The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 24, No. 5, pp.117-122, Oct. 31, 2024. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2024.24.5.117
JIIBC 2024-5-17

전이학습 기법들을 이용한 교차 프로젝트 결함 예측

Cross-Project Defect Prediction using Transfer Learning Methods

홍의석*

Euyseok Hong*

요 약 매우 많은 소프트웨어 결함 예측에 관한 연구들이 수행되었으나, 학습 데이터의 부족으로 이들을 사용하기에 어려움이 있었다. 교차 프로젝트 결함 예측은 이를 해결하기 위한 기법으로 학습 데이터가 충분한 소스 프로젝트의 데이터로 학습한 모델을 타겟 프로젝트의 결함 예측에 사용하는 것이다. 학습을 하기 전에 두 프로젝트간의 데이터 분포 차이를 최소화하기 위해 전이학습의 일종인 도메인 적응 기법들을 사용한다. 본 논문은 W-BDA, MEDA를 사용한 새로운 모델들을 제작하여 TCA, BDA를 사용한 기존 모델들과 성능을 비교하였다. 평가 실험 결과 MEDA는 타 모델들에 비해 불규칙적이고 나쁜 성능을 보였지만 BDA는 TCA보다 더 나은 성능을 보였고, W-BDA는 BDA보다 약간 더 좋은 성능을 보였다.

Abstract Many studies on software defect prediction have been conducted, but it has been difficult to use them due to a lack of training data. Cross-project defect prediction is a technique to solve this problem, where a prediction model learned with sufficient training data from existing source project is used to predict defects in the target project. Before learning, domain adaptation techniques, a type of transfer learning, are used to minimize the difference in data distribution between the two projects. In this paper, we produced new prediction models using W-BDA and MEDA and compared their performance with existing models using TCA and BDA. As a result of the evaluation experiment, MEDA showed irregular and poor performance compared to other models, but BDA showed better performance than TCA, and W-BDA showed slightly better performance than BDA.

Key Words : Cross-project defect prediction, Transfer learning, Domain adaptation

Ⅰ. 서 론

대형 소프트웨어 개발 프로세스에서 문제 부분을 미리 예측하는 결함 예측 기술은 적절한 자원할당 등을 가능케하므로 최소 비용으로 최대 품질의 시스템 개발을 가능케한다. 해당 분야에 많은 연구들이 진행되었으며 대부분은 결함 여부 데이터인 훈련 데이터를 학습하는 감독형 모델들에 관한 것들이었다. 하지만 대부분 개발집

*정회원, 성신여자대학교 컴퓨터공학과
접수일자 2024년 7월 25일, 수정완료 2024년 9월 12일
게재확정일자 2024년 10월 4일

Received: 25 July, 2024 / Revised: 12 September, 2024 / Accepted: 4 October, 2024
*Corresponding Author: hes@sungshin.ac.kr
Dept. of Computer Engineering, Sungshin Women's University, Korea

단은 학습할 결함 데이터를 갖고 있지 않는다는 문제가 있다. 이를 해결하기 위해 소수의 비감독형 모델, 세미 감독형 모델들이 제안되었으며[1], CPDP(Cross-Project Defect Prediction)도 이 문제를 해결하기 위한 기법들 중 한 부류이다. CPDP란 예측 대상 프로젝트의 데이터가 부족하므로 학습 데이터가 충분한 다른 소스 프로젝트의 데이터로 학습한 모델을 예측 대상인 타켓 프로젝트에 사용하는 기법이다. 소스와 타겟 프로젝트는 유사한 성질을 지닌 것들을 사용해야하지만 근본적으로 데이터 분포가 다르므로 이들의 분포의 차이를 줄이기 위해 전이학습의 도메인 적응 기법들이 이용된다. 본 논문의 목적은 도메인 적응 기법들인 W-BDA, MEDA 기법을 사용한 모델들을 만든 후 기존에 CPDP에 사용되었던 TCP, BDA 이용 모델들과 성능을 비교해보는 것이다.

2장에서는 도메인 적응 기법들을 사용한 기존 CPDP 연구들과 새로 제작할 모델에 사용할 기법들을 살펴보고, 3장에서는 실험을 위한 모델 제작 및 파라미터 설정에 대해 설명한다. 4장에서는 실험 결과를 언급하고, 5장에서 결론을 기술한다.

