

**【서지사항】****【서류명】** 특허출원서**【참조번호】** PD21324**【출원구분】** 특허출원**【출원인】****【명칭】** 연세대학교 산학협력단**【특허고객번호】** 2-2005-009509-9**【대리인】****【명칭】** 특허법인(유한)아이시스**【대리인번호】** 9-2016-100121-4**【지정된변리사】** 김형상**【포괄위임등록번호】** 2020-009519-7**【발명의 국문명칭】** 기계 학습 기반 데이터베이스 튜닝 방법 및 장치**【발명의 영문명칭】** METHOD AND APPARATUS FOR TUNING DATABASE BASED ON  
MACHINE LEARNING**【발명자】****【성명】** 박상현**【성명의 영문표기】** PARK, Sang Hyun**【주민등록번호】** 670101-1XXXXXX**【우편번호】** 08004**【주소】** 서울특별시 양천구 오목로 300, 204동 3701호(목동, 현대하  
이페리온2)**【발명자】**

【성명】 진희준

【성명의 영문표기】 JIN, Hui Jun

【주민등록번호】 951031-5XXXXXX

【우편번호】 03789

【주소】 서울특별시 서대문구 연세로5나길 14, 403호(창천동)

【발명자】

【성명】 최중환

【성명의 영문표기】 CHOI, Jong Hwan

【주민등록번호】 910226-1XXXXXX

【우편번호】 21090

【주소】 인천광역시 계양구 봉오대로691번길 4, 103동 409호(작전동, 코오롱아파트)

【발명자】

【성명】 최원기

【성명의 영문표기】 CHOI, Won Gi

【주민등록번호】 910204-1XXXXXX

【우편번호】 07216

【주소】 서울특별시 영등포구 당산로42길 16, 507동 202호(당산동4가, 당산현대5차아파트)

【발명자】

【성명】 성한승

【성명의 영문표기】 SUNG, Han Seung

【주민등록번호】 901103-1XXXXXX

【우편번호】 06116

【주소】 서울특별시 강남구 봉은사로21길 45, 3층(논현동 172-7)

【출원언어】 국어

【심사청구】 청구

【이 발명을 지원한 국가연구개발사업】

【과제고유번호】 1711103018

【과제번호】 2017-0-00477-004

【부처명】 과학기술정보통신부

【과제관리(전문)기관명】 정보통신기획평가원

【연구사업명】 정보통신방송연구개발사업

【연구과제명】 (SW 스타랩) IoT 환경을 위한 고성능 플래시 메모리 스토리  
지 기반 인메모리 분산 DBMS 연구개발

【기여율】 1/1

【과제수행기관명】 연세대학교 산학협력단

【연구기간】 2021.01.01 ~ 2021.12.31

【취지】 위와 같이 특허청장에게 제출합니다.

대리인 특허법인(유한)아이시스

(서명 또는 인)

【수수료】

【출원료】 0 면 46,000 원

【가산출원료】 50 면 0 원

【우선권주장료】	0	건	0	원
【심사청구료】	17	항	891,000	원
【합계】	937,000		원	
【감면사유】	전담조직(50%감면)[1]			
【감면후 수수료】	468,500		원	

## 【발명의 설명】

### 【발명의 명칭】

기계 학습 기반 데이터베이스 튜닝 방법 및 장치{METHOD AND APPARATUS FOR TUNING DATABASE BASED ON MACHINE LEARNING}

### 【기술분야】

【0001】 이하 설명하는 기술은 기계 학습 기반 데이터베이스 튜닝 방법 및 장치에 관한 것이다.

### 【발명의 배경이 되는 기술】

【0002】 전통적으로, 관계형 데이터베이스는 구조화된 데이터를 처리하는 데 사용되어왔다. 그러나, 빅데이터 시대에는 관계형 데이터베이스를 이용하여 처리하기 어려운 비정형 데이터가 대량으로 생산되어 소비되고 있다.

【0003】 따라서 기존 데이터베이스 시스템의 한계를 극복하기 위해 키-값(key-value) 쌍의 형태로 비정형 데이터를 저장 및 관리할 수 있는 키-값 데이터베이스가 제안되었다. 예를 들어, 로그 구조 병합 트리(LSM-Tree: Log-Structured Merge-Tree)를 사용하는 디스크 기반 키-값 데이터베이스인 RocksDB가 제안되었다.

【0004】 데이터베이스 시스템에는 일반적으로 데이터베이스 관리자와 사용자가 설정해야 하는 많은 파라미터가 있다. 예를 들어, 데이터베이스 파라미터는 버퍼 크기나 체크포인트 간격과 같은 정보를 포함할 수 있다.

【0005】 데이터베이스 튜닝(Database Tuning)은 데이터베이스와 관련된 파라미터를 조정하여 데이터베이스 시스템의 성능을 향상시키는 작업을 말한다. 종래에 빅데이터를 처리하는 데이터베이스의 데이터 처리 속도를 높이고, 메모리 수명을 연장시키기 위해, 데이터베이스를 튜닝하는 기술이 연구되어 왔다.

### 【발명의 내용】

### 【해결하고자 하는 과제】

【0006】 RocksDB는 신속한 데이터 처리를 위해 로그 구조 병합 트리(LSM-Tree: Log-Structured Merge-Tree)를 사용하는 디스크 기반 키-값 데이터베이스이다. LSM-Tree는 순차 쓰기를 유도하여 우수한 쓰기 성능을 나타낼 수 있다.

【0007】 그러나, LSM-Tree를 사용하기 위해서는, 데이터 컴팩션(compaction, 압축)을 위한 추가적인 쓰기 및 공간이 필요할 수 있다. 추가적인 쓰기는 데이터베이스의 처리 성능을 저하시킬 수 있다. 또한, 과도한 쓰기 연산은 데이터가 저장되는 SSD(Solid-State Drive)의 수명을 단축시킬 수 있다. 이러한 쓰기는 예상보다 많은 데이터를 저장하게 되며, 이는 데이터베이스 운영 측면에서 저장 장치에 대한 부담을 증가시킬 수 있다.

【0008】 RocksDB에서 발생하는 쓰기 증폭(WA: write amplification) 및 공간 증폭(SA: space amplification) 문제를 최소화하기 위해서는, 데이터베이스 파라미터를 최적화해야 할 필요가 있다. 그러나, 데이터베이스의 많은 파라미터를 조정하여 성능을 향상시키는 것은 데이터베이스 전문가에게도 어려운 작업이며, 상당한

시간과 노력이 필요하다.

【0009】 따라서, 기계 학습 기반 데이터베이스 튜닝 방법 및 장치에 대한 연구가 필요하다.

### 【과제의 해결 수단】

【0010】 본 발명의 일 양태에 따른 데이터베이스 튜닝 방법은 데이터베이스 튜닝 장치에서 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계, 상기 데이터베이스 튜닝 장치에서 상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택하는 단계, 및 상기 데이터베이스 튜닝 장치에서 유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득하는 단계를 포함할 수 있다.

【0011】 본 발명의 일 양태에 따른 데이터베이스 튜닝 장치는 프로세서, 및 상기 프로세서와 결합되어 작동되는 메모리를 포함하되, 상기 프로세서는, 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계, 상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택하는 단계, 및 유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득하는 단계를 수행하도록 구성될 수 있다.

