
	Xerces1.4	508	76.81
	Synapse1.2	256	33.63
	Xalan2.6	875	53.13

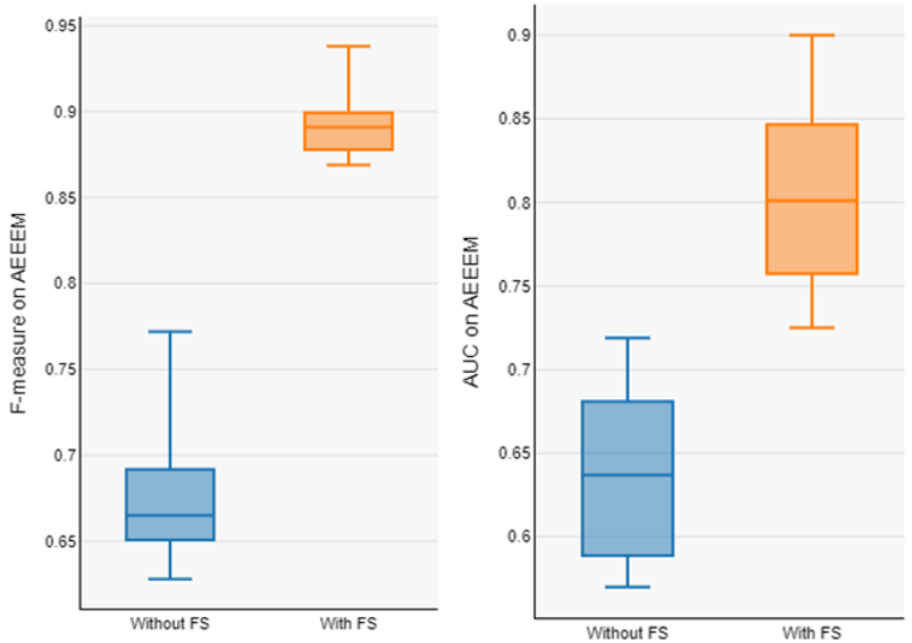
Results

F1- measure and AUC Analysis with and without Feature Selection on AEEM

Source	Target	F1-measure		AUC	
		Without FS	With FS	Without FS	With FS
EQ	JDT	0.737	0.900	0.681	0.875
EQ	LC	0.742	0.917	0.590	0.750
EQ	ML	0.677	0.895	0.589	0.731
EQ	PDE	0.659	0.879	0.610	0.771
JDT	EQ	0.668	0.892	0.719	0.900
JDT	LC	0.652	0.879	0.670	0.810
JDT	ML	0.661	0.887	0.680	0.762
JDT	PDE	0.649	0.875	0.586	0.761
LC	EQ	0.650	0.878	0.570	0.846
LC	JDT	0.760	0.923	0.674	0.848
LC	ML	0.643	0.869	0.571	0.738
LC	PDE	0.651	0.871	0.592	0.760
ML	EQ	0.665	0.893	0.688	0.848
ML	JDT	0.669	0.899	0.660	0.823
ML	LC	0.772	0.938	0.688	0.835
ML	PDE	0.644	0.873	0.580	0.750
PDE	EQ	0.733	0.908	0.699	0.883
PDE	JDT	0.665	0.891	0.629	0.791
PDE	LC	0.654	0.884	0.681	0.820
PDE	ML	0.628	0.877	0.588	0.725
Average		0.678	0.891	0.637	0.801

Results

(a) Boxplot Analysis of F1- measure with and without Feature Selection on AEEEM
(b) Boxplot Analysis of AUC with and without Feature Selection on AEEEM



(a) (b)