II. 교차 프로젝트 결함 예측

도메인 적응 기법의 목표는 소스 도메인 D_s와 타겟 도메인 D_t의 분포 차이를 최소화하는 것이다. TCA는 도메인의 주변 분포(marginal distribution)만 고려하였고, BDA는 여기에 조건 분포(conditional distribution)까지고려한 기법이다. 이들 기법에서 사용한 두 도메인의 차이를 이론식으로 나타내면 (1)과 같이 두 주변 분포와 두 조건 분포의 차이를 적정 비율로 분배하여 합한 것이다. 균형 팩터인 μ는 두 도메인 분포가 유사할수록 1에 가깝고 그 반대일수록 0에 가까운 값이 된다. TCA는 주변 분포만을 고려하므로 μ를 0으로 한 식을 사용하고[2], BDA는 두가지 분포를 모두 고려하여 분포에 맞게 μ를 조절하여 분포를 할당한다[3].

$$D(D_s, D_t) \approx (1-\mu)D(P(x_s), P(x_t)) + \mu D(P(y_s|x_s), P(y_t|x_t)) \quad (1)$$

도메인 분포 사이의 차이 즉 거리는 MMD를 사용하였다. MMD는 RKHS(Reproducing Kernel Hilbert Space)에서 분포의 차이를 측정하는 방법이다. MMD를 적용하여 (1)을 변형한 수식이 (2)이다. n, m, C는 소스와 타겟 도메인의 샘플수, 클래스 라벨수를 나타내고 n_c, m_c는 각각 소스와 타겟 도메인에서 클래스 c인 샘플수를 나타낸다.

$$D(D_s, D_t) \approx (1-\mu)\left|\left|\frac{1}{n}\sum_{i=1}^{n}x_{s_i} - \frac{1}{m}\sum_{j=1}^{m}x_{t_j}\right|\right|_H^2 + \mu\sum_{c=1}^{C}\left|\left|\frac{1}{n_c}\sum_{x_{s_i}\in D_s^{(c)}}x_{s_i} - \frac{1}{m_c}\sum_{x_{t_j}\in D_t^{(c)}}x_{t_j}\right|\right|_H^2 \quad (2)$$

TCA, BDA는 비선형 매핑을 커널 트릭으로 구현하였으며, 변환행렬 및 커널행렬 및 최적화 방법을 사용하여 최종 변형 결과를 얻는다. [4]는 TCA를 CPDP에 적용하였으며, 데이터 정규화 기법이 예측 모델의 성능에 크게 영향을 미친다는 것을 발견하고 적당한 정규화 기법을 선택하는 TCA 확장인 TCA+를 제안하였다. [5]는 4개 데이터 집합의 18개 프로젝트들을 TCA 포함 12개의 전이학습 기반 CPDP에 적용하였으며, 그 결과 BDA가 평균적으로 높은 성능을 보였다.

W-BDA는 BDA의 확장으로 각 클래스의 가중치를 조정하는 기법이다[5]. 식 (1)을 계산 시 조건 분포 부분 중 $P(y_t|x_t)$는 타겟 라벨값을 알 수 없으므로 계산하기 불가능하여 $P(x_t|y_t)$를 근사치로 사용하며 이 때 소스 도메인에서 학습한 분류기를 사용한다. 이는 이 클래스에 대한 두 도메인의 확률이 유사하다는 가정하에 이루어지므로 문제가 있다. 실제 데이터들은 클래스 불균형 문제가 있으며 W-BDA는 이를 해결하기 위해 조건 분포 차이를 나타내는 식 (3)을 최소화하기 위해 각 클래스를 위한 α_s와 α_t를 가중치 행렬을 이용해 구한다.

$$||P(y_s|x_s) - P(y_t|x_t)||_H^2 = ||\alpha_s P(x_s|y_s) - \alpha_t P(x_t|y_t)||_H^2 \quad (3)$$

MEDA는 도메인의 속성 공간에는 여러 데이터 왜곡이 존재하므로 원본 고차원 데이터를 저차원의 매니폴드 공간으로 변환하여 데이터 왜곡을 해결하려는 기법이다[6]. 이를 매니폴드 임베딩 과정이라 하며, 원본 데이터를 매니폴드 학습 기법을 이용하여 데이터의 내재적 기하학적 구조를 잘 표현하는 더 나은 형태로 변환하는 것을 목적으로 한다. 그 후 매니폴드 공간의 속성들을 이용하여 주변 분포와 조건 분포에 대한 동적 할당이 이루어진다. 이는 두 분포의 밸런스를 할당하는 BDA와 매우 유사하며, 차별점은 μ의 추정치 식을 제공하여 반복을 통해 가장 좋은 할당을 찾는다는 것이다.