### 【발명의 효과】

【0012】 이하에서 설명하는 기술은 RocksDB와 같은 대규모 데이터베이스에서 주요 파라미터를 최적화함으로써 데이터베이스의 성능 및 속도를 향상시킬 수

있다.

【0013】 예를 들어, 본 발명의 몇몇 실시예에 따르면, 랜덤 포레스트 방법에 따라 파라미터를 선택하고 선택된 파라미터에 유전자 알고리즘을 적용하여, 효율적으로 최적화된 파라미터 값을 얻을 수 있다.

【0014】 또한, 이하에서 제안된 방법을 통해 결정된 파라미터의 최적 값을 사용하는 경우, WA 및 SA를 줄이고 데이터 처리 속도를 가속화하는 할 수 있다. 나아가, 전체 데이터 처리 시간은 RocksDB의 기본 설정에서 필요한 시간과 거의 동일하게 유지될 수 있다.

【0015】 이하에서 설명하는 기술은 빅데이터를 관리하는 데이터베이스의 성능 개선을 위한 시스템을 구축할 수 있다.

【0016】 예를 들어, 본 발명의 몇몇 실시예에 따르면, 파라미터 최적화에 소요되는 시간을 단축하여 데이터베이스를 효율적으로 관리할 수 있다.

【0017】 예를 들어, 본 발명의 몇몇 실시예에 따르면, RocksDB에서 WA와 SA의 성능에 큰 영향을 미치는 중요한 파라미터를 식별함으로써, 튜닝 부하(tuning load)를 줄일 수 있다.

### 【도면의 간단한 설명】

【0018】 도 1은 RocksDB의 컴팩션 동작의 일 예를 설명하기 위한 다이어그램이다.

도 2는 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방



법의 일 예를 나타낸다.

도 3은 본 발명의 몇몇 실시예에 따른 랜덤 포레스트 프로세스의 일 예를 나타낸다.

도 4는 본 발명의 몇몇 실시예에 따른 유전자 알고리즘의 일 예를 나타낸다.

도 5는 본 발명의 기계 학습 기반 데이터베이스 튜닝 방법에 따른, RocksDB의 성능 지표 각각에 대한 예측 정확도를 나타낸다.

도 6은 본 발명의 기계 학습 기반 데이터베이스 튜닝 방법에 따른, 지니 평균 감소(즉, 평균 감소 불순물(MDI: mean decrease impurity))에 기반한 파라미터 기여도에 대한 분석 결과를 나타낸다.

도 7은 본 발명의 기계 학습 기반 데이터베이스 튜닝 방법에 따른, 최고 및 최저 기여도를 갖는 파라미터의 비교 분석한 결과를 나타낸다.

도 8은 본 발명의 기계 학습 기반 데이터베이스 튜닝 방법에 따른, 유전자 알고리즘의 수렴 과정을 나타낸다.

도 9는 본 발명의 기계 학습 기반 데이터베이스 튜닝 방법에 의해 최적화된 RocksDB의 향상된 성능을 나타낸다.

도 10은 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법의 흐름도이다.

도 11은 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 장치의 개념도이다.

## 【발명을 실시하기 위한 구체적인 내용】

【0019】 이하 설명하는 기술은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

【0020】 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

【0021】 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 설명된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계 동작 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.

【0022】도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에도 다른 구성부가 담당하는 기능 중 일부 또는 전부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.

【0023】또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다.

【0024】본 발명의 몇몇 실시예에 따르면, 기계 학습 기반 데이터베이스 튜닝 방법은 수많은 RocksDB 파라미터를 처리하기 위해, 기계 학습 (machine learning) 기반 특징 선택 기술(feature selection technique)을 활용하여 중요한 RocksDB 파라미터(parameter)를 식별하고, 유전자 알고리즘(genetic algorithm)을 활용하여 넓은 검색 공간을 탐색하고 최적의 값을 효율적으로 찾을 수 있다.

【0025】구체적으로, 기계 학습 기반 데이터베이스 튜닝 방법은 다음과 같은 동작을 수행할 수 있다.

【0026】 - 튜닝 부하(tuning load)를 줄이기 위해 RocksDB에서 WA와 SA의 성능에 큰 영향을 미치는 중요한 파라미터를 식별할 수 있다.

【0027】 - RocksDB에 대한 특정 적합도 함수(fitness function)가 있는 유전자 알고리즘을 활용하여 식별된 데이터베이스 파라미터의 최적 값을 얻을 수 있다.

【0028】 - 유전자 알고리즘을 통해 결정된 최적값을 사용하면 WA 및 SA를 줄이고 데이터 처리 속도를 가속화하는 반면, 전체 데이터 처리 시간은 RocksDB의 기본 설정에서 필요한 시간과 거의 동일하게 유지할 수 있다.

【0029】 이하에서, 도 1 내지 도 4를 참조하여 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법에 대해 자세히 설명한다.

【0030】 도 1은 RocksDB의 컴팩션 동작의 일 예를 설명하기 위한 다이어그램이다. 도 1은 (a) 컴팩션 전 RocksDB의 LSM-Tree 상태 및 (b) 컴팩션 후 RocksDB의 LSM-Tree 상태를 나타낸다.

【0031】 도 1을 참조하면, RocksDB는 LSM-Tree를 기본 구조로 사용할 수 있다. LSM-Tree는 키로 정렬된 파일로 구성된 여러 레벨로 구성될 수 있다. LSM-Tree는 컴팩션 작업을 사용하여 데이터를 관리할 수 있다. 컴팩션(Compaction)은 LSM-Tree 구조의 유지를 가능하게 하는 쓰기 방법이다. 컴팩션은 각 레벨의 임계값을 초과하면 트리거될 수 있다. RocksDB는 SST(정적 정렬 테이블)를 사용하여 영구 데이터를 저장하고 관리한다.

【0032】 도 1의 (a)를 참조하면, 컴팩션이 트리거되면 레벨  $n$ 에 존재하는 희생 SST 파일(victim SST)이 선택될 수 있다. 키 범위가 희생 SST 파일과 겹치는 SST 파일(Overlapped SST)은, 임계점을 초과한 레벨의 하위 레벨인, 레벨  $n + 1$ 에서 추가로 선택될 수 있다.

【0033】 선택된 SST 파일은 도 1의 (b)와 같이 병합 정렬(merge sort)을 통해 새로 생성된 SST 파일(Newly generated SST)로 생성되고, 하위 레벨( $n+1$  레벨)에 저장될 수 있다.

【0034】 컴팩션을 위해 선택된 SST 파일은 쓸모없는 SST 파일(Obsolete SST)로 관리되고 나중에 가비지 컬렉션 (garbage collection) 시스템을 통해 삭제될 수 있다.

【0035】 이와 같이, 컴팩션은 LSM-Tree에서 공간을 관리하는 작업이다. 컴팩션 동작은 임계 값에 도달한 레벨에 존재하는 파일을 하위 레벨로 전송하는 것에 의해 수행될 수 있다.

【0036】 도 1을 참조하면, 레벨  $n$ 의 희생 SST 파일을 하위 레벨로 보내기 위해, RocksDB는 레벨  $n + 1$ 에서 겹치는 SST 파일을 찾고 희생 SST 파일과 겹치는 SST 파일 모두에 병합 정렬(merge sort)를 적용할 수 있다.

