Journal of the Korea Academia-Industrial cooperation Society
Vol. 25, No. 1 pp. 773-779, 2024

https://doi.org/10.5762/KAIS.2024.25.1.773
ISSN 1975-4701 / eISSN 2288-4688

클라우드 환경에서 메트릭 로그와 머신러닝을 이용한 지능형 장애 탐지 모델 고찰

이준호[1], 박재표[2*]
[1]숭실대학교 금융기술융합학과, [2]숭실대학교 정보보안학과

Examining Intelligent Failure Detection Models Using Metric Logs and Machine Learning in a Cloud Environment

Junho Lee[1], Jae-Pyo Park[2*]
[1]Department of Financial Technology Covergence, Soongsil University
[2]Department of Information Security, Soongsil University

요 약 본 논문에서는 게임 서비스 장애를 예측하기 위해 서버의 메트릭 로그 데이터를 사용하여 머신러닝 모델을 구축하고 최적화하는 과정을 기술했다. 특정 게임에서 실제 운용 중인 334대의 서버 중 크기가 큰 로그 29개를 활용했고 장애 중에서 비교적 횟수가 많은 동시 접속 하락을 활용했다. 평균 10억 건 이상의 레코드를 갖고 있는 1년 치의 메트릭 로그를 클라우드 서비스에 적재했고 이 중 126개의 메트릭을 선별했다. 장애 빈도가 낮은 관계로 장애 시간 기준 1시간~3시간 전과 1일~3일 전 데이터를 추출하여 머신러닝 원본 데이터 세트를 생성했다. 분류 분석 알고리즘을 이용한 머신러닝 수행 결과, 날짜 기반 데이터 세트의 성능이 가장 잘 나왔다. 그 중에 의사 결정 트리 알고리즘과 랜덤 포레스트는 0.98 이상의 예측 성능이 나왔다. 메트릭 4개와 서버 2대를 선별해 머신러닝했을 때 의사 결정 트리 알고리즘과 랜덤 포레스트 모두 1.0의 성능이 나왔고 장애 미 탐지 오류 건수는 0건이 나왔다. 과거 연구와 달리 프로그래밍 방식으로 선택한 126개의 메트릭 중 높은 성능을 갖는 예상치 못한 메트릭을 발견할 수 있었고 이 메트릭을 활용함으로써 서버 수와 관계없는 높은 예측 성능을 달성할 수 있었다. 이 연구를 통해 특정 시점의 데이터 추출이 장애 예측에 유용한 점과 프로그램적 접근을 통해 메트릭을 선별하고 머신러닝 후에 메트릭의 중요도를 재평가하는 방식이 장애 탐지 모델 구축에 효과적임을 확인할 수 있었다.

Abstract This paper describes the process of building and optimizing a machine learning model using server metric logs to predict game service failures. Of 334 servers in operation for the game service, 29 logs were utilized, and the simultaneous connection drop was utilized. We loaded the first year's metric log, which had one billion records, to the cloud. Due to the low frequency of failures, we extracted data from one to three hours before, and from one to three days before, the failure time to generate the dataset. The proposed model uses classification algorithms for machine learning, and performs the best on date-based datasets. The decision tree algorithm and random forest showed predictive performances of more 0.98. When applying machine learning with four metrics and two servers selected, both the decision tree algorithm and random forest performed at 1.0 with zero undetected errors. Unlike past studies, after selecting 126 metrics programmatically, we found higher performance from unexpected metrics. By leveraging these metrics, we could achieve high predictive performance regardless of the number of servers. Findings show that data extraction at a certain point in time is useful for failure prediction, and identifying the importance of metrics through a programmatic approach after machine learning is effective for building predictive models.

