The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 24, No. 5, pp.77-82, Oct. 31, 2024. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2024.24.5.77
JIIBC 2024-5-11

분류 머신러닝 모델의 동치 클래스 분할 테스트의 충분성 평가

Adequacy Assessment of Equivalent Class Test in Classifier Machine Learning Model

윤회진*

Hoijin Yoon*

요 약 머신러닝의 테스트 집합은 학습 데이터로 참여하지 않은 나머지 데이터들로 이루어진다. 학습 데이터와 테스트 데이터를 나누는 기준은 양적 분할 즉 일정 양의 데이터를 떼어두는 방식을 적용하여, 랜덤 선택과 같은 효과를 나타낸다. 그러나 소프트웨어 테스트 관점에서 보면, 랜덤 선택보다 오류를 잡아내기에 충분한 테스트 케이스들을 테스트 집합으로 선정한다. 이를 테스트 케이스의 충분성(adequacy)이라 하며, 충분성이 높을수록 잘 선정된 테스트 케이스가 된다. 머신러닝에서 사용되는 테스트 케이스는 이런 관점에서 충분한지를 소프트웨어 테스트의 동치분할 방식과 비교하여 살펴보고자 한다. 만일 소프트웨어 테스트 설계 기법, 즉 동치분할을 적용한 테스트 집합이 높은 충분성을 보장한다면, 적은 수의 테스트 집합으로 높은 효과를 볼 수 있다. 이는 테스트 집합의 크기를 작게하여 학습 데이터 집합의 크기가 상대적으로 커지고, 결국 학습할 데이터를 많이 확보하게 된다. 보다 큰 학습 데이터 집합으로 보다 정교한 모델을 구축할 수 있음을 기대할 수 있다.

Abstract The test set of machine learning consists of the remaining data that did not participate as training data. It is quantitative division and it is setting aside a certain amount of data which has the same effect as random selection. However from a software testing perspective, test cases sufficient to catch errors are selected as a test set rather than a random selection. This is called the adequacy of the test case, and the higher the adequacy, the better the test case is selected. We want to examine whether the test cases used in machine learning are sufficient from this perspective by comparing them with the equivalence split method of software testing. If higher sufficiency is guaranteed when applying a software test design technique, that is, equivalence splitting, high effectiveness can be achieved with a small number of test sets. This reduces the size of the test set, thereby increasing the size of the training data set and ultimately securing more data to learn. It can be expected that more sophisticated models can be built with larger training data sets.

Key Words : Equivalent class partition, Machine Learning, Test adequacy, Random partition

*정회원, 협성대학교 컴퓨터공학과 (교신저자)
접수일자 2024년 7월 18일, 수정완료 2024년 9월 6일
게재확정일자 2024년 10월 4일

Received: 18 July, 2024 / Revised: 6 September, 2024 /
Accepted: 4 October, 2024
*Corresponding Author: hj.yoon@uhs.ac.kr
Dept. of Computer Engineering, Hyupsung University, Korea

Ⅰ. 서 론

소프트웨어 테스팅에서 "잘 설계된 테스트 케이스는 소프트웨어에 오류가 있음을 보일 수 있어야 한다."[1] 오류가 있음을 보이지 못하는 테스트 케이스는 무능한 것이고, 잘 못 설계된 테스트 케이스이다. 소프트웨어 테스팅 분야에서 많이 연구되는 분야 가운데 하나가 테스트 케이스 설계 기법이다. 테스트 케이스 설계의 목적은 소프트웨어를 잘 자극하여 그 내부의 오류를 외부의 장애로 표현하게 하는 테스트 케이스가 되도록 하는 것이 다. 소프트웨어에 오류가 있음을 보이기 위하여, 테스트 케이스의 수를 많이 하여 오류가 있다는 것을 알게 될 때까지 반복적으로 테스트 케이스를 실행시킬 수도 있으나, 이는 테스트 비용을 고려할 때 실행가능성이 매우 낮다. 따라서 소프트웨어에 입력 가능한 데이터 영역에서, 오류에 예민할 것으로 보이는 효율적이고 효과적인 값을 테스트 케이스로 선정하기 위한 테스트 케이스 선정 기준(Criteria)을 테스트 케이스 설계 단계에서 매우 중요하게 다루게 된다[2]. 잘 설계된 테스트 케이스가 소프트웨어의 오류 있음을 보일 수 있다. 소프트웨어에 오류가 있음을 보일 수 없는 테스트 케이스는 즉 해야 할 일을 잘 하지 못한, 잘 못 설계된 테스트 케이스이다. 본 연구는 이와 같은 소프트웨어 테스트의 이론적 관점에서, 머신러닝의 테스트 집합을 살펴보고자 한다.