【0037】 이 과정에서 추가 쓰기 작업이 수행되며, 이러한 현상을 쓰기 증폭(즉, WA)이라고 한다. RocksDB는 다중 스레드(multi-thread) 데이터베이스이기 때문에 많은 컴팩션 작업을 동시에 수행하므로, 쓰기 작업 수가 기하급수적으로 증가

할 수 있다.

【0038】 과도한 디스크 입출력 작업을 수행하는 경우, 데이터 처리 성능이 저하될 수 있다. 또한 WA는 데이터 처리 성능을 저하시키고, SSD와 같은 플래시 메모리 기반 장치의 수명을 단축시킬 수 있다. SSD는 파일을 쓰기 위한 쓰기 횟수에 제한이 있기 때문에, 불필요한 디스크 입출력 작업을 피하는 것이 중요하다.

【0039】 또한 WA 현상 외에도, RocksDB는 데이터 저장 공간 활용과 관련된 또 다른 문제가 있다. 도 1과 같이 더 이상 유효하지 않은 SST 파일 (invalid SST) (또는, 쓸모없는 SST 파일)은 가비지 컬렉션에 의해 삭제될 때까지 디스크 공간을 차지할 수 있다. 이러한 공간 점유 현상을 공간 증폭(즉, SA)이라고 한다. 이렇게 누적된 잘못된 SST 파일은 수많은 컴팩션 작업이 동시에 수행되기 때문에 상당한 공간을 차지할 수 있다.

【0040】 도 2는 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법의 일 예를 나타낸다.

【0041】 도 2를 참조하면, S201 단계에서, data를 생성할 수 있다. 예를 들어, RocksDB는 db\_bench를 사용하여 2,000개의 임의 파라미터 조합 (random parameter combination)으로 샘플을 생성할 수 있다.

【0042】 S202 단계에서, 파라미터를 추출(parameter extraction)할 수 있다. 예를 들어, 랜덤 포레스트를 이용하여 각 파라미터의 기여도를 계산하고, 기여도가 가장 큰 영향력 있는 파라미터를 선정할 수 있다.

【0043】 예를 들어, 기여도가 높은 파라미터부터 차례대로 파라미터를 정렬할 수 있다. 예를 들어, 도 2를 예시, 파라미터 1 (parameter 1)은 기여도가 가장 높은 파라미터다. 또한, 파라미터 2, 파라미터 3, ... 파라미터 7, ... 순서로 낮은 기여도를 가질 수 있다.

【0044】 예를 들어, S202단계에서는 파라미터들 중 가장 기여도가 높은 4개의 파라미터를 추출하여, 파라미터를 최적화 하는데 사용할 수 있다.

【0045】 도 2의 S202 단계에서는 4개의 파라미터를 추출하는 구성을 개시하였으나, 본 발명은 이에 제한되지 않는다. 예를 들어, S202단계에서는 하나 이상의 파라미터를 추출할 수 있다. 예를 들어, S202단계에서는 자연수 N 개의 파라미터를 추출할 수 있다. 이때, N은 사용자에게 의해 설정되거나, Random Forest의 결과값에 따라 결정될 수 있다. 예를 들어, S202 단계에서, 일정 기여도 이상을 갖는 파라미터를 추출할 수 있다.

【0046】 S203 단계에서, 추출된 파라미터를 검증(verification)할 수 있다. 예를 들어, 분산 분석(ANOVA: analysis of variance)을 이용하여 RocksDB의 파라미터와 성능 간의 상관관계를 확인할 수 있다.

【0047】 S204 단계에서, 파라미터를 최적화(parameter optimization) 할 수 있다. 예를 들어, 유전자 알고리즘을 사용하여, S202 단계에서 선택된 파라미터를 최적화시킬 수 있다. 예를 들어, S202 단계에서 4개의 파라미터가 선택된 경우, 파라미터 1, 2, 3, 및 4의 값을 각각 x1, x2, x3, 및 x4로 최적화할 수 있다.

【0048】 최적화된 파라미터는 RocksDB에 사용될 수 있다. 이 경우, 본 발명에 따른 최적화된 파라미터를 사용한 RocksDB는 기본 설정을 사용한 RocksDB보다 크게 향상된 성능을 가질 수 있다.

【0049】 도 3을 참조하여, 랜덤 포레스트(Random Forest) 및/또는 기여도 분석(Contribution Analysis) 프로세스(process)에 대해 설명한다. 랜덤 포레스트 및/또는 기여도 분석 프로세스는 도 2의 S202 단계에서 사용될 수 있다.

【0050】 도 3은 본 발명의 몇몇 실시예에 따른 랜덤 포레스트 프로세스의 일 예를 나타낸다.

【0051】 랜덤 포레스트는 다수의 결정 트리들을 학습하는 앙상블 방법이다. 랜덤 포레스트 프로세서는 부트스트래핑(bootstrap)을 이용하여 트레이닝 데이터로부터 복수의 서브 데이터를 생성할 수 있다. 랜덤 포레스트 프로세서는 각각의 서브데이터에 대한 결정 트리(decision tree)를 생성할 수 있다.

【0052】 예를 들어, 각각의 결정 트리는 계층 구조로 이루어진 노드(node)들과 에지(edge)들의 집합을 포함할 수 있다. 결정 트리는 결정을 내리기 위해 사용하는 트리로, 결정 과정을 간단한 문제들로 이루어진 계층 구조를 포함할 수 있다.

【0053】 랜덤 포레스트 프로세스는 각 훈련 샘플로 훈련된 결정 트리를 결합하여, 예측 또는 회귀 분석을 수행할 수 있다.



【0054】 도 3을 참조하면, 랜덤 포레스트 프로세서는 부트스트래핑을 이용하여, 훈련 데이터로부터 서브 데이터 1, 2, 및 3을 생성할 수 있다. 랜덤 포레스트 프로세서는 서브 데이터 1, 2, 및 3 각각으로부터 결정 트리를 생성할 수 있다. 랜덤 포레스트 프로세서는 각각의 결정 트리를 이용하여, 예측을 수행할 수 있다. 예를 들어, 각각의 결정 트리를 이용한 예측 1, 2, 및 3 (prediction 1, 2, 및 3)을 이용하여, 평균화된 예측(prediction by Averaging)을 결정할 수 있다.

【0055】 다시 말하면, 랜덤 포레스트 프로세스는 부트스트랩(bootstrap)을 통해 여러 훈련 데이터 (training data) 또는 훈련 샘플(training sample)을 생성할 수 있다. 랜덤 포레스트 프로세스는 머신 러닝 모델일 수 있다.

【0056】 랜덤 포레스트 분류기 (random forest classifier)는 랜덤 포레스트 프로세스를 사용하는 분류기일 수 있다. 랜덤 포레스트 분류기는 엔트로피 또는 지니 지수(Gini index)를 이용하여 각 특징의 불순물(impurity)을 계산하고, 불순물을 최대한 줄이는 특징 노드를 구축할 수 있다.

【0057】 유사하게, 랜덤 포레스트 회귀 모델(random forest regressor)은 랜덤 포레스트 프로세스를 사용하는 회귀 모델일 수 있다. 랜덤 포레스트 회귀 모델은 평균 제곱 오차를 사용하여 불순물을 정의하고 최적의 가치를 만들 수 있다.

【0058】 랜덤 포레스트 회귀 모델은 각 특성의 불순물을 사용하여 평균 감소 불순물(MDI: mean decrease impurity)을 계산할 수 있고, 이들은 여러 결정 트리에서 계산될 수 있다. 계산된 MDI를 기반으로, 분류(classification) 및 회귀

(regression)에 대한 특성 기여도(feature contribution)가 계산될 수 있다.