Keywords : Failure Detection, Server Logs, Machine Learning, Decision Tree Algorithm, Random Forest

*Corresponding Author : Jae-Pyo Park(Soongsil Univ.)
email: pjerry@ssu.ac.kr
Received October 30, 2023 Revised December 11, 2023
Accepted January 5, 2024 Published January 31, 2024

1. 개요

일반적으로 장애는 발생 이후 피해를 최소화하는 방식으로 관리하고 있으며 그 이유는 예상치 못한 여러 상황 때문이다. 예를 들어, 특정 서비스에서 동시 접속자 수가 순식간에 줄어드는 경우 서버가 여러 대라면 일반적으로 그 원인을 찾기가 쉽지 않다. 하드웨어의 장애라면 여러 모니터링 서비스를 통해 원인 파악이 빠르지만 그렇지 않은 경우 관련 엔지니어들이 참여하여 개별 서버들의 과거 로그, 자원 사용량 그래프, 접속 IP 등을 통해 근본 원인을 찾기 위해 노력하게 된다. 그런 노력에도 불구하고 원인 파악이 안되는 경우 해당 서비스 개발자를 호출해 배포된 응용프로그램의 소스 수준에서 오류를 파악하기까지 하게 된다. 이처럼 다수의 서버가 포함된 서비스의 장애는 장애 유발 요인이 다양해 근본 원인의 파악이 쉽지 않다. 장애가 발생하면 원인 파악을 빠르게 하지 못해 장애 복구에 많은 시간이 소요되기 때문에 일반적으로는 장애 복구 시간을 줄이기 위해 일단 이전 소스 등으로 서비스를 원상 복구한 후 원인 파악을 다음에 진행한다.

본 연구의 내용은 특정 서비스에 대해 사전에 저장한 서버별 메트릭(수치형) 로그와 원본 데이터를 가공해 데이터 세트를 생성하고 이 데이터 세트로 머신러닝을 수행하고 모델을 구축하기 위한 구현 절차를 기술하고 구축된 모델의 성능을 측정하고, 모델의 성능을 높이기 위해 여러 가지 방식으로 원본 데이터를 가공하여 머신러닝을 수행하여 최적의 성능을 찾는 것이다. 경험치에 근거해 독립변수를 선정했던 지난 연구와 달리 본 연구에서는 독립변수의 선별을 프로그래밍 방식으로 선정한다. 끝으로 이런 방식으로 새롭게 생성된 모델과 기존 모델의 차이점으로 고찰한다[1].

2장은 이 연구와 연관된 기술인 빅데이터와 머신러닝에 대해 간략히 기술하고 이 연구와 유사한 선행 연구를 검토하고 선행 연구의 주요 특징을 기술한다. 3장은 지능형 장애 탐지 모델을 구축하기 위한 환경을 소개하고 머신러닝을 위한 데이터 세트의 가공 절차 및 머신러닝 절차를 기술하고 실제 머신러닝을 수행해 알고리즘별 머신러닝 성능을 비교 분석한다. 머신러닝 결과와 이전 연구의 결과를 비교한 후 개별 독립변수를 통해 성능을 측정한 후 최적의 머신러닝 결과를 도출한다. 4장은 3장에서 진행한 머신러닝 결과에 대한 결론을 정리한다.

2. 관련 연구

2.1 빅데이터

빅데이터는 매우 큰 DB(Very large Database), 극단적으로 큰 DB(Extremely large Database), 극단적인 데이터(Extreme data), 전체 데이터(Total data)로 기존 방식으로는 저장하고 관리하고 분석하기 어려운 대용량의 데이터를 의미한다. 또는 일반 DBMS(Database Management System)로 관리하고 분석할 수 있는 용량을 초과한 데이터를 의미한다[2].

빅데이터 플랫폼은 빅데이터를 저장하고 조회하고 관리할 수 있는 응용프로그램의 집합을 의미하며 대표적인 빅데이터 플랫폼으로 Apache Hadoop이 있다. Hadoop 아키텍처는 다수 개의 응용프로그램으로 구성돼 있으며 이를 Hadoop 에코 시스템이라고 한다[3].

클라우드 기반의 빅데이터 시스템은 Platform as a Service(PaaS) 또는 Software as a Service(SaaS)로 제공하고 있으며 적은 비용으로 빅데이터 시스템을 빠르게 구성하여 활용할 수 있다. 클라우드 기반 빅데이터 시스템에는 분산 데이터 처리 시스템(Map-Reduce)으로 아마존 웹 서비스(이하 AWS)의 EMR(Elastic Map-Reduce), Google 클라우드 플랫폼(이하 GCP)의 Google Dataflow가 있으며, 분산 데이터 저장 시스템(HDFS)으로 AWS의 S3(Simple Storage Service), GCP의 GCS(Google Cloud Storage)가 있고, 분산 데이터 DB로는 AWS의 Redshift, GCP의 Big Query가 있다.