Ⅲ. 모델 및 실험 설정

1. 데이터 집합 및 평가 척도

실험에 사용할 데이터 집합은 성능 비교를 위해 [4]에서 사용한 ReLink[7]와 AEEEM[8]을 사용하였다. ReLink는 소프트웨어 결함 정보에서 버그와 변경 사이의 링크를 복구하는 방법을 다루는 연구를 위한 데이터 집합으로 다음과 같은 3개의 프로젝트들을 포함한다: Apache HTTP Server(Apache), OpenIntents Safe(Safe), ZXing. 각 프로젝트 데이터는 26개의 정적 복잡도 메트릭 속성들로 구성되며 데이터 모듈 수는 50~200으로 작은 편이다.

AEEEM은 결함 예측 모델들을 비교하고 새로운 모델이 기존 모델들보다 개선되었는지를 평가하기 위해 만들어진 데이터 집합이며 AEEEM 이라는 이름은 각 프로젝트의 첫 글자를 딴 것이다. AEEEM은 다음과 같은 5개의 프로젝트들을 포함한다: Apache Lucene(LC), Equinox(EQ), Eclipse JDT Core(JDT), Eclipse PDE UI(PDE), Mylyn(ML). 각 프로젝트 데이터는 biweekly 버전, CVS change log에서 추출한 히스토리 정보, 소스 코드 메트릭 등 총 61개의 속성들로 구성되며, 이 속성들은 17개의 소스 코드 속성, 5개의 이전 결함 속성, 5개의 변경 엔트로피 속성, 17개의 소스코드 엔트로피 속성, 17개의 소스코드 변동 속성들을 포함한다. 데이터 모듈 수는 320~1,860으로 ReLink보다는 크다.

실험에서 소스와 타겟 프로젝트는 같은 속성 집합을 가져야하므로 각 데이터 집합을 구성하는 내부 프로젝트들끼리 소스-타겟 쌍이 구성된다. 각 쌍의 가능 수는 소스, 타겟 2개를 선택하는 순열의 수와 같으므로 ReLink는 6개쌍, AEEEM은 20개쌍이 가능하며 실험은 이들 쌍에 대해 수행한다. 예를 들어, ReLink의 Safe가 소스이고 Apache가 타켓인 경우는 Safe->Apache로 표현한다.

모델의 성능평가를 위한 척도로는 [4]에서 사용한 F-measure와 추가로 결함 예측 연구에서 많이 사용되는 AUC를 사용하였다. 두 척도 모두 1에 가까울수록 높은 성능을 나타낸다. F-measure은 정밀도(precision)와 재현율(recall)의 조화평균으로 아래의 식 (4)로 정의된다. 실제 (1)은 F-measure 식의 가중치 값을 0.5로 한 F1-measure이지만 편의를 위해 F-measure라 지칭한다. F-meaure를 사용하는 이유는 정밀도와 재현율 사이에 트레이드오프가 존재하므로 하나의 값으로 모델을 평가하기에 좋은 척도이기 때문이다. F-measure는 1에 가까울수록 높은 성능을 나타낸다.

$$F-measure = \frac{2 \times precision \times recall}{precision + recall} \quad (4)$$

AUC는 ROC curve의 면적을 측정한 척도이며, 결함 모듈수가 비결함 모듈수보다 매우 작은 불균형 데이터 집합의 이진 분류 성능을 평가하기에 적당하다. AUC 값이 0.7 이상인 모델은 성능이 높은 모델로 분류된다.

2. 파라미터 및 분류기 설정

실험을 위해서 각 도메인 적응 기법 모델을 구현하기 위해 모델의 여러 파라미터 값들을 설정해야한다. 기법들 마다 필요한 파라미터들은 조금 차이가 있다.