【0059】 평균 불순물 감소(average impurity reduction)가 크다는 것은 해당 특성(즉, 파라미터)가 중요하다는 것을 의미할 수 있다. 기능 기여도는 추가 분석을 통해 검토해야 한다.

【0060】 이하에서, 회귀 모델 결정 계수 (Regression Model Coefficient of Determination)에 대해 설명한다. 회귀 모델 결정 계수는, 도 2의 단계 S202에서 사용되는 랜덤 포레스트 프로세스를 평가하기 위해 사용될 수 있다.

【0061】 회귀 모델 결정 계수는, 랜덤 포레스트 회귀 모델이 훈련 샘플을 효율적으로 학습했는지 여부를 평가하기 위한 결정 계수이다.

【0062】 아래의 식 (1)은 회귀 모델 결정 계수를 계산하기 위한 수학적식을 나타낸다.

【0063】

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (1)$$

【0064】 여기에서,  $R^2$ 는 실제 값()과 예측 값()에 대한 잔차의 제곱과 실제 값()과 실제 값의 평균의 잔차의 제곱의 비율로 계산될 수 있다.

【0065】 여기서  $\bar{y}$  는 실제 값의 평균이다.

【0066】  $R^2$ 의 값은 0과 1 사이이다.  $R^2$ 의 값이 1에 가까운 경우, 모델이 훈련 데이터를 더 효과적으로 설명하는 것으로 볼 수 있다.

【0067】 이하에서, 분산 분석 (ANOVA)에 대해 설명한다. 분산 분석은 도 2의 단계 S203에서 사용될 수 있다.

【0068】 분산 분석은 평균 간의 차이를 분석하는 데 사용되는 통계 모델 및 관련 추정 절차를 포함할 수 있다. 예를 들어, 분산 분석은 전체 분산의 법칙을 기반으로 하며, 여기서 특정 변수에서 관찰된 분산은 다양한 변동 소스에 기인하는 구성요소로 분할될 수 있다. 가장 간단한 형태의 분산 분석은 둘 이상의 모집단 평균이 동일한지 여부에 대한 통계적 검정을 제공할 수 있다.

【0069】 본 발명의 몇몇 실시예에서, 범주형 변수(categorical variables)를 기반으로, 각 성과 지표(performance indicator)의 차이를 분석하기 위해 분산 분석이 사용될 수 있다.

【0070】 예를 들어, 분산 분석은, '주어진 범주형 변수(categorical variables)에 대한 종속 변수(dependent variable)의 전체 값의 분산의 합'에 대한 '범주 항목(category items)들에 따른 종속 변수의 분산의 합'의 비율을 계산함으로써, 항목 간의 종속 변수 분포의 차이를 통계적으로 검정할 수 있다.

【0071】 p-값의 유의 확률(significance probability)은 검정 통계(test statistics)를 사용하여 계산될 수 있다. 일반적으로 p-value가 0.05보다 작은 경우에, 변수 값에 기초한 성능에 차이가 있는 것으로 해석될 수 있다.

【0072】 이하에서, 도 4를 참조하여, 유전자 알고리즘(Genetic Algorithm)에 대해 설명한다. 유전자 알고리즘은 도 2의 단계 S204에서 사용될 수 있다.

【0073】 유전자 알고리즘은 생명체의 진화를 모방한 모델이다. 유전자 알고리즘에서는 돌연변이(mutation), 교배(crossover), 세대(generation), 염색체(chromosome), 유전자(gene) 등의 용어가 사용될 수 있다. 유전자 알고리즘은 최적화 문제를 해결하기 위해 사용될 수 있다.

【0074】 도 4는 본 발명의 몇몇 실시예에 따른 유전자 알고리즘의 일 예를 나타낸다. 보다 구체적으로, 도 4는 유전자 알고리즘의 부분적 과정을 도시한다.

【0075】 도 4를 참조하면, 네 가지의 상이한 랜덤 파라미터 세트(random parameter set)가 초기 세대(initial population)으로 사용될 수 있다.

【0076】 각각의 랜덤 파라미터 세트는 염색체(chromosome)으로 불릴 수 있다. 각 파라미터(parameter)는 유전자(gene)로 불릴 수 있다.

【0077】 단계 S401에서, 초기 세대(initial generation)의 4개 염색체를 포함할 수 있다.

【0078】 단계 S402에서, 초기 세대의 4개 염색체의 순위(ranking)가 정해질 수 있다. 예를 들어, 4개의 염색체 각각에 피트니스 함수(Fitness Function)가 적용되어, 순위가 매겨질 수 있다. 피트니스 함수는 RocksDB의 4가지 성능 지표의 비율의 합으로 정의될 수 있다.

【0079】 예를 들어, 피트니스 함수는 아래의 수학식 (2)와 같다.

【0080】

$$Fitness = 1000 \times \left[ \sum_{\Gamma \in \{WAF, SA, TIME\}} \left( \frac{\Gamma_{default} - \Gamma_{chromosome}}{\Gamma_{default}} \right) + \left( \frac{RATE_{chromosome} - RATE_{default}}{RATE_{default}} \right) \right] \quad (2)$$

【0081】 단계 S403에서, 가장 높은 적합도 값(fitness value)을 가진 최고의 염색체가, 유전자 알고리즘의 요구 사항을 만족하는지 여부를 결정하기 위해 평가(estimation)될 수 있다. 최고의 염색체가 이 요구 사항을 충족하면 유전자 알고리즘이 중지될 수 있다.

【0082】 단계 S404에서, 최고의 염색체가 유전자 알고리즘의 요구사항을 만족하지 않는 경우, 우수한 염색체들만 선택할 수 있다. 구체적으로, 가장 높은 적합도 값과 다음으로 높은 적합도 값을 가진 두 개의 염색체를 보존(preserve)할 수 있다. 반면에, 최하위 적합도 값을 가진 염색체들은 폐기(discard)될 수 있다.

【0083】 단계 S405에서, 살아남은 상위 2개의 염색체는 다음 세대의 부모가 될 수 있다. 이들을 교배하여 새로운 염색체가 생성될 수 있다. 한편, 유전자 돌연변이가 특정 확률로 발생하여, 새로운 유전자가 삽입될 수 있다. 따라서, 다양한 염색체가 발생할 수 있다. 다음 세대(next generation) 염색체는 부모 염색체 2개, 교차 염색체 1개, 돌연변이 염색체 1개를 포함할 수 있다.

【0084】 다음 세대 염색체는 피트니스 함수를 사용하여 다시 순위가 매겨질 수 있다. 즉, 다음 세대 염색체에 대해, 다시 단계 S402가 적용될 수 있다. 즉, 유전자 알고리즘의 단계 S402, S403, S404, 및 S405는 반복될 수 있다.

【0085】 도 4에 따른 유전자 알고리즘에서, 적합도 값이 높은 염색체는 계속해서 생존할 수 있다. 또한, 세대가 거듭될수록 적합도 값이 가장 높은 염색체는 최적화된 해결책(optimal solution)에 가까워질 수 있다.

【0086】 이하에서, 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법의 구체적인 실시예 및 유리한 효과에 대해 설명한다.