2.2 머신러닝

머신러닝은 데이터 분석 방법론 중 하나로 기존 데이터를 활용하여 새로운 통찰력을 유도하기 위한 방법론 중 하나이다. 머신러닝의 절차에는 원본 데이터에서 표본 추출, 데이터 정제/요약, 차원 축소 등을 실시하여 데이터 세트를 확보하고 각종 머신러닝 알고리즘을 활용하여 머신러닝을 수행하여 모델을 생성하고 생성된 모델을 평가하여 예측 및 분류가 정상적인지 확인하는 단계가 있다[4].

머신러닝은 분석 목적에 따라 분류, 예측, 군집 분석, 연관성 분석으로 구분할 수 있고, 분석 방법에 따라 지도 학습, 비지도 학습, 준지도 학습, 강화 학습으로 구분할 수 있다.

예측 분석을 위한 주요 알고리즘으로 선형 회귀[5], 회귀 트리[6], K-최근접 이웃 기법[7], 인공 신경망 등이

있으며 분류 분석을 위한 주요 알고리즘으로 로지스틱 회귀(Logistic Regression), 단순 베이즈 분류 모형(Naive Bayes), 의사 결정 트리(Decision Tree), 서포트 벡터 머신(SVM), 랜덤 포레스트(Random Forest), 인공 신경망(ANN) 등이 있다[8]. 군집 분석을 위한 주요 알고리즘으로 계층적 군집 분석, 혼합 분포 군집, K-평균 군집 분석 등이 있다[9]. 그 외 연관성 분석 알고리즘 등이 있다[10].

2.3 선행 연구

Table 1은 장애 탐지(예측)에 관한 연구의 사례와 특징이다. 선행 연구 사례에는 공개된 메트릭(수치형) 로그를 활용한 예측, Syslog의 등장 키워드 집계를 통한 예측, 부하 발생을 통한 메트릭(수치형) 로그를 활용한 예측, 예측을 위한 프레임워크 소개 등이 있다.

“Syslog를 이용한 장애 예측 시스템 설계”에서는 Syslog를 시뮬레이션용으로 생성하여 DB에 적재하고 장애 예상 문자열을 사전에 등록하여 주기적인 SQL 쿼리를 통해 장애를 검출했다[11]. “클라우드 환경에서의 머신러닝 기반 장애 탐지 및 근본 원인분석 연구”에서는 프로메테우스를 통해 생성된 수치형(메트릭) 로그를 가상으로 생성하고 장애를 임의로 유발한 후 이 데이터를 이용하여 여러 가지 머신러닝 알고리즘으로 장애 탐지 성능을 비교 분석했다[12]. “시스템 가용성 보장을 위한 기계 학습 기반의 장애 예측 자동화 프레임워크 설계 및 구현”에서는 외부 공개 데이터를 활용하여 장애 탐지보다는 장애를 예측할 수 있는 프레임워크의 설계와 구현에 중점을 두었다[13]. “시스템 자원의 장애 예측 자동화를 위한 AutoML 기반의 시스템 장애 예측”에서는 장애를 포함한 외부 공개 데이터를 활용하고 AutoML을 이용하여 머신러닝을 수행하고 장애 예측 성능을 파악했다[14]. “서버 성능 관리를 위한 장애 예측 시스템”에서는 부하 발생을 시켜서 나온 수치형(메트릭) 로그 데이터를 시계열 분석하여 장애 예측 성능을 파악했다[15].

선행 연구의 주요 특징은 가상 데이터 또는 외부 데이터를 활용했고 데이터양은 1분~4주 분량을 활용했으며 가상 데이터 또는 임의로 부하를 유발하여 데이터를 생성 또는 확보했으며 장비 수량은 1대~4대 정도를 사용했다는 점이다.