TCA는 $\dim, \lambda, \gamma$ 파라미터가 있고 BDA는 $\dim$, λ, μ, γ, T 파라미터가 있으며, MEDA는 $\dim, \lambda, \gamma, T$ 파라미터가 있다. dim은 도메인 적응 기법을 적용한 후에 변환된 속성의 차원을 결정하는 파라미터이다. λ는 정규화를 어느 정도 시킬 것인지 결정하는 정규화 파라미터이다. 값이 너무 크면 과대적합이 발생할 수 있고 너무 작으면 과소적합이 발생할 수 있다. γ는 RBF 커널의 대역폭 파라미터로, γ가 작아지면 가우시안 함수의 표준편차가 커져 데이터 간의 거리가 더 멀게 고려된다. 이는 처리 시간이 더 오래 걸리지만 더 부드러운 결정 경계를 가지게 한다. T는 알고리즘을 반복하는 횟수로, 반복을 통해 도메인 간의 차이를 줄여 두 도메인의 분포를 더욱 유사하게 만들려는 파라미터이다.

본 연구는 관련연구들인 [4], [5]에서 사용한 파라미터 설정들을 주로 사용하였다. dim은 전체 속성 개수의 5%~20%가 적당하다[5]. 실험에서는 5, 10, 15, 20%와 40%, 50%로 실험을 해보았고 dim값이 커질수록 MEDA를 제외한 모델들의 성능이 대체적으로 좋아졌지만 MEDA 성능은 더 안좋아졌으므로 실험 결과에는 5%인 경우를 나타내었다. BDA에서 두 분포의 가중치에 대한 균형 팩터인 μ는 사용된 데이터의 분포에 의존하는 파라미터로 [5]와 같이 0, 0.1,...,0.9,1을 넣어 실험하였으며 실험 결과에는 0.9인 경우를 나타내었다. 그 외에는 λ=0.1, γ=1로 하였다.

변환된 데이터를 학습할 분류기의 선정도 모델 성능을 결정하는 중요한 요인이다. [4]는 로지스틱 회귀분석을 사용하였다. [5]는 로지스틱 회귀분석 외에 Naive Bayse, Nearest Neighbor, CART, Random Forest, Adaboost 기법들을 사용하여 실험하였지만 로지스틱 회귀분석을 사용한 결과가 가장 좋았다. 따라서 본 연구에서도 로지스틱 회귀분석 분류기를 사용하였다.

Ⅳ. 실험 결과

기존 연구들에서 CPDP 제작에 사용하였던 TCA, BDA와 새롭게 실험하는 w-BDA, MEDA 모델들의 성능을 비교한다. TCA 결과는 [4]의 실험 결과를 나타내었고 기법 이름을 TCA[4]로 표현하였으며, BDA는 새롭게 실험한 결과를 나타내었다. 커널은 Linear, Primal, RBF 모두 실험하였지만 기존 연구들에서 사용하지 않은 Primal 커널의 결과는 제외하였다. 정규화 기법 역시 Min-Max와 Z-score를 모두 실험하였지만 기존 연구들에서 더 나은 결과를 보인 Z-score 결과만 나타내었다.

1. ReLink 실험 결과

표 1과 표 2는 각각 Linear 커널과 RBF 커널을 사용하여 구현한 모델의 F-measure 결과이다. [4]는 한 커널만 사용하였으므로 두 표의 TCA 결과는 같다.

6개쌍의 경우를 관찰한 결과 어느 모델도 다른 모델보다 모든 경우에서 좋은 경우는 없었다. 심지어 TCA[4]의 경우도 Z-score 사용 모델이 Min-Max 사용 모델보다 평균은 훨씬 좋은 성능을 보였지만 6개의 경우를 비교해보면 3개는 좋고, 3개는 더 나쁘다. 따라서 평균값에 기반한 모델 비교를 해보면 Linear 커널을 쓴 경우는 MEDA가 가장 좋은 성능을 보였다. 하지만 이 결과도 Safe->Apache의 경우가 매우 좋았고 TCA와 비교해볼 때 3개의 경우만 좋은 결과를 보였다. RBF 커널을 쓴 경우도 MEDA가 BDA, W-BDA보다 좋은 성능을 보였지만 TCA보다는 약간 낮은 성능을 보였다. 즉, F-measure로 평가한 실험에서는 가장 좋은 성능을 보인 모델이 없었다. 단지 MEDA가 BDA, W-BDA보다 조금 나은 성능을 보였다. W-BDA는 BDA보다 의미있는 상능 향상을 보이지 못했다. 또한 RBF 커널을 사용한 모델이 Linear 커널을 사용한 같은 기법 모델들보다 성능이 약간 낮았다.