【0087】 【표 1】

OS	CentOS 7.3.1611 (x86_64)
CPU	Intel® Xeon® CPU E5-2660 v2 @ 2.20GHz
RAM	Samsung M393B2G70QH0-YK 16 GB*4
NVMe	Intel SSDPEDMD800G4 DC P3700 800 GB
RocksDB	Version 6.13

【0088】 표 1은 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법이 적용된 환경을 나타낸다. 예를 들어, RocksDB의 성능을 측정하기 위해 db\_bench를 사용할 수 있는데 db\_bench는 RocksDB가 제공하는 성능 측정 도구가 제공될 수 있다. 예를 들어, RocksDB 레벨 크기, 컴팩션 기법, 데이터 크기, 개수 등 다양한 파라미터에 대한 벤치마크가 제공될 수 있다.

【0089】 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법이 적용되는 RocksDB의 성능을 측정하기 위해, 1,000,000개의 키-값 쌍을 로드할 수 있다. 이때, 키 크기와 값 크기는 각각 16B와 1KB로 고정될 수 있다. 또한, 약 200개의 RocksDB 파라미터 중에서 WA 및 SA와 관련된 30개의 파라미터를

선택할 수 있다. 이 파라미터는 임의의 값을 가질 수 있다. 2,000개의 임의 조합에 대한 데이터를 생성하였으며, LSM-Tree의 크기를 기본 크기의 1/64로 줄여 데이터 생성 시간을 효율적으로 단축할 수 있다.

【0090】 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법이 적용되는 RocksDB의 성능을 측정하기 위해, 다양한 파라미터를 기반으로 RocksDB의 성능 변화를 분석할 수 있다. 구체적으로, WA 요소(WAF: Write Amplification Factor), 로드된 데이터 크기(SA), 데이터 처리 속도(RATE), 실행 시간(TIME)의 4가지 성능 지표(Performance Indicators)를 선택하여 사용할 수 있다. 이때, WAF 및 SA를 개선하기 위해 실제로 사용할 수 없는 RATE 및 TIME 값을 피할 수 있도록, RATE 및 TIME을 선택할 수 있다. 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법의 목적은 전반적인 성능을 향상시키는 것일 수 있다. WAF는 스토리지에 기록된 데이터 양과 데이터베이스에 기록된 데이터 양의 비율을 나타내는 성능 지표이며, 아래의 식 (3)과 같이 계산될 수 있다.

【0091】

$$WAF = \frac{\text{Bytes written to storage}}{\text{Bytes written to database}} \quad (3)$$

【0092】 SA(즉, 로드된 데이터 크기)는 실제 LSM 트리에 기록된 데이터의 크기로 측정될 수 있다. LSM-Tree에는 유효한 SST 파일과 유효하지 않은 SST 파일이 모두 포함되어 있기 때문에 LSM-Tree에 기록된 데이터의 크기는 RocksDB가 이용하는 실제 물리적 공간 크기에 대한 효과적인 지표로 사용될 수 있다. 데이터 처리

속도는 컴팩션, 읽기 및 쓰기 작업을 포함하여 RocksDB가 초당 처리하는 작업 수를 나타낼 수 있다. 수행 시간은 데이터 기록의 시작부터 끝까지의 시간 간격을 나타낼 수 있다. RATE와 TIME은 이 실험에서 정비례하지 않을 수 있다. 예를 들어, RocksDB에 할당된 버퍼 크기가 증가하면 RATE가 증가할 수 있다. 그러나, 높은 압축률을 갖기 때문에, 연산 횟수가 증가되는 경우에도, TIME은 동일하게 유지될 수 있다.

【0093】 도 5는 본 발명의 기계 학습 기반 데이터베이스 튜닝 방법에 따른, RocksDB의 성능 지표 각각에 대한 예측 정확도를 나타낸다. 구체적으로, 도 5는 4가지 성능 지표인 WAF, SA, RATE, 및 TIME에 대한 랜덤 포레스트 기반 분석 결과를 도시한다. 예를 들어, 도 5의 예측 결과는 scikit-learn에서 구현된 랜덤 포레스트를 사용한 결과일 수 있다.

【0094】 도 5를 참조하면, 2,000개의 샘플을 8:2의 비율로 훈련 샘플과 테스트 샘플로 나누어, 랜덤 포레스트의 성능을 평가하기 위한 교차 검증을 수행할 수 있다. 테스트 샘플의 경우 WAF, SA, RATE 및 TIME을 예측하기 위한 결정 계수는 각각 0.663, 0.747, 0.712 및 0.697일 수 있다. 따라서, 학습 효과가 있다고 볼 수 있다. 훈련된 랜덤 포레스트는 파라미터 기여 분석을 위해, 이하에서 지속적으로 사용될 수 있다.

【0095】 도 6은 본 발명의 기계 학습 기반 데이터베이스 튜닝 방법에 따른, 지니 평균 감소(즉, 평균 감소 불순물(MDI: mean decrease impurity))에 기반한 파라미터 기여도에 대한 분석 결과를 나타낸다.



【0096】 도 6을 참조하면, `min_level_to_compress`의 기여도가 가장 컸고, `compression_type`, `thread`, `level0_slowdown_writes_trigger`, `write_buffer_size`, `compression_ratio` 등이 그 뒤를 이었다.

【0097】 `min_level_to_compress` 파라미터는 파라미터 값과 동일한 레벨에서 컴팩션을 수행하도록 RocksDB를 설정할 수 있다. 이 파라미터보다 낮은 레벨은 컴팩션되지 않을 수 있다. `min_level_to_compress` 파라미터는 기본 값은 -1이며, 이 경우 모든 레벨에서 컴팩션이 수행될 수 있다.

【0098】 `compression_type`은 적용할 컴팩션 알고리즘을 나타낸다. 예를 들어, 기본 알고리즘은 `snappy`일 수 있다. 한편, `none`, `zlib`, `bzip2` 및 `lz4`와 같은 다른 알고리즘이 있을 수 있다.

【0099】 `thread`는 요청된 명령을 실행할 스레드 수를 나타낼 수 있다. 예를 들어, 기본값은 1일 수 있다.

【0100】 `level0_slowdown_writes_trigger`는 RocksDB의 쓰기 속도를 늦추는 L0 파일의 수를 제어할 수 있다. 예를 들어, 기본값은 20일 수 있다.

【0101】 `write_buffer_size`는 컴팩션 전에 RocksDB memtable에 버퍼링되는 바이트 수 일 수 있다. 기본값은 64MB일 수 있다.

【0102】 `compression_ratio`는 데이터 컴팩션률을 나타낼 수 있다. 기본값은 0.5일 수 있다.

【0103】 도 7은 본 발명의 기계 학습 기반 데이터베이스 튜닝 방법에 따른, 최고 및 최저 기여도를 갖는 파라미터의 비교 분석한 결과를 나타낸다.

【0104】 구체적으로, 도 7은, 도 6에 개시된 파라미터의 기여도를 기반으로, 상단과 하단에서 각각 두 개의 파라미터를 선택하여 ANOVA를 수행한 결과를 나타낸다.

【0105】 구체적으로, 도 7(a)와 7(b)는 가장 높은 파라미터에 해당하는 `min_level_to_compress` 및 `compress_type`에 기초한, SA와 RATE의 성능 차이를 나타낸다. 각 성능 차이를 ANOVA를 통해 통계적으로 검정한 결과, p-value는 각각  $4.84e-31$ ,  $8.53e-68$ 이며, 이들은 유의 수준 (significance level) 0.05보다 유의하게 작을 수 있다. 따라서, `min_level_to_compress`와 `compress_type`에 따른, 성능 차이가 상당함을 확인할 수 있다.