Table 1. Prior Research Cases

Subject	Data	Servers	Size
A Design of Failure Prediction System by using Syslog	Syslog, Generated Data	1	4 weeks
Machine Learning-based Fault Detection and Root Cause Analysis in Cloud Infrastructure	Matric logs, Generated Data	4	4 weeks
Design and Implementation of Machine Learning based Failure Prediction Automation Framework to Ensure System Availability	Public Shared Data(Baidu)	1	7 days
Automated Machine Learning based System Failure Prediction Framework Implement for Failure Prediction Automation of System Resource	Public Shared Data(Baidu)	1	7 days
A Prediction System for Server Performance Management	Generated Data	1	1 mins

3. 지능형 장애 탐지 모델 구축

3.1 지능형 탐지 모델 구축 절차

3.1.1 머신러닝용 데이터 세트 가공 절차

Fig. 1은 지능형 장애 탐지 모델의 구축 절차이다. 1년 분량의 로그는 클라우드 스토리지에 GZIP 형태로 저장돼 있으며 장애 기록은 엑셀을 통해 날짜순으로 정리된 상태로 제공된다. 오픈 소스 기반의 ETL(Extract-Transform-Load) 라이브러리와 Map-Reduce 기반의 클라우드 빅데이터 처리 플랫폼을 이용하며 기본 데이터 세트를 가공한 후 SQL 기반의 빅데이터 데이터베이스에 저장한다.

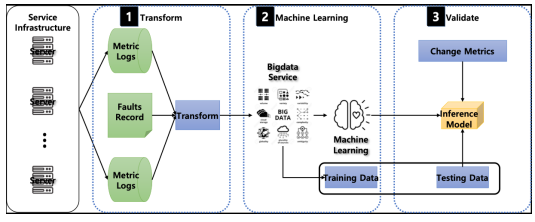


Fig. 1. Procedure for Intelligent Failure Detection Model

빅데이터 데이터베이스로 특정 장애 시간대를 SQL로 쿼리하고 결과를 데이터 세트를 처리하는 라이브러리로 장애 기록과 수치형 로그를 변환하여 머신러닝을 위한 원본 데이터 세트를 생성한다. 생성한 데이터 세

트를 훈련 데이터와 테스트 데이터로 구분한 후 머신러닝 알고리즘 라이브러리 패키지를 활용하여 머신러닝을 실행하고 예측을 수행하여 모델의 성능을 평가한다. 또한 머신러닝에 참여하는 서버 수를 증가시키면서 예측 모델의 성능 변화 추이도 측정하고 머신러닝에 참여하는 메트릭을 바꿔가면서 예측 모델의 성능 변화 추이도 측정한다.

3.1.2 원본 가공 절차

Fig. 2는 머신러닝 데이터 세트를 생성하기 위한 원본 가공 절차이다. 가공 절차는 먼저 클라우드 저장소에 저장된 메트릭(수치형) 로그를 읽는 것으로 시작한다(①). 하나의 메트릭 로그 파일을 읽어서 날짜, 키, 값으로 구문 분석하여 Big Query 테이블에 적재한다. 메트릭 파일마다 같게 처리하여 각각 Big Query 테이블에 저장한다. 메트릭 로그 저장 파일이 여러 개이므로 위 작업을 반복적으로 수행한다.

다음으로 장애 기록을 가공한다(②). 엑셀에는 장애 설명, 발생 시간, 지속 시간, 종료 시각, 장애 등급 등이 기록돼 있고 이를 구문분석하고 발생 시간을 UTC(Universal Time Coordinated)로 변환한다. 그런 다음 장애를 유형별로 구분하고 유형 컬럼을 추가한 후 CSV(Comma-Separated Values) 파일로 저장한다.

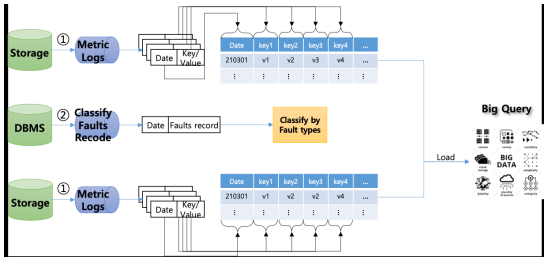


Fig. 2. Procedure for Transforming Original Data

3.1.3 머신러닝용 데이터 세트 가공 절차

Fig. 3은 머신러닝 데이터 세트 가공 절차이다. 메트릭 데이터를 Dataflow를 통해 Big Query에 저장하는 Data Mining 절차와 가공한 장애 기록을 이용해 Big Query에서 장애 시간대와 연관된 부분 집합 데이터를 가져와 장애 여부를 라벨링 하는 Query Filtering 절차와 서버별 공통 메트릭 정보를 추출하고 서버의 일련번호를 붙이는 Extraction 절차로 나눌 수 있다.