06

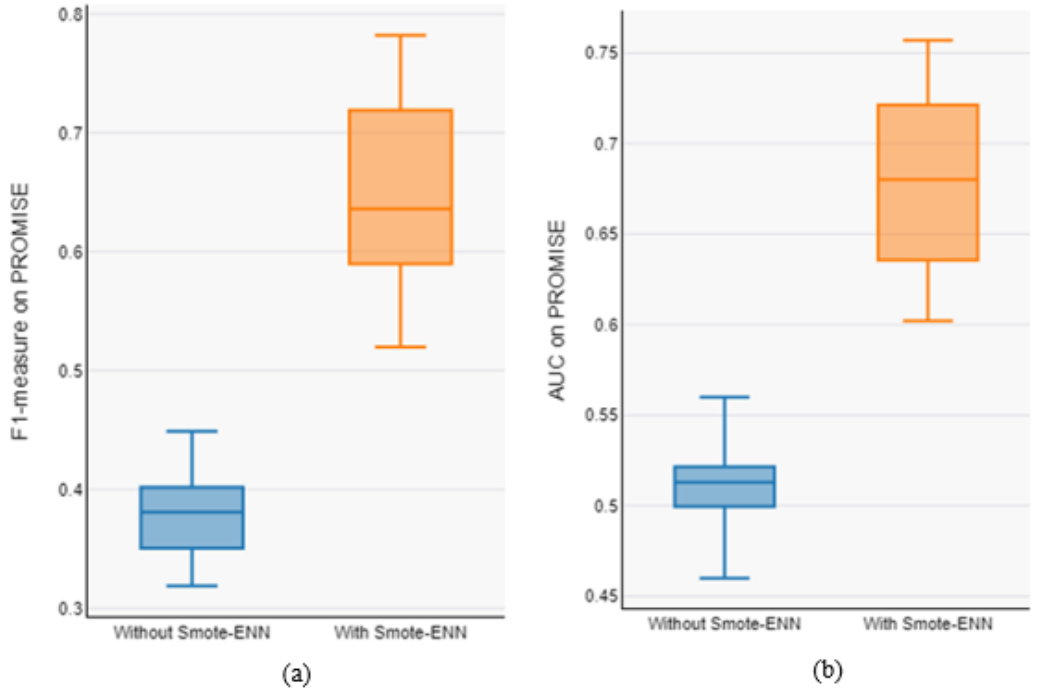
Results

F1- measure and AUC Analysis with and without Data Balancing Method on AEEEM.

Source	Target	F1-measure		AUC	
		Without SMOTE-ENN	With SMOTE-ENN	Without SMOTE-ENN	With SMOTE-ENN
EQ	JDT	0.580	0.900	0.447	0.875
EQ	LC	0.588	0.917	0.416	0.750
EQ	ML	0.563	0.895	0.413	0.731
EQ	PDE	0.457	0.879	0.412	0.771
JDT	EQ	0.578	0.892	0.602	0.900
JDT	LC	0.494	0.879	0.510	0.810
JDT	ML	0.489	0.887	0.427	0.762
JDT	PDE	0.483	0.875	0.420	0.761
LC	EQ	0.488	0.878	0.510	0.846
LC	JDT	0.491	0.923	0.515	0.848
LC	ML	0.481	0.869	0.422	0.738
LC	PDE	0.478	0.871	0.429	0.760
ML	EQ	0.480	0.893	0.513	0.848
ML	JDT	0.466	0.899	0.509	0.823
ML	LC	0.499	0.938	0.507	0.835
ML	PDE	0.487	0.873	0.418	0.750
PDE	EQ	0.489	0.908	0.514	0.883
PDE	JDT	0.497	0.891	0.412	0.791
PDE	LC	0.480	0.884	0.519	0.820
PDE	ML	0.495	0.877	0.415	0.725
Average		0.503	0.891	0.466	0.801

Results

(a) Boxplot Analysis of F1- measure with and without Smote-Enn on PROMISE
(b) Boxplot Analysis of AUC with and without Smote-Enn on PROMISE



Experimental Results and Evaluations

F1-measure Analysis of The Proposed Approach and Baseline Methods on PROMISE

Source	Target	TPTL	DA-KTSVMO	GB-CPDP	Ours
synapse_1.2	poi-2.5	0.462	0.533	0.631	0.651
synapse_1.2	xerces-1.2	0.433	0.542	0.466	0.602
camel-1.4	ant-1.6	0.575	0.463	0.416	0.656
camel-1.4	jedit_4.1	0.396	0.402	0.356	0.636
xerces-1.3	poi-2.5	0.349	0.537	0.544	0.595
xerces-1.3	synapse_1.1	0.536	0.329	0.469	0.588
xerces-1.2	xalan-2.5	0.447	0.462	0.383	0.571
lucene_2.2	xalan-2.5	0.506	0.438	0.502	0.612
synapse_1.1	poi-3.0	0.342	0.566	0.537	0.602
ant-1.6	poi-3.0	0.353	0.315	0.384	0.520
camel-1.4	ant-1.6	0.556	0.511	0.652	0.782
lucene_2.2	ant-1.6	0.377	0.539	0.669	0.772
log4j-1.1	ant-1.6	0.595	0.585	0.676	0.745
log4j-1.1	lucene_2.0	0.478	0.576	0.622	0.733
lucene_2.0	log4j-1.1	0.419	0.561	0.489	0.742
lucene_2.0	xalan-2.5	0.510	0.510	0.514	0.546
jedit_4.1	camel-1.4	0.447	0.502	0.501	0.678
jedit_4.1	xalan-2.4	0.332	0.386	0.443	0.552
Average		0.451	0.487	0.514	0.643