【0106】 도 7(c) 및 7(d)는 가장 낮은 파라미터에 해당하는 `mmap_write`와 `target_file_size_multiplier`를 기반으로 WAF와 SA의 성능 차이를 분석한 결과를 나타낸다.

【0107】 가장 높은 파라미터의 결과와 달리, p-value는 각각 0.741, 0.954이며, 유의 수준보다 높은 값을 나타낼 수 있다. 따라서, 두 파라미터는 RocksDB의 성능에 크게 기여하지 않는 것으로 볼 수 있다.

【0108】 즉, `min_level_to_compress` 값을 기준으로 최소(821.5)와 최대(1328) 평균 SA를 비교하면 61.7%로 성능에 상당한 차이가 있었다.

compression\_type 기준으로 평균 RATE 범위는 최소 36.57에서 최대 80.16으로 RocksDB 최적화에 중요한 것으로 볼 수 있다.

【0109】반면에, 가장 낮은 파라미터인 mmap\_write와 target\_file\_size\_multiplier에 대한 최대 및 최소 WAF와 SA의 평균 차이는 각각 3.77%와 0.51%이며, 이들은 RocksDB의 성능 향상과 큰 관련이 없다고 볼 수 있다.

【0110】본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법에 따른, 유전자 알고리즘 최적화 솔루션(Genetic Algorithm Optimization Solution)에 대해 설명한다. 예를 들어, 도 4에서 설명된 유전자 알고리즘이 적용될 수 있다.

【0111】랜덤 포레스트 분석을 통해 기여도가 가장 높은 RocksDB 상위 5개 파라미터를 선택할 수 있다. 선택한 파라미터 중, thread는 제거되었는데, 이는 사용자 수에 따라 변경될 수 있기 때문이다. 결과적으로, 4개의 파라미터(즉, compression\_type, level0\_slowdown\_writes\_trigger, write\_buffer\_size, compression\_ratio)에 유전자 알고리즘을 적용하였다.

【0112】초기 세대로서 8개의 랜덤 염색체가 정의될 수 있다. 즉, 각 세대에는 항상 8개의 염색체가 있을 수 있다. 최상의 염색체가 요구 사항을 충족하지 못하는 경우, 하위 4개의 염색체는 각 반복(iteration)에서 폐기될 수 있다.

【0113】 4개의 염색체를 보충하기 위해, 돌연변이가 없는 부모 염색체의 교배에 의해 2개의 염색체를 생성하고, 완전 돌연변이(complete mutations)(즉, 신 염색체)를 통해 나머지 2개의 염색체를 생성할 수 있다. 즉, 알고리즘이 초기 세대의 영향을 덜 받게 하기 위해, 2개의 염색체를 완전 돌연변이 시킬 수 있다.

【0114】 도 8은 본 발명의 몇몇 기계 학습 기반 데이터베이스 튜닝 방법에 따른, 유전자 알고리즘의 수렴 과정을 나타낸다.

【0115】 유전자 알고리즘은 처음 두 세대에서, 높은 적합도 점수를 갖지 않을 수 있다. 그러나, 이전 단계의 잠재적으로 좋은 유전자는 계속 살아 남고, 결과적으로 7번째 세대에서 약 1600의 높은 적합도 점수를 얻을 수 있다.

【0116】 따라서, 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법은 짧은 시간에 파라미터들의 최적의 값을 찾을 수 있다. 즉, 이는 본 명세서에서 제안된 파이프라인(즉, 본 발명에 따른 기계 학습 기반 데이터베이스 튜닝 방법)이 RocksDB에 대한 파라미터 튜닝의 효율적인 방법이라는 것을 나타낸다.

【0117】 【표 2】

Indicator	TIME (s)	RATE (MB/s)	SA (MB)	WAF
Default Setting	<b>63.8</b>	15.5	432.94	9.2
Genetic Algorithm	67.4	<b>29.4</b>	<b>197</b>	<b>7</b>

【0118】 표 2는 기본 파라미터를 사용한 RocksDB 및 유전자 알고리즘을 통해 최적화된 파라미터를 사용한 RocksDB의 성능을 나타낸다. 구체적으로, 표 2는 유전자 알고리즘을 이용하여 상위 4개 파라미터의 최적 값을 적용한 RocksDB에 대한 성능을 계산한 결과를 나타낸다.

【0119】 표2를 참조하면, 파라미터의 최적 값을 적용한 데이터베이스가 기본 설정 값을 적용한 데이터베이스보다, RATE, SA 및 WAF과 관련하여 더 나은 성능을 달성했음을 알 수 있다.

【0120】 도 9는 본 발명의 기계 학습 기반 데이터베이스 튜닝 방법에 의해 최적화된 RocksDB의 향상된 성능을 나타낸다. 즉, 도 9는 기본 설정 파라미터와 비교하여, 유전자 알고리즘을 통해 얻은 최적 파라미터의 개선 정도를 보여준다.

【0121】 도 9를 참조하면, TIME이 약간 감소(-5.64%)되었지만, 파라미터 WAF, SA 및 RATE는 각각 20.65%, 54.50% 및 89.68%으로 크게 향상된 것을 볼 수 있다.

【0122】 결과적으로, 도 9는 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법을 적용한 RocksDB는 불필요한 쓰기 작업과 공간 점유가 제거되었으며, 지연된 실행 시간 없이 높은 데이터 처리 속도를 달성할 수 있다는 점을 볼 수 있다.

【0123】 본 발명의 몇몇 실시예에 따르면, 본 발명에서 제안한 기계 학습 기반 데이터베이스 튜닝 방법은 LSM-Tree를 사용하는 다른 데이터베이스의 파라미터

를 최적화하기 위해 사용될 수 있다.

【0124】 이하, 도 10 및 도 11를 참조하여, 본 발명의 몇몇 실시예에 따른, 기계 학습 기반 데이터베이스 튜닝 방법 및 장치에 대해 설명한다.

【0125】 도 10은 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 방법의 흐름도이다. 예를 들어, 기계 학습 기반 데이터베이스 튜닝 방법은 이하의 도 11에 개시된 데이터베이스 튜닝 장치에 의해 수행될 수 있다.

【0126】 단계 S1001에서, 상기 데이터베이스 튜닝 장치는 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석할 수 있다.

【0127】 예를 들어, 상기 데이터베이스는 RocksDB일 수 있다.

【0128】 예를 들어, 상기 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계는, 랜덤 포레스트 (random forest) 기법을 사용하여 상기 데이터베이스와 관련된 상기 복수의 파라미터의 기여도를 분석하는 단계를 포함할 수 있다.

【0129】 예를 들어, 상기 랜덤 포레스트 기법을 사용하여 상기 데이터베이스와 관련된 상기 복수의 파라미터의 기여도를 분석하는 단계는, (1) 상기 데이터베이스로부터 훈련 데이터 세트를 수신하는 단계, (2) 상기 훈련 데이터 세트로부터 복수의 서브 데이터 세트를 생성하는 단계, (3) 상기 복수의 서브 데이터 세트 각각에 대한 결정 트리(decision tree)를 생성하는 단계, 및 (4) 상기 생성된 결정 트리를 이용하여 상기 데이터베이스와 관련된 상기 복수의 파라미터의 기여도를 분

석하는 단계를 더 포함할 수 있다.