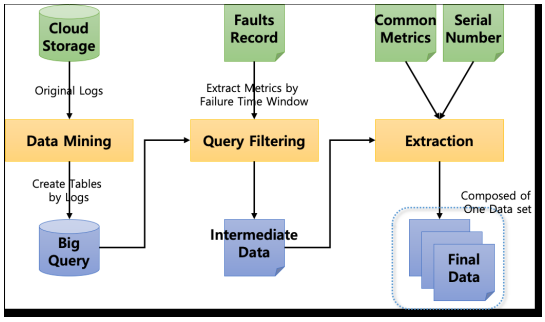


Fig. 3. Procedure for Transforming Data Set

3.1.4 머신러닝 절차

Fig. 4는 머신러닝용 데이터 세트를 이용하는 머신러닝 절차이다. 유형에 따라 분류한 장애 기록 중 특정 장애에 대해 장애 발생 시간, 장애 발생 1시간 전에서 3시간 전까지의 메트릭 기록을 Big Query에서 추출하고, 장애 발생 시간, 장애 발생 1일 전에서 3일 전까지의 메트릭 기록도 추출한다(①), 장애 기록 추출 시 장애 여부 컬럼을 추가하여 장애 시간 여부에 따라 1 또는 0으로 기록한다.

Fig. 4. Machine Learning Procedure

추출한 메트릭 및 장애 기록을 훈련 데이터와 테스트 데이터로 나누고 훈련 데이터로 머신러닝을 수행한다(②). 머신러닝은 분류 기반 알고리즘을 활용하여 실시한다. 분류 알고리즘은 로지스틱 회귀, 선형 회귀, K-최소 근접, 의사 결정 트리, 가우시안 나이브 베이즈, 서포트 벡터 머신 및 랜덤 포레스트를 이용한다(③). 시간 기반 데이터 세트와 날짜 기반 데이터 세트로 각각 머신러닝한다.

테스트는 머신러닝에 사용한 모든 알고리즘을 이용하며 실시하며 혼동행렬 및 ROC(Receiver Operating Characteristic) 커브 등으로 정밀도 등을 확인한다. 또한 시간 기반 및 날짜 기반의 성능 차이 등도 함께 테스트한다. 그리고 서버 수를 증가시켜 가면서 머신러닝을 반복적으로 수행하여 모델을 생성하고 성능 테스트를 수

행하여 서버 수에 따른 장애 탐지 성능의 변화 추이도 함께 확인한다. 추가로 참여 독립변수를 바꿔가면서 변화되는 성능 추이도 함께 확인한다. 또한 서버별 머신러닝을 통해 이 장애에 대해 어떤 서버가 영향을 직접 주는지 파악한다.

3.1.5 머신러닝 원본 생성

장애 원본 데이터는 서비스명, 장애 등급, 접수 일시, 장애 접수 제목, 발생 시간, 종료 시각 등이 기록된 엑셀 파일이다. 한국에서 제공하는 서비스이므로 장애 시간대가 한국 시간대로 기록이 돼 있다. 메트릭 로그에 있는 타임스탬프 시간은 UTC이므로 장애 원본 데이터의 시간대 변경을 위한 가공이 필요하다. Table 2는 장애에 대한 분류표이다. 시간대를 UTC로 변환한 후 특정 장애에 대한 장애 탐지를 해야 하므로 장애 유형을 분류해야 한다.

Table 2. Classification of Failure

Classification	Count
Instant Disconnect	34
Simultaneous Connect Drop	11
Network	4
DB	1
External Service	2
Hardware	2

본 연구에서는 지난 연구와 같이 동시 접속 하락을 장애 탐지의 목표로 선정한다. 데이터 추출은 지난 연구와 달리 for 루틴을 통해 일괄적으로 추출하고 dropna(axis=1)를 사용해 N/A 컬럼을 제거함으로 이전 데이터가 컬럼이 56개지만 신규 데이터는 컬럼이 129개이다. 컬럼수가 변화된 원인은 이전 연구에서는 경험에 근거해 무의미 보이는 컬럼을 임의로 제거했으나 이번 연구에서는 프로그램적으로(dropna(axis=1)) 값이 N/A인 컬럼을 제거하기 때문이다[1].