머신러닝은 데이터 집합을 분할하여 훈련 집합과 테스트 집합으로 나눈다. 훈련 집합으로 학습된 모델을 구축하고, 해당 모델이 일반적인 상황에서 잘 동작하는지를 테스트하기 위하여 테스트 집합을 적용하여 모델의 예측이 정확한지에 대한 정확성을 평가한다[3]. 모델의 입장에서 보면 100% 정확하다고 판단되는 것을 성공일 것이다. 즉 테스트 집합의 모든 데이터에 대하여 모델의 예측이 모두 맞다고 판단되는 것을 의미한다. 그러나, 테스트 집합의 품질 면에서 생각하며, 모델 내부의 오류가 있음을 보이는 것이 테스트를 잘 한 것이다. 오류률이 낮게 나올수록 좋지만, 테스트 입장에서 보면 오류가 있음을 보일수록 좋은 것이다. 소프트웨어 테스트 관점에서 보면, 일을 잘 하는 테스트 집합을 적용했을 때 비로서 테스트를 제대로 했다고 보는 것이다. 그런 이유로 오류가 있음을 보일 수 있는 테스트 집합인지 아닌지가 중요해진다.

그러나 머신러닝의 테스트 집합은 양적 분할이며 랜덤 선정기준으로 설계된다. 그동안의 소프트웨어 테스트 연구들에서 제안한 선정기준들은 랜덤 방식보다 오류에 더 예민한 결과를 갖는다. 머신러닝에서도 같은 결과가 나온다면, 현재의 랜덤 선정기준을 수정할 필요가 생긴다. 여기서 머신러닝의 특징을 간과할 수 없다. 즉 머신러닝의 특성상 데이터의 볼륨이 일정수준 이상이기 때문에 랜덤 선정도 의미가 있을 수 있다. 만일 데이터 볼륨의 크기가 랜덤 선정기준과 잘 설계된 선정기준의 차이를 줄인다면, 그 차이가 미미해지는 데이터 볼륨의 크기는 어느정도인지 분석할 필요가 있다. 이렇게 본 연구는 앞서 기술한 내용을 기반으로 다음 질문을 중심으로 분석을 진행한다.

RQ. 머신러닝에서도 테스트 케이스 선정기준(Criterion)을 적용한 테스트 집합이 랜덤 테스트 집합에 비하여 더 많은 오류를 감지하는가?

RQ에 대한 긍정 결과가 나온다면 현재의 랜덤 테스트 집합 대신 선정기준에 따른 테스트 집합 구성이 머신러닝에서도 의미가 있다고 판단할 수 있다. 그렇지 않다면, 머신러닝의 테스트 집합을 구성할 때, 소프트웨어 테스팅의 테스트 케이스 선정 기준이 영향력이 없음을 보일 수 있다.

본 논문은 2장에서 테스트 케이스 선정기준과 머신러닝 데이터 분할에 대하여 소개하고, 3장에서 RQ을 풀기 위한 실험과 분석을 기술한다. 연구의 결과를 5장에서 언급한다.

Ⅱ. 관련 연구

1. 소프트웨어 테스트 케이스 충분성

소프트웨어 테스트를 하려면, 소프트웨어의 가능한 입력과 출력들 가운데 테스트에 사용할 테스트 케이스를 선정한다. 이를 소프트웨어 테스트 설계라고 하며, 어떤 선정기준을 적용하느냐에 대한 많은 연구들이 있다. 어떤 데이터를 테스트 케이스로 선정하느냐에 따라 테스트 결과 오류를 감지하기도 하며 못하기도 하므로 테스트 케이스 선정 기준이 설계에서 매우 중요한 역할을 차지한다. 설계 과정을 통하여 선정된 테스트 케이스들을 평가하는 도구 또한 다양하다.