【0130】 예를 들어, 상기 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계는, 상기 데이터베이스의 쓰기 증폭 및 공간 증폭에 대한 상기 복수의 파라미터의 기여도를 분석하는 단계를 포함할 수 있다.

【0131】 단계 S1002에서, 상기 데이터베이스 튜닝 장치는 상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택할 수 있다.

【0132】 예를 들어, 상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택하는 단계는, (1) 상기 복수의 파라미터들을 기여도가 높은 순서대로 정렬하는 단계, 및 (2) 상기 정렬된 복수의 파라미터들 중 가장 높은 기여도를 갖는 파라미터부터 N 번째 높은 기여도를 갖는 파라미터까지의, N 개의 파라미터를 선택하는 단계를 포함할 수 있다. 이때, 상기 N은 1이상의 자연수일 수 있다.

【0133】 예를 들어, 상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택하는 단계는, 상기 복수의 파라미터들 중 미리 정의된 임계 값 이상의 기여도를 갖는 하나 이상의 파라미터를 선택하는 단계를 포함할 수 있다.

【0134】 예를 들어, 상기 선택된 하나 이상의 파라미터는 (i) min\_level\_to\_compress, (ii) compression\_type, (iii)

level0\_slowdown\_writes\_trigger, (iv) write\_buffer\_size, 및 (v) compression\_ratio 중 적어도 하나를 포함할 수 있다.

【0135】 예를 들어, 상기 선택된 하나 이상의 파라미터는 thread를 포함하지 않을 수 있다.

【0136】 단계 S1003에서, 상기 데이터베이스 튜닝 장치는 분산 분석(ANOVA)을 이용하여 상기 선택된 하나 이상의 파라미터들의 기여도를 검증할 수 있다.

【0137】 예를 들어, 상기 데이터베이스 튜닝 장치는 분산 분석을 통해 단계 S1002에서 선택된 파라미터들이 데이터베이스에 대한 기여도가 높다는 것을 검증할 수 있다.

【0138】 단계 S1004에서, 상기 데이터베이스 튜닝 장치는 유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득할 수 있다. 예를 들어, 상기 최적화된 값은 상기 데이터베이스에 적용될 수 있다.

【0139】 예를 들어, 상기 유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득하는 단계는, (1) 복수의 염색체(chromosome)를 포함하는 초기 세대(initial population)를 구성하되, 상기 초기 세대는 복수의 염색체를 포함하고, 상기 복수의 염색체 각각은 상기 선택된 하나 이상의 파라미터들 각각에 대한 랜덤 값을 갖는 단계, (2) 상기 초기 세대에 피트니스 함수를 적용하여, 상기 초기 세대에 포함된 상기 복수의 염색체들의 순위를 결정하는 단계, (3) 상기 초기 세대에서 가장 높은 순위를 갖는 염색체가 상기 유



전자 알고리즘의 요구 사항을 만족하는지 여부를 결정하는 단계, 및 (4) 상기 초기 세대에서 가장 높은 순위를 갖는 염색체가 상기 유전자 알고리즘의 상기 요구 사항을 만족시키지 못하는 경우, 다음 세대(next generation)를 구성하는 단계를 포함할 수 있다.

【0140】 예를 들어, 상기 유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득하는 단계는, (1) 상기 다음 세대에 상기 피트니스 함수를 적용하여, 상기 다음 세대에 포함된 복수의 염색체들의 순위를 결정하는 단계, (2) 상기 다음 세대에서 가장 높은 순위를 갖는 염색체가 상기 유전자 알고리즘의 요구 사항을 만족하는지 여부를 결정하는 단계, 및 (3) 상기 다음 세대에서 가장 높은 순위를 갖는 염색체가 상기 유전자 알고리즘의 상기 요구 사항을 만족시키는 경우, 상기 다음 세대에서 가장 높은 순위를 갖는 염색체에 포함된 하나 이상의 파라미터들의 값을, 상기 선택된 하나 이상의 파라미터들에 대한 상기 최적화된 값으로 결정하는 단계를 더 포함할 수 있다.

【0141】 이 경우, 상기 다음 세대(next generation)를 구성하는 단계는, (1) 상기 초기 단계의 복수의 염색체들 중 낮은 우선 순위를 갖는 염색체들을 제거하는 단계, (2) 상기 초기 단계의 복수의 염색체들 중 높은 우선 순위를 갖는 염색체들로부터 교배된 염색체를 생성하는 단계, 및 (3) 상기 초기 단계의 복수의 염색체들 중 높은 우선 순위를 갖는 염색체들로부터 돌연변이 염색체를 생성하는 단계를 포함할 수 있다.

【0142】 도 11은 본 발명의 몇몇 실시예에 따른 기계 학습 기반 데이터베이스 튜닝 장치의 개념도이다.

【0143】 도 11을 참조하면, 기계 학습 기반 데이터베이스 튜닝 장치(1000)는 메모리(1200) 및 프로세서(1400)를 포함할 수 있다.

【0144】 메모리(1200) 및 프로세서(1400)는 각각 별도의 칩으로 구현되거나, 하나의 칩을 통해 구현될 수 있다. 메모리(1200) 및 프로세서(1400)는 서로 유기적으로 결합되어 작동될 수 있다. 예를 들어, 프로세서(1400)는 메모리(1200)에 저장된 데이터를 사용할 수 있고, 프로세서(1400)에서 출력된 데이터는 다시 메모리(1200)에 저장될 수 있다. 또한, 메모리(1200)는, 휘발성 및/또는 비휘발성 메모리를 포함할 수 있다. 메모리(1200)는, 프로세서(1400)에 의해 실행되는 명령들(instructions) 또는 프로그램을 저장할 수 있다. 또한, 프로세서(1400)는, 소프트웨어를 구동하여 프로세서(1400)에 연결된 데이터베이스 튜닝 장치(1000)를 제어할 수 있다. 또한, 프로세서(1400)는 본 발명과 관련된 다양한 연산, 처리, 데이터 생성, 가공 등의 동작을 수행할 수 있다.

【0145】 예를 들어, 데이터베이스 튜닝 장치(1000)의 프로세서(1400)는 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계, 상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택하는 단계, 및 유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득하는 단계를 수행하도록 구성될 수 있다.

【0146】 또한, 상술한 바와 같은 기계 학습 기반 데이터베이스 튜닝 방법은

컴퓨터에서 실행될 수 있는 실행가능한 알고리즘을 포함하는 프로그램(또는 어플리케이션)으로 구현될 수 있다. 상기 프로그램은 일시적 또는 비일시적 판독 가능 매체(non-transitory computer readable medium)에 저장되어 제공될 수 있다.

【0147】 비일시적 판독 가능 매체란 레지스터, 캐쉬, 메모리 등과 같이 짧은 순간 동안 데이터를 저장하는 매체가 아니라 반영구적으로 데이터를 저장하며, 기기에 의해 판독(reading)이 가능한 매체를 의미한다. 구체적으로는, 상술한 다양한 어플리케이션 또는 프로그램들은 CD, DVD, 하드 디스크, 블루레이 디스크, USB, 메모리카드, ROM (read-only memory), PROM (programmable read only memory), EPROM(Erasable PROM, EPROM) 또는 EEPROM(Electrically EPROM) 또는 플래시 메모리 등과 같은 비일시적 판독 가능 매체에 저장되어 제공될 수 있다.