메트릭 로그가 총 334개이고 각 로그에 대해 일련번호를 부여해 이를 서버 ID로 활용한다. 메트릭 로그 중에는 1년 치 데이터를 모두 수집한 로그와 일부 데이터만 가지고 있는 로그가 있으므로 파일 사이즈가 큰 로그를 선택하여 1년 치 전체 데이터를 포함할 만한 메트릭 로그를 선택한다. 이번에 선정한 서버 대수는 모두 29대이다. Table 3은 선정한 서버 일부이다. 메트릭 로그를 순차적으로 포함한 상태로 머신러닝을 진행하여 서버 추가에 따른 머신러닝 성능 추이를 측정한다[1].

Table 3. The Example of Selected Logs

File Name	Size(bytes)	Server ID
SA-RA****.tsv.gz	1684516559	2
SA-CA****.tsv.gz	1595123904	3
SA-Co****tsv.gz	1591813914	4
SA-GM****.tsv.gz	1517675020	5
SA-M-****.tsv.gz	1501422188	6
SA-CO****.tsv.gz	1410147868	10

3.2 머신러닝 수행

3.2.1 머신러닝 수행 및 테스트 결과

Table 4는 7가지 알고리즘을 이용한 머신러닝 수행 결과이며 Time-based는 장애 시간, 장애 시간 1시간 전, 2시간 전, 3시간 전의 메트릭을 추출해 생성한 원본을 이용한 경우이고 Date-based는 장애 시간, 장애 시간 1일 전, 2일 전, 3일 전의 메트릭을 추출해 생성한 원본을 이용한 경우이다. Time-based와 Date-based의 0.9/0.3등은 머신러닝 모델 점수와 정밀도 점수이고 혼동행렬은 [[tn, fp], [fn, tp]] 형태이다.

머신러닝 수행 결과 알고리즘 전반적으로 시간 기반보다는 날짜 기반의 점수가 더 높은 것을 알 수 있다. 로지스틱 회귀의 경우 시간 기반은 예측 점수가 0.76인데 비해 날짜 기반은 예측 점수가 0.90이고 장애를 탐지하지 못한 건수(False-Negative)는 574:349이다. 선형 판별 분석의 경우는 시간 기반은 예측 점수가 0.76인데 비해 날짜 기반은 0.88로 False-Negative는 569:310이다. K-최소 근접 분석의 경우 시간 기반은 0.78이지만 날짜 기반은 0.93으로 점수가 높은 편이고 장애를 예측하지 못하는 건수(False-Negative)도 412:226이다. 의사 결정 트리의 경우 시간 기반의 경우 1.00, 날짜 기반의 경우 1.00이며 장애를 예측하지 못하는 건수는 1:0이다. 가우시안 나이브 베이즈의 경우는 시간 기반은 0.71이고 날짜 기반은 0.83이고 장애를 탐지하지 못하는 건수(False-Negative)는 490:502 수준이다. 랜덤 포레스트의 경우는 시간 기반은 0.96, 날짜 기반의 경우는 0.99이고 장애를 탐지하지 못하는 건수(False-Negative)는 96:35이다. 서포트 벡터 머신의 경우 시간 기반은 0.78이고 날짜 기반은 0.92이며 장애 미탐지 건수(False-Negative)는 543:350이다. 지난 연구와 다른 점은 의사 결정 트리의 경우는 시간 기반과 날짜 기반이 모두 1.0이 나온 점이다. 랜덤 포레스트의 경우는 여러 번 실험을 통해 Estimator와 Tree Depth 최적값을 도출한 후 테스트한다.

Table 4. Machine Learning Test Results

Algorithm	Time-based		Date-based	
Logistics Regression	0.76/ 0.76	[[1816 19] [574 33]]	0.90/ 0.90	[[3780 78] [349 251]]
Linear Discriminant	0.76/ 0.76	[[1820 15] [569 38]]	0.88/ 0.88	[[3640 218] [310 290]]
KNeighbors Classifier	0.84/ 0.78	[[1715 120] [412 195]]	0.95/ 0.93	[[3793 65] [226 374]]
Decision Tree	1.00/ 1.00	[[1835 0] [1 606]]	1.00/ 1.00	[[3858 0] [0 600]]
GaussianNB	0.71/ 0.71	[[1615 220] [490 117]]	0.81/ 0.81	[[3501 357] [502 98]]
Random Forest	1.00/ 0.96	[[1827 8] [96 511]]	1.00/ 0.99	[[3852 6] [35 565]]
SVM	0.78/ 0.78	[[1829 6] [543 64]]	0.92/ 0.92	[[3839 19] [350 250]]

3.2.2 독립변수에 따른 머신러닝 테스트 결과

머신러닝에는 126개의 독립변수를 사용하고 있으므로 어떤 독립변수가 성능에 직접적인 영향을 주는지 확인한다. 이를 위해 날짜 기반의 랜덤 포레스트 알고리즘을 이용한다. Table 5는 메트릭 별 테스트 결과 예시이다.