이 가운데 충분성(Adequacy)은 여러 연구에서 각각 해석의 의미 차이가 있으나, 본 연구에서는 프로그램 뮤테이션 분야에서 언급하는 충분성[4, 5, 6]을 기준으로 다음과 같이 살펴본다. 충분성을 갖는 테스트 집합이란, 테스트 대상에 존재하는 모든 오류에 대하여 그것을 감지하는 테스트 케이스를 집합에 포함하고 있어야 한다[7].

뮤테이션 분석은 테스트 케이스의 충분성 평가를 위하여 뮤테이션 점수를 계산하며, 이는 테스트 케이스로 해당 프로그램에 오류가 있음을 알게 될 때 점수를 확보하게 된다. 만일 선정된 테스트 케이스들이 모두 소프트웨어의 오류를 감지하는데 역할을 했다면 100%의 충분성을 갖는다. 그러나 그렇지 않고 선정된 테스트 케이스의 매우 일부분만 소프트웨어 오류 감지에 역할을 하였다면 충분성은 떨어지게 된다.

2. 머신러닝 데이터 분할

기존의 프로그래밍이 결정적 솔루션을 알고리즘으로 설계하여 코딩하는 방식이었다면, 머신러닝은 솔루션 알고리즘을 코딩하는 대신, 이미 존재하는 볼륨이 큰 데이터와 같은 패턴으로 행동하도록 모델을 구현하는 것이다[8]. 머신러닝, 즉 기계학습이라는 이름처럼 '학습'을 할 수 있는 데이터가 필수로 제공되어져야 한다. 최근 데이터의 볼륨이 큰 빅데이터 시대에 들어서서 머신러닝 분야가 실질적인 결과를 낼 수 있었던 것도 이런 이유에서이다.

소프트웨어 개발의 테스트와 같이, 머신러닝에서 구축한 모델이 잘 작성되어졌는지를 확인하는 과정이 있다. 이를 위하여 머신러닝은 주어진 데이터를 두 개의 집합으로 우선 분할한다. 하나는 학습 데이터 집합이며, 또 다른 하나는 테스트 데이터 집합이다. 테스트 데이터는 모델 입장에서 본적이 없는 매우 새로운 데이터여야 한다. 그래야 테스트가 가능한 것이다. 문제 유출없이 시험지를 작성해야 하는 원리와 같은 것이다. 이때 두 개의 집합으로의 분할은 의미적 분할이 아닌 양적 분할이다. 기본적으로 25%의 데이터를 테스트를 위한 집합으로 두고, 나머지 75%를 학습을 위한 데이터로 사용한다. 이 값은 설정에 의하여 바꿀 수 있고, 25%와 75%는 머신러닝을 구현하는 라이브러리 함수들이 설정한 기본값에 해당한다.

머신러닝의 테스트 데이터 집합은 기존의 소프트웨어 테스트에서의 테스트 케이스에 해당되나, 머신러닝은 단순한 양적분할을 통하여 테스트 데이터 집합을 구성하나, 소프트웨어 테스트의 테스트 케이스는 그 선정기법이 테스트의 성능을 좌우한다. 본 연구에서는 양적분할로 이루어진 테스트 데이터 집합과 매우 간단한 선정기준인 동치분할 방식을 적용한 테스트 데이터 집합을 "충분성" 관점에서 실험을 통하여 측정하고자 한다.

Ⅲ. 테스트 집합 충분성 평가

1. 실험 설계

머신러닝 과정을 진행하며, 이때 그림1의 (1)데이터 분할을 두가지 다른 접근으로 실행한다. 하나는 기존의 양적분할이며, 다른 하나는 소프트웨어 테스트에서 사용하는 동치분할이다. 두가지 다른 분할을 통하여 테스트 데이터의 내용이 서로 달라지며 각각의 테스트 충분성을 측정하여 실험을 진행한다. 이때 동원되는 4가지 모델과 4가지 데이터셋은 다음과 같다.

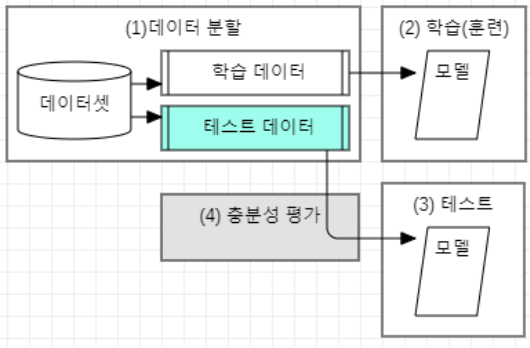

그림 1. 충분성 평가 과정
Fig. 1. Process of Adequacy Assessment

가. 분류 머신러닝 모델

테스트 집합의 충분성 평가를 위하여 결과가 명확한 지도학습모델을 실험 모델로 활용한다. 결과값이 명확하게 나오는 지도학습모델 가운데 특히 분류(Classifier) 모델을 대상으로 한다. 본 실험은 이 기준에 따라 표1의 4가지 모델을 선정하였다.