표 1. 실험 결과(ReLink-Linear-F-measure)
Table 1. Experimental results(ReLink-Linear-F-measure)

Src->Tgt	TCA[4] (Min-Max)	TCA[4] (Z-score)	BDA (Z-score)	W-BDA (Z-score)	MEDA (Z-score)
Safe->Apache	0.75	0.60	0.653	0.577	0.821
Zxing->Apache	0.26	0.65	0.328	0.328	0.670
Apache->Safe	0.54	0.64	0.576	0.643	0.522
Zxing->Safe	0.08	0.65	0.459	0.459	0.541
Apache->Zxing	0.49	0.42	0.616	0.647	0.409
Safe->Zxing	0.52	0.43	0.654	0.643	0.559
Average	0.44	0.57	0.548	0.550	0.587

표 2. 실험 결과(ReLink-RBF-F-measure)
Table 2. Experimental results(ReLink-RBF-F-measure)

Src->Tgt	TCA[4] (Min-Max)	TCA[4] (Z-score)	BDA (Z-score)	W-BDA (Z-score)	MEDA (Z-score)
Safe->Apache	0.75	0.60	0.404	0.420	0.671
Zxing->Apache	0.26	0.65	0.328	0.350	0.676
Apache->Safe	0.54	0.64	0.629	0.541	0.564
Zxing->Safe	0.08	0.65	0.459	0.459	0.556
Apache->Zxing	0.49	0.42	0.660	0.642	0.457
Safe->Zxing	0.52	0.43	0.582	0.647	0.457
Average	0.44	0.57	0.510	0.510	0.564

표 3. 실험 결과(ReLink-AUC)
Table 3. Experimental results(ReLink-AUC)

Src->Tgt	Linear kernel			RBF kernel		
	BDA	W-BDA	MEDA	BDA	W-BDA	MEDA
Safe->Apache	0.680	0.644	0.503	0.607	0.607	0.481
Zxing->Apache	0.687	0.687	0.542	0.704	0.694	0.476
Apache->Safe	0.636	0.723	0.476	0.660	0.634	0.423
Zxing->Safe	0.731	0.731	0.441	0.754	0.754	0.586
Apache->Zxing	0.608	0.646	0.511	0.618	0.631	0.484
Safe->Zxing	0.637	0.614	0.519	0.492	0.615	0.538
Average	0.663	0.674	0.499	0.639	0.656	0.498

표 3은 AUC 결과이다. [4]는 AUC 실험을 수행하지 않았으므로 표에는 TCA 결과가 없다. F-measure 결과와 다른 점은 Safe->Zxing의 경우 BDA가 MEDA보다 나쁜 결과를 제외하고는 6개 경우 모두 MEDA의 결과가 나머지 2개 모델들보다 나빴다. 평균도 MEDA가 훨씬 나쁜 결과를 보였다. BDA와 W-BDA 모델들 비교는 두 커널 모두 W-BDA가 BDA보다 6개 중 4개가 같거나 좋은 성능을 보였고, 평균에서도 W-BDA가 더 나은 성능을 보였다. 두 커널들의 비교는 F-meaure와 같이 Linear 커널 이용 모델들의 성능이 조금 더 좋았다.