Table 5. New Test Results by Metrics

Metrics	Index	Performance
system\uptime in days	100	1
logicaldisk(c:)\free megabytes	20	0.974428
logicaldisk(c:)\% free space	12	0.963885
logicaldisk(d:)\free megabytes	31	0.95424
memory\pool paged bytes	42	0.944594
logicaldisk(d:)\% free space	23	0.939659

3.2.3 최적 머신러닝 테스트 결과 도출

테스트 결과가 1로 나온 임의의 서버 두 대를 이용한 머신러닝 테스트 결과를 확인한다. 알고리즘은 랜덤 포레스트를 사용한다. Table 7은 임의의 서버(5, 14)와 특정 메트릭 4개를 기반으로 한 테스트 결과이다. 테스트 값이 1.0이며 탐지하지 못한 경우는 0건이 나온다.

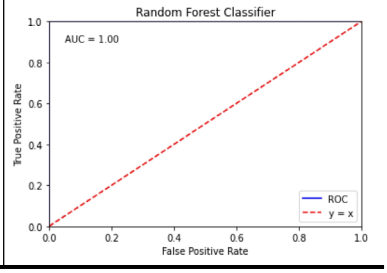
3.2.4 기존모델과의 비교

지난 연구에서는 Table 6과 같이 메모리에 관련된 독립변수의 성능이 높게 나왔다. 반면에 이번 연구에서는 'system\uptime in days', 'logical disk….' 등이 좋은 성능을 보여 준다. 지난 연구에서는 무의미해 보이는 독립변수로 여겨 제외했던 독립변수들이 대부분 좋은 성능을 보여 주고 있다. 이는 장애 발생 시 서버 재시작 등을 함으로써 가동 시간이 줄어들고, 디스크 사용량이 줄어들기 때문으로 보인다.

Table 6. Old Test Results by Metrics

Metrics	Index	Performance
memory\pool paged bytes	15	0.957789
memory\available mbytes	7	0.923618
process(_total)\virtual bytes	40	0.918593
memory\pool nonpaged bytes	14	0.915578
memory\% committed bytes in use	6	0.915327
process(_total)\private bytes	38	0.905779

Table 7. Test Results after Optimization

Machine Learning Result		
Joined Servers	5 - CA-GM**** 14 - CA-MB****	
Used Metrics	12 - logicaldisk(c:)\% free space 20 - logicaldisk(c:)\free megabytes 31 - logicaldisk(d:)\free megabytes 42 - memory\pool paged bytes	
Test Result	1.0/1.0	
Confusion Matrix	[TN FP] [FN TP]	[[280 0] [0 39]]
ROC		

4. 결론

본 연구에서는 특정 서비스의 순단 장애 탐지를 목적으로 29대 서버의 메트릭 로그 데이터를 활용하여 머신러닝 모델을 구축하고 최적화하였다. 사용한 데이터는 장애 당일부터 사흘 전까지의 로그로, 랜덤 포레스트 알고리즘을 활용한 결과, 모델 성능은 1.0을 보여 주었다.

경험에 근거해 56개의 메트릭을 선정한 지난 연구와 달리 이번 연구에서는 프로그래밍 방식으로 126개의 메트릭을 선정했으며 예상치 못한 메트릭이 기존의 가장 성능 높은 메트릭 보다 더 높게(0.958:0.974) 나왔다. 이를 통해 디스크 관련 메트릭은 동시 접속 하락 장애에 직접적으로 연관됨을 확인할 수 있었다. 이 메트릭 들을 이

용하면 참여 서버 수가 많든 적든 기존 대비 높은 예측 성능(0.987:1.0)을 보여 준다.