표 1. 실험 대상 머신러닝 모델
Table 1. Machine Learning Models in Experiment

모델	장점	단점
KNN	구현이 쉽다	예측 속도가 느리다
SVM	예측 속도가 빠르다.	과대적합되기 쉽다.
Decision Tree	학습 및 예측 속도가 빠르다.	과대적합되기 쉽다.
Naive Bayes	고차원 데이터 처리가 쉽다.	정확도가 떨어진다.

나. 데이터셋

UCI Machine Learning Repository[9]에 있는 데이터셋으로서 많이 사용되는 데이터셋들 가운데 4가지를 택하여 실험에 적용하였다. 라벨의 개수와 피쳐의 수, 그

리고 데이터의 수를 다양하게 구성하고 있다. 표2는 4가지 데이터셋에 대하여 설명하고 있다.

표 2. 실험 대상 데이터셋
Table 2. Data Sets in Experiment

데이터셋	설명	샘플	피쳐	라벨
Iris	4개의 특징값으로 분꽃의 종류를 분류하는 데이터	150	4	3
Breast Cancer	30가지 특징값으로 유방암을 진단하는 데이터	569	30	2
Wine	와인의 13가지 특징값으로 와인종류를 분류한 데이터	178	14	3
Digits	손글씨 데이터로 분류	1767	64	10

다. 충분성 평가 방법

기존의 머신러닝의 데스트 결과가 정확도(Accuracy)에 맞추어져 있는 것에 반하여, 본 실험의 그림 1 (4)충분성 평가는 테스트를 통하여 오류를 얼마나 잘 찾았는지를 평가한다. 이는 뮤테이션 분석의 뮤테이션 점수 계산 방식이다. 실험에서 구현한 함수는 다음과 같다. getAdequacy함수 매개변수인 x와 y는 테스트 데이터에 대한 모델의 예측값과 테스트 데이터에 담고 있는 실제값, 즉 정답에 해당된다. 즉 x와 y가 다르다는 것은 모델의 예측이 틀렸다는 것을 의미한다.

```
def getAdequacy(x,y,size):
  i=0
  pass_no=0
  fail_no=0
  while i < size:
    if(x[i] == y[i]):
       pass_no+=1
    else:
      fail_no+=1
    i+=1
  adequacy=fail_no / size
  return adequacy
```

그림 2. 충분성 측정 함수
Fig. 2. Function of Adequacy Assessment

2. 실험 결과

4가지 머신러닝 모델과 4가지 데이터셋의 조합으로 충분성을 측정하면 16가지로 나온다. 그러나 양적분할이 랜덤으로 이루어지는 특징으로 인하여 실행할 때마다 결과값이 조금씩 차이가 있다. 실험 결과를 객관적으로 유지하기 위하여 16가지가 나오는 라운드를 100번 반복수행하여, 1600가지 케이스를 만들어 충분성을 측정하였다. 그림 3은 1600개의 충분성 평가 결과 그래프이다. 붉은색의 선이 동치분할의 충분성이며 파란색의 선이 기존의 양적분할의 충분성 평가 결과이다. 대략의 모양으로 볼 때 동치분할의 충분성이 높은 수치로 보인다. 어느 정도 차이인지를 보기 위하여, 기존의 양적분할 방식의 테스트 집합의 충분성과 동치분할 방식으로 선정한 테스트 집합의 충분성을 구하고, 두 값의 차이를 계산하였다.