2. AEEEM 실험 결과

표 4와 표 5는 AEEEM 데이터 집합을 이용한 두 커널 사용 모델들의 F-measure 결과이다. ReLink 경우와 다른 점은 MEDA가 매우 나쁜 성능을 보인다는 것과 각 프로젝트 쌍의 결과가 모델별로 좀 더 명확해졌다는 것이다. TCA의 경우를 보면 Z-score 사용 모델이 Min-Max 사용 모델보다 20개 모든 경우에서 더 좋은 성능을 보였다. Linear 커널을 쓴 경우 BDA, W-BDA는 TCA보다 17개 경우가 더 좋은 성능을 보였고, RBF 커널을 쓴 경우는 TCA보다 BDA가 16개 W-BDA가 17개에서 더 좋은 성능을 보였다. 평균으로 보면 BDA, W-BDA는 TCA보

다 훨씬 좋은 성능을 보였다. 두 커널 사용 모델들의 비교는 ReLink 경우와 같이 Linear 커널이 약간 좋은 성능을 보였다. BDA와 W-BDA의 비교는 W-BDA가 BDA보다 Linear 커널에서는 18개 경우에서, RBF 커널에서는 16개 경우에서 같거나 좋은 성능을 보였다. 하지만 두 기법은 성능값이 같은 경우도 많았다. 두 기법의 평균값 비교에서도 W-BDA가 약간 높은 성능을 보였다.

표 4. 실험 결과(AEEEM-Linear-F-measure)
Table 4. Experimental results(AEEEM-Linear-F-measure)

Src->Tgt	TCA[4] (Min-Max)	TCA[4] (Z-score)	BDA (Z-score)	W-BDA (Z-score)	MEDA (Z-score)
JDT->EQ	0.43	0.59	0.640	0.659	0.813
LC->EQ	0.29	0.62	0.452	0.452	0.798
ML->EQ	0.19	0.56	0.452	0.452	0.450
PDE->EQ	0.13	0.58	0.452	0.452	0.764
EQ->JDT	0.36	0.48	0.682	0.687	0.384
LC->JDT	0.41	0.56	0.702	0.702	0.553
ML->JDT	0.02	0.54	0.718	0.702	0.326
PDE->JDT	0.25	0.52	0.702	0.702	0.484
EQ->LC	0.21	0.27	0.713	0.771	0.161
JDT->LC	0.29	0.31	0.883	0.847	0.274
ML->LC	0.14	0.25	0.863	0.863	0.151
PDE->LC	0.06	0.27	0.863	0.863	0.203
EQ->ML	0.20	0.23	0.636	0.698	0.163
JDT->ML	0.27	0.32	0.825	0.826	0.291
LC->ML	0.20	0.29	0.807	0.807	0.305
PDE->ML	0.11	0.29	0.807	0.807	0.253
EQ->PDE	0.25	0.33	0.712	0.694	0.244
JDT->PDE	0.38	0.39	0.831	0.844	0.429
LC->PDE	0.25	0.37	0.796	0.796	0.426
ML->PDE	0.32	0.37	0.796	0.796	0.312
Average	0.24	0.41	0.717	0.721	0.389

표 5. 실험 결과(AEEEM-RBF-F-measure)
Table 5. Experimental results(AEEEM-RBF-F-measure)

Src->Tgt	TCA[4] (Min-Max)	TCA[4] (Z-score)	BDA (Z-score)	W-BDA (Z-score)	MEDA (Z-score)
JDT->EQ	0.43	0.59	0.471	0.630	0.555
LC->EQ	0.29	0.62	0.452	0.452	0.569
ML->EQ	0.19	0.56	0.452	0.452	0.557
PDE->EQ	0.13	0.58	0.452	0.452	0.574
EQ->JDT	0.36	0.48	0.737	0.734	0.343
LC->JDT	0.41	0.56	0.702	0.702	0.340
ML->JDT	0.02	0.54	0.702	0.703	0.343
PDE->JDT	0.25	0.52	0.702	0.702	0.343
EQ->LC	0.21	0.27	0.790	0.634	0.170
JDT->LC	0.29	0.31	0.860	0.891	0.169
ML->LC	0.14	0.25	0.867	0.867	0.169
PDE->LC	0.06	0.27	0.863	0.863	0.171
EQ->ML	0.20	0.23	0.631	0.757	0.233
JDT->ML	0.27	0.32	0.811	0.814	0.233
LC->ML	0.20	0.29	0.807	0.807	0.236
PDE->ML	0.11	0.29	0.807	0.807	0.234
EQ->PDE	0.25	0.33	0.754	0.695	0.245
JDT->PDE	0.38	0.39	0.799	0.803	0.245
LC->PDE	0.25	0.37	0.796	0.796	0.247
ML->PDE	0.32	0.37	0.796	0.795	0.252
Average	0.24	0.41	0.713	0.718	0.311