이전 연구에서 특정 시점을 한정해 데이터를 추출하면 장애 예측을 위한 좋은 원본의 추출이 가능함을 인지했다면 이번 연구에서는 전에 활용하지 않았던 메트릭들이 높은 예측 성능을 보여 주는 것을 발견했다는 점이 새로운 성과이다. 향후 이와 비슷한 연구를 진행할 때 메트릭의 중요도를 사전에 지정하기보다는 프로그램적으로 선별하고 머신러닝 이후 각 매트릭의 성능을 파악하는 방식으로 전환하면 좋은 예측 성능의 모델을 구축하는 데 많은 이점이 있을 것으로 예상된다.

References

[1] J. H. Lee, *An Intelligent Failure Detection Model Using Metric Logs and Machine Learning in a Cloud Environment*, Master's thesis, Soongsil University, 2022.

[2] S. Choi & S. G. Woo, "Definition, Utilization, and Trends of Big Data", *Journal of Korea Information Processing Society Review*, v.19, no.2, pp.10-19, 2012.

[3] J. H. Jeong, Start Hadoop Programming, p,760, Wikibooks, 2016.
DOI: https://doi.org/10.979.115839/0389

[4] Galit Shmueli & Jo Jae-hee, Data Mining for Business Intelligence, p.480, E&B Plus, 2012.
DOI: https://doi.org/10.978.8994246/222

[5] Peter Bruce & Andrew Bruce, Practical Statistics for Data Scientist, p.380, Hanbit Media, 2018.
DOI: https://doi.org/10.979.116224/9321

[6] S. W. Jeong, *Regression Trees & Random Forests for Symbolic Data*, Master's thesis, Sungkyunkwan University, 2018.

[7] H. S. Park, Self-Study Machine Learning + Deep Learning, p.580, Hanbit Media, 2020.
DOI: https://doi.org/10.979.116224/3664

[8] J. H. Bang, *A Study on the Market Segmentation Combining Correspondence Analysis and Cluster Analysis*, Master's thesis, Soongsil University, 2000.

[9] H. J. Seo, *Design and Implementation of Machine Learning based Failure Prediction Automation Framework to Ensure System Availability*, Master's thesis, Sejong University, 2021.

[10] Y. Kim, "A Study on Design and Implementation of Personalized Information Recommendation System based on Apriori Algorithm", *Journal of Korean Bibliography*, vol.23, no.4, pp.283-308, 2012.
DOI: https://doi.org/10.14699/KBIBLIA.2012.23.4.283

[11] I. C. Cho, *A Design of Failure Predicition System by using Syslog*, Master's thesis, Soongsil University, 2016.

[12] H. J. Won, *Machine Learning-based Fault Detection and Root Cause Analysis in Cloud Infrastructure*, Master's thesis, Soongsil University, 2020.

[13] H. J. Seo, *Design and Implementation of Machine Learning based Failure Prediction Automation Framework to Ensure System Availability*, Master's thesis, Sejong University, 2021.

[14] S. H. Choi, *Automated Machine Learning based System Failure Prediction Framework Implement for Failure Prediction Automation of System Resource*, Master's thesis, Sejong University, 2019.

[15] B. C. Lim, S. G. Kim, "A Prediction System for Server Performance Management", *Korea Information Electron Communication Technology*, Vol. 11, No. 6, pp.684-690, 2018,
DOI: http://dx.doi.org/10.17661/jkiiect.2018.11.6.684

이 준 호(Junho Lee) [정회원]

- 2022년 8월 : 숭실대학교 정보과학대학원 소프트웨어공학과 (공학석사)
- 2023년 3월 ~ 현재 : 숭실대학교 일반대학원 금융기술융합학과 박사 과정 재학 중
- 2016년 12월 ~ 현재 : (주)넥슨 코리아 재직 중

〈관심분야〉
인프라 엔지니어링/자동화, 데이터 공학, 정보 전략

박 재 표(Jae-Pyo Park) [종신회원]

- 1998년 2월 : 숭실대학교 대학원 컴퓨터학과 (공학석사)
- 2002년 8월 : 숭실대학교 대학원 컴퓨터학과 (공학박사)
- 2010년 3월 ~ 현재 : 숭실대학교 정보과학대학원 교수

〈관심분야〉
정보보안, 보안평가 및 인증, 디지털포렌식, FinTech