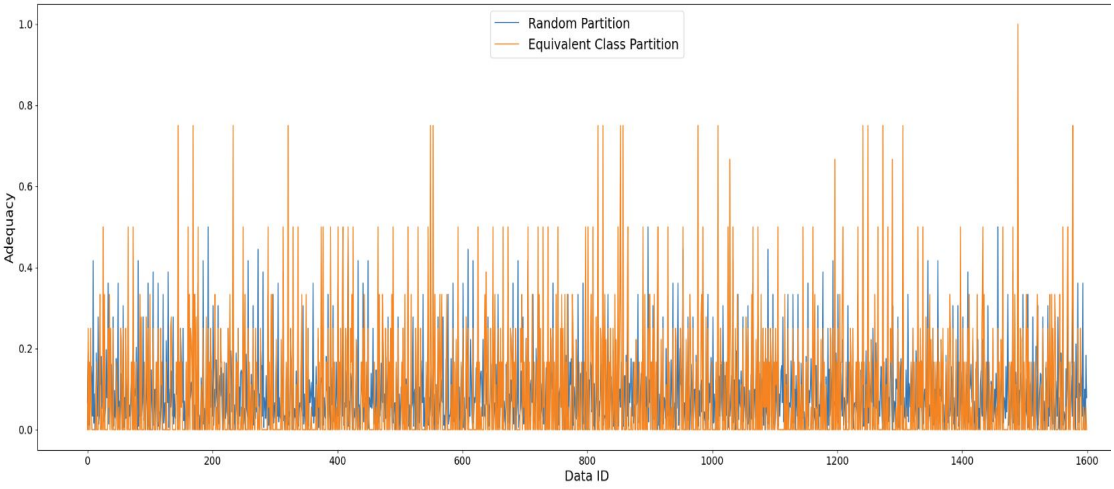

그림 3. 충분평 평가 결과 (임의 분할 vs. 동치 클래스 분할)
Fig. 3. Adequacy Comparision (Random Partiotion vs. Equivalent Class Partition)

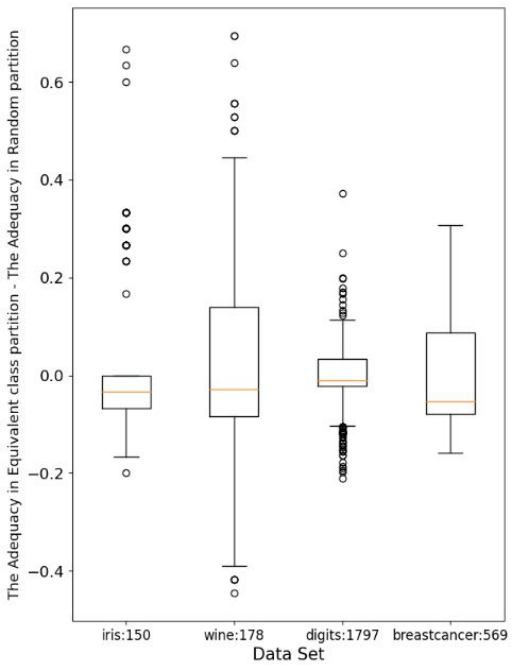

그림 4. 양적분할과 동치클래스분할의 충분성 차이
Fig. 4. Adequacy Difference between Random partition and Equivalent class partition

그림 4는 두 경우의 충분성 차이를 데이터셋에 따라 나누어 표현하였다. 동치분할 방식의 충분성에서 양적분할 방식의 충분성값을 빼는 공식으로 산출하였기에, 0이면 두 값이 같고, 양수이면 동치 클래스 분할의 충분성이 더 좋다는 의미를 갖는다. 많은 값들이 0보다 큰 양수이므로 동치 클래스 분할로 만들어진 테스트 데이터 집합의 충분성이 기존의 양적 분할로 만들어진 테스트 데이터 충분성보다 크다는 것을 보여준다.

Ⅳ. 분석 및 결론

머신러닝의 테스트 데이터는 얼마나 모델이 정확한지를 평가하는데 사용된다. 최근의 머신러닝 적용 분야의 확대[10, 11]와 그에 대한 정확성 등의 품질 요구 수준이 높아짐에 따라, 본 연구는 그 소프트웨어 테스트 관점에서 테스트에 사용되는 데이터가 오류를 감지하기에 충분한지를 평가하고자 하였다. 서론에서 제시한 RQ의 답을 찾기 위하여 실험을 진행하였으며 그 결과가 보이는대로 양적분할 보다는 동치 클래스 분할이라는 매우 간단한 테스트 케이스 선정기준을 적용한 경우가 높은 충분성을 나타내고 있다. 그렇다면 소프트웨어 테스트 선정기준을 적용할 때의 또 다른 장점은 테스트 데이터의 개수를 매우 적절하게 운영할 수 있다는 것이다. 주어진 데이터 크기의 기본적으로 25%를 테스트를 위하여 떼어두는 것이 아니라, 선정기준에 따라 능력이 있는 적은 수의 테스트 데이터를 선정하게되면, 학습에 사용되는 데이터 집합의 크기가 75%보다 커질 수 있다. 머신러님의 검증을 위하여 Cross Validation 기법을 사용하는 이유를 생각할 때, 보다 작은 사이즈의 테스트 데이터 집합을 구성하는 것은 의미가 있으며, 특히 적은 수의 테스트 데이터로 충분히 오류를 감지하는 충분성을 유지할 수 있다.