표 6. 실험 결과(AEEEM-AUC)
Table 6. Experimental results(AEEEM-AUC)

Src->Tgt	Linear kernel			RBF kernel		
	BDA	W-BDA	MEDA	BDA	W-BDA	MEDA
JDT->EQ	0.684	0.696	0.508	0.784	0.807	0.525
LC->EQ	0.771	0.771	0.538	0.683	0.683	0.541
ML->EQ	0.704	0.704	0.536	0.572	0.572	0.492
PDE->EQ	0.721	0.721	0.491	0.687	0.687	0.456
EQ->JDT	0.674	0.668	0.507	0.802	0.803	0.528
LC->JDT	0.728	0.728	0.516	0.799	0.799	0.547
ML->JDT	0.738	0.779	0.502	0.741	0.741	0.509
PDE->JDT	0.761	0.761	0.522	0.762	0.762	0.511
EQ->LC	0.629	0.714	0.411	0.712	0.661	0.515
JDT->LC	0.761	0.755	0.467	0.790	0.645	0.449
ML->LC	0.713	0.713	0.521	0.727	0.723	0.470
PDE->LC	0.784	0.784	0.537	0.760	0.760	0.544
EQ->ML	0.527	0.528	0.499	0.441	0.682	0.520
JDT->ML	0.644	0.671	0.519	0.574	0.680	0.488
LC->ML	0.709	0.709	0.490	0.587	0.587	0.472
PDE->ML	0.681	0.681	0.512	0.605	0.605	0.522
EQ->PDE	0.673	0.632	0.523	0.566	0.642	0.501
JDT->PDE	0.718	0.741	0.500	0.647	0.647	0.500
LC->PDE	0.675	0.675	0.482	0.700	0.700	0.482
ML->PDE	0.632	0.632	0.488	0.694	0.697	0.519
Average	0.696	0.703	0.503	0.682	0.694	0.505

AUC 결과를 나타낸 표 6을 보면 두 커널을 사용한 경우 모두 W-BDA가 BDA보다 17개에서 더 좋은 성능을 보였다. 평균값 비교도 W-BDA가 약간 더 나은 결과를 보였으며, Linear 커널을 사용한 경우가 RBF 커널 사용 모델보다 좀 더 좋은 성능을 보였다. MEDA를 사용한 모델 결과가 F-measure만큼은 아니지만 꽤 큰 격차로 타 모델들 보다 좋지 않았다.

3. 토론

실험 결과들을 종합적으로 분석하여 몇가지 결론을 내릴 수 있다. BDA는 기존의 TCA 모델보다 좋은 성능을 보였다. ReLink의 F-measure 평가는 약간 안좋은 결과를 보였지만 dim을 20% 이상으로 조정한 실험부터는 더 좋은 성능을 보였다. 이는 [5]의 실험 결과와 일치한다. W-BDA는 BDA보다 6개 실험 결과들 중 1개만 같은

성능을 보이고 약간 더 나은 성능을 보였다. MEDA는 ReLink의 F-measure 실험에서는 꽤 좋은 성능을 보였지만 다른 4개의 실험 결과에서는 가장 안좋은 결과를 보였다. 또한 데이터 집합과 평가 척도에 따라 너무 심한 변동성이 나타나 CPDP 모델 제작에는 부적당한 기법으로 판단된다. 커널 간 비교는 Linear 커널을 사용한 모델이 RBF 커널 사용 모델보다 모든 경우에 성능이 더 좋았다.

실험 결과로 보면 ReLink와 AEEEM 사이에 차이가 많이 나타났다. 고성능 모델의 기준인 AUC 값이 0.7 이상인 모델은 AEEEM을 사용한 경우 W-BDA가 유일하다. 그러나 이도 평균값이고 각 프로젝트 쌍의 경우들은 매우 다른 결과들을 가진다. 이같은 결과는 프로젝트 도메인과 프로젝트 참여자, 기타 환경 요소들이 다른 소스 프로젝트와 타겟 프로젝트의 근본적인 차이점에 기인한 것이라 판단된다.