Experimental Results and Evaluations

AUC Analysis of The Proposed Approach and Baseline Methods on PROMISE

Source	Target	TPTL	DA-KTSVMO	GB-CPDP	Ours
synapse_1.2	poi-2.5	0.485	0.498	0.593	0.674
synapse_1.2	xerces-1.2	0.485	0.563	0.681	0.712
camel-1.4	ant-1.6	0.541	0.655	0.532	0.669
camel-1.4	jedit_4.1	0.329	0.441	0.466	0.612
xerces-1.3	poi-2.5	0.588	0.477	0.568	0.633
xerces-1.3	synapse_1.1	0.488	0.468	0.502	0.602
xerces-1.2	xalan-2.5	0.471	0.437	0.696	0.722
lucene_2.2	xalan-2.5	0.621	0.702	0.568	0.733
synapse_1.1	poi-3.0	0.493	0.510	0.571	0.630
ant-1.6	poi-3.0	0.518	0.383	0.572	0.619
camel-1.4	ant-1.6	0.603	0.642	0.661	0.713
lucene_2.2	ant-1.6	0.411	0.570	0.658	0.729
log4j-1.1	ant-1.6	0.631	0.509	0.682	0.733
log4j-1.1	lucene_2.0	0.529	0.621	0.613	0.757
lucene_2.0	log4j-1.1	0.546	0.571	0.647	0.719
lucene_2.0	xalan-2.5	0.632	0.604	0.594	0.669
jedit_4.1	camel-1.4	0.267	0.355	0.556	0.644
jedit_4.1	xalan-2.4	0.425	0.563	0.669	0.680
Average		0.504	0.532	0.602	0.680

07

Conclusion& Future Prospects

Conclusion & Future Prospects

- Explored **domain adaptation techniques**, leveraged to overcome different data distribution and class imbalance problem in CPDP and a deep learning model that combines bi-directional LSTM and GRU with attention mechanism for Cross-project defect prediction model.
- Exploring **hybrid models** combining traditional machine learning with deep learning approaches, Ahmed et al. and **model averaging** in cross-project defect prediction can improve prediction performance of model, Li et al.

07

References

- Javed, K.; Shengbing, R.; Asim, M.; Wani, M.A. Cross-Project Defect Prediction Based on Domain Adaptation and LSTM Optimization. Algorithms 2024, 17, 175. https://doi.org/10.3390/a17050175
- Li, T., Wang, Z. & Shi, P. Within-project and cross-project defect prediction based on model averaging. *Sci Rep* **15**, 6390 (2025). https://doi.org/10.1038/s41598-025-90832-4
- Ahmed Abdu, Zhengjun Zhai, Hakim A. Abdo, Sungon Lee, Mohammed A. Al-masni, Yeong Hyeon Gu, Redhwan Algabri, Cross-project software defect prediction based on the reduction and hybridization of software metrics, Alexandria Engineering Journal, Volume 112, 2025, Pages 161-176, ISSN 1110-0168, https://doi.org/10.1016/j.aej.2024.10.034.
- Zhang, W., Zhao, J., Qin, G. et al. Cross-project defect prediction based on autoencoder with dynamic adversarial adaptation. Appl Intell 55, 324 (2025). https://doi.org/10.1007/s10489-024-06087-5
- Berahmand K, Daneshfar F, Salehi ES et al (2024) Autoencoders and their applications in machine learning: a survey. Art Intell Rev 57(2):28. https://doi.org/10.1007/s10462-023-10662-6
- Li Z, Zhang H, Jing X, Xie J, Guo M, Ren J (2023) DSSDPP: data selection and sampling based domain programming predictor for cross-project defect prediction. IEEE Trans Software Eng 49:1941–1963. https://doi.org/10.1109/TSE.2022.3204589
- Liu C, Yang D, Xia X, Yan M, Zhang X (2019) A two-phase transfer learning model for cross-project defect prediction. Inf SoftwTechnol 107:125–136. https://doi.org/10.1016/j.infsof.2018.11. 005
- Tong H, Liu B, Wang S, Li Q (2019) Transfer-learning oriented class imbalance learning for cross-project defect prediction, ArXiv, pp 1–38. https://doi.org/10.48550/arXiv.1901.08429

Thank you

Q&A Time