본 연구는 머신러닝을 시도해야 하는 도메인에서, 특히 데이터의 볼륨이 크지 않은 경우, 테스트 데이터의 양적분할 보다는 테스트 케이스 선정기준을 적용하여 의미있는 테스트 데이터를 활용한다면, 학습에 보다 많은 데이터를 할애할 수 있어서 모델 구축에 큰 도움이 될거라 기대된다. 본 연구는 다양한 선정기준 가운데 매우 간단한 동치 클래스 분할을 적용한 실험을 진행하였기에, 충분성 향상의 최소한의 모습만을 보일 수 있었다. 만일 다른 복잡하고 지능적인 선정기준을 적용한다면 높은 충분성 향상을 기대할 수 있다.

References

[1] Parul Ammann, Jeff Offutt, Introduction to Software Testing, Cambridge University Press, 2016.
DOI: https://doi.org/10.1017/9781316771273

[2] Mats Brindal et al, “An evaluation of combination strategies for test case selection” Empirical software engineering, Vol. 11 No. 4, 2006.
DOI: 10.1007/s10664-006-9024-2

[3] Ilsuk Oh, Machine Learning, Hanvit Academy, 2013

[4] R. A. DeMillo, R. J. Lipton and F. G. Sayward, ‘Hints on test data selection: Help for the practicing programmer’, IEEE Computer 11(4), 34-41 (1978).
DOI: 10.1109/C-M.1978.218136

[5] T. A. Budd and D. Angluin, ‘Two notions of correctness and their relation to testing’, Acta Informatica, 18(1), 31-45 (1982).

[6] E. J. Weyuker, ‘Axiomatizing software test data adequacy’, IEEE Transactions on Software Engineering 12, 1128-1138 (1986).
DOI: 10.1109/TSE.1986.6313008

[7] Jeff Offutt et al, “An Experimental Evaluation of Data Flow and Mutation Testing” Software: Pracrice and Experience, Vol.26, Issue 2, 1996.
DOI: 10.1002/(SICI)1097-024X

[8] Jae-Won Lee et al, "Design and Implementation of Machine Learning System for Fine Dust Anomaly Detection based on Big Data", The Journal of the Institute of Internet, Broadcasting and Communication, Vol.24, No. 1, 2024.
DOI: 10.7236/JIIBC.2024.24.1.55

[9] University of California Irvine Machine Learning Repository, https://archive.ics.uci.edu/

[10] Junho Lee and Jae-Pyo Park, "Examining Intelligent Failure Detection Models Using Metric Logs and Machine Learning in a Cloud Environment," Journal of the Korea Academia-Industrial cooperation Society (JKAIS), Vol. 25, No. 1, pp. 773-779, 2024.
DOI : 10.5762/KAIS.2024.25.1.773

[11] Seung-Gyu Choi, Seung-Jae Lee, and Choon-Sung Nam, Enhanced Machine Learning Preprocessing Techniques for Optimization of Semiconductor Process Data in Smart Factories," The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 24, No. 4, pp. 57-64, 2024.
DOI: 10.7236/JIIBC.2024.24.4.57

저 자 소 개

윤 회 진(정회원)

- 2004년 2월 : 이화여자대학교 컴퓨터학과(박사)
- 2005년 9월 ~ 2007년 8월 : 이화여자대학교 컴퓨터학과 전임강사
- 2007년 9월 ~ 현재 : 협성대학교 컴퓨터공학과 교수
- 관심분야 : 소프트웨어 테스팅

※ 이 연구는 2023년도 협성대학교 교내연구비 지원에 의한 연구임(2023-0040)