V. 결 론

수십년간 매우 많은 소프트웨어 결함 예측 모델 관련 연구들이 수행되어왔으나, 학습 데이터의 부족으로 실제 소프트웨어 개발 업무에서 모델들을 구축 및 사용하기에 어려움이 있었다. CPDP는 이를 극복하기 위한 기법들 중 하나로 학습 데이터가 충분한 소스 프로젝트의 데이터를 사용하여 구축한 모델로 타겟 프로젝트의 결함을 예측하는 기법이다. 대부분의 CPDP 연구들은 두 프로젝트의 분포 차이를 줄이기 위해 전이학습의 도메인 적응 기법들을 사용해왔다.

본 연구는 W-BDA, MEDA를 이용한 모델들을 구축하여 기존 도메인 적응 기법 모델들인 TCA, BDA와 성능을 비교하였다. 그 결과 MEDA는 다른 모델들에 비해 매우 불규칙적이고 나쁜 결과를 보였다. 하지만 BDA와 W-BDA는 TCA보다 좋은 성능을 보였다. W-BDA는 BDA보다 약간 더 좋은 성능을 보였다. 이 결과들은 대부분 평균을 비교한 결과이고 각 실험 프로젝트 쌍의 결과들은 매우 다른 결과들을 보였다. 이는 CPDP 모델들을 구축할 때는 모델 구축 기법들보다 두 프로젝트 간의 시멘틱한 유사성이 더 중요하다는 것을 의미한다.

References

[1] E. Hong, "Semi-supervised Model for Fault Prediction using Tree Methods," Journal of the Institute of Internet, Broadcasting and Communication, vol. 20, no. 4, pp. 107-113, 2020.
DOI: https://doi.org/10.7236/JIIBC.2020.20.4.107

[2] S. J. Pan, I. W. Tsang, J. T. Kwok and Q. Yang, "Domain Adaptation via Transfer Component Analysis," IEEE Trans. Neural Networks, vol. 22, no. 2, pp. 199-210, 2011.
DOI: https://doi.org/10.1109/TNN.2010.2091281

[3] J. Wang, Y. Chen, S. Hao, W. Feng and Z. Shen, "Balanced Distribution Adaptation for Transfer Learning," Proc. of ICDM, pp. 1129-1134, Nov. 2017.
DOI: https://doi.org/10.1109/ICDM.2017.150

[4] J. Nam, S. J. Pan and S. Kim, "Transfer defect learning," Proc. of ICSE, pp. 382-391, May 2013.
DOI: https://doi.org/10.1109/ICSE.2013.6606584

[5] Z. Xu, S. Pang, T. Zhang, et al. "Cross Project Defect Prediction via Balanced Distribution Adaptation Based Transfer Learning", J. Comput. Sci. Technol. vol. 34, pp. 1039-1062, 2019.
DOI: https://doi.org/10.1007/s11390-019-1959-z

[6] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang and P. S. Yu, "Visual Domain Adaptation with Manifold Embedded Distribution Alignment." Proc. of ACM Int. Conf. on Multimedia. pp. 402-410, Oct. 2018. DOI: https://doi.org/10.1145/3240508.3240512

[7] R. Wu, H. Zhang, S. Kim and S. Cheung, "Relink: Recovering links between bugs and changes," Proc. of ESEC/FSE, pp. 15-25, Sept. 2011.
DOI: https://doi.org/10.1145/2025113.2025120

[8] M. D'Ambros, M. Lanza and R. Robbes, "An extensive comparison of bug prediction approaches," Proc. of MSR, pp. 31-41, May 2010.
DOI: https://doi.org/10.1109/MSR.2010.5463279

저 자 소 개

홍 의 석(정회원)

- 1992년 : 서울대학교 계산통계학과 전산과학전공 학사
- 1994년 : 서울대학교 계산통계학과 전산과학전공 석사
- 1999년 : 서울대학교 전산과학과 박사
- 현재 : 성신여자대학교 컴퓨터공학과 교수
- 관심분야 : 소프트웨어 품질 예측, AI 엔지니어링, MSR 등

※ 이 논문은 2023년도 성신여자대학교 학술연구조성비 지원에 의하여 연구되었음.