【0148】 일시적 판독 가능 매체는 스테틱 램(Static RAM, SRAM), 다이내믹 램(Dynamic RAM, DRAM), 싱크로너스 디램 (Synchronous DRAM, SDRAM), 2배속 SDRAM(Double Data Rate SDRAM, DDR SDRAM), 증강형 SDRAM(Enhanced SDRAM, ESDRAM), 동기화 DRAM(Synclink DRAM, SLDRAM) 및 직접 램버스 램(Direct Rambus RAM, DRRAM) 과 같은 다양한 RAM을 의미한다.

【0149】 본 실시예 및 본 명세서에 첨부된 도면은 전술한 기술에 포함되는 기술적 사상의 일부를 명확하게 나타내고 있는 것에 불과하며, 전술한 기술의 명세서 및 도면에 포함된 기술적 사상의 범위 내에서 당업자가 용이하게 유추할 수 있는 변형 예와 구체적인 실시예는 모두 전술한 기술의 권리범위에 포함되는 것이 자명하다고 할 것이다.

**【청구범위】****【청구항 1】**

데이터베이스 튜닝 방법으로서,

데이터베이스 튜닝 장치에서 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계;

상기 데이터베이스 튜닝 장치에서 상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택하는 단계; 및

상기 데이터베이스 튜닝 장치에서 유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득하는 단계를 포함하는 것을 특징으로 하는 데이터베이스 튜닝 방법.

**【청구항 2】**

제1 항에 있어서,

상기 데이터베이스는 RocksDB인 것을 특징으로 하는 데이터베이스 튜닝 방법.

**【청구항 3】**

제1 항에 있어서,

상기 데이터베이스 튜닝 장치에서 분산 분석(ANOVA)을 이용하여 상기 선택된 하나 이상의 파라미터들의 기여도를 검증하는 단계를 더 포함하는 것을 특징으로 하는 데이터베이스 튜닝 방법.

**【청구항 4】**

제1 항에 있어서, 상기 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계는,

랜덤 포레스트 (random forest) 기법을 사용하여 상기 데이터베이스와 관련된 상기 복수의 파라미터의 기여도를 분석하는 단계를 포함하는 것을 특징으로 하는 데이터베이스 튜닝 방법.

**【청구항 5】**

제4 항에 있어서, 상기 랜덤 포레스트 기법을 사용하여 상기 데이터베이스와 관련된 상기 복수의 파라미터의 기여도를 분석하는 단계는,

상기 데이터베이스로부터 훈련 데이터 세트를 수신하는 단계;

상기 훈련 데이터 세트로부터 복수의 서브 데이터 세트를 생성하는 단계;

상기 복수의 서브 데이터 세트 각각에 대한 결정 트리(decision tree)를 생성하는 단계; 및

상기 생성된 결정 트리를 이용하여 상기 데이터베이스와 관련된 상기 복수의 파라미터의 기여도를 분석하는 단계를 더 포함하는 것을 특징으로 하는 데이터베이스 튜닝 방법.

**【청구항 6】**

제1 항에 있어서, 상기 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계는,

상기 데이터베이스의 쓰기 증폭 및 공간 증폭에 대한 상기 복수의 파라미터의 기여도를 분석하는 단계를 포함하는 것을 특징으로 하는 데이터베이스 튜닝 방법.

### 【청구항 7】

제1 항에 있어서, 상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택하는 단계는,

상기 복수의 파라미터들을 기여도가 높은 순서대로 정렬하는 단계; 및

상기 정렬된 복수의 파라미터들 중 가장 높은 기여도를 갖는 파라미터부터 N 번째 높은 기여도를 갖는 파라미터까지의, N 개의 파라미터를 선택하는 단계를 포함하고,

상기 N은 1이상의 자연수인 것을 특징으로 하는 데이터베이스 튜닝 방법.

### 【청구항 8】

제1 항에 있어서, 상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택하는 단계는,

상기 복수의 파라미터들 중 미리 정의된 임계 값 이상의 기여도를 갖는 하나 이상의 파라미터를 선택하는 단계를 포함하는 것을 특징으로 하는 데이터베이스 튜닝 방법.

### 【청구항 9】

제1 항에 있어서,

상기 선택된 하나 이상의 파라미터는 (i) min\_level\_to\_compress, (ii) compression\_type, (iii) level0\_slowdown\_writes\_trigger, (iv) write\_buffer\_size, 및 (v) compression\_ratio 중 적어도 하나를 포함하는 것을 특징으로 하는 데이터베이스 튜닝 방법.

### 【청구항 10】

제1 항에 있어서,

상기 선택된 하나 이상의 파라미터는 thread를 포함하지 않는 것을 특징으로 하는 데이터베이스 튜닝 방법.

### 【청구항 11】

제1 항에 있어서, 상기 유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득하는 단계는,

복수의 염색체(chromosome)를 포함하는 초기 세대(initial population)를 구성하되, 상기 초기 세대는 복수의 염색체를 포함하고, 상기 복수의 염색체 각각은 상기 선택된 하나 이상의 파라미터들 각각에 대한 랜덤 값을 갖는 단계;

상기 초기 세대에 피트니스 함수를 적용하여, 상기 초기 세대에 포함된 상기 복수의 염색체들의 순위를 결정하는 단계;

상기 초기 세대에서 가장 높은 순위를 갖는 염색체가 상기 유전자 알고리즘의 요구 사항을 만족하는지 여부를 결정하는 단계; 및

상기 초기 세대에서 가장 높은 순위를 갖는 염색체가 상기 유전자 알고리즘

의 상기 요구 사항을 만족시키지 못하는 경우, 다음 세대(next generation)를 구성하는 단계를 포함하는 것을 특징으로 하는 데이터베이스 튜닝 방법.

### 【청구항 12】

제11 항에 있어서, 유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득하는 단계는,

상기 다음 세대에 상기 피트니스 함수를 적용하여, 상기 다음 세대에 포함된 복수의 염색체들의 순위를 결정하는 단계;

상기 다음 세대에서 가장 높은 순위를 갖는 염색체가 상기 유전자 알고리즘의 요구 사항을 만족하는지 여부를 결정하는 단계; 및

상기 다음 세대에서 가장 높은 순위를 갖는 염색체가 상기 유전자 알고리즘의 상기 요구 사항을 만족시키는 경우, 상기 다음 세대에서 가장 높은 순위를 갖는 염색체에 포함된 하나 이상의 파라미터들의 값을, 상기 선택된 하나 이상의 파라미터들에 대한 상기 최적화된 값으로 결정하는 단계를 더 포함하는 것을 특징으로 하는 데이터베이스 튜닝 방법.

### 【청구항 13】

제11 항에 있어서, 상기 다음 세대(next generation)를 구성하는 단계는,

상기 초기 단계의 복수의 염색체들 중 낮은 우선 순위를 갖는 염색체들을 제거하는 단계;

상기 초기 단계의 복수의 염색체들 중 높은 우선 순위를 갖는 염색체들로부터



터 교배된 염색체를 생성하는 단계; 및

상기 초기 단계의 복수의 염색체들 중 높은 우선 순위를 갖는 염색체들로부터 돌연변이 염색체를 생성하는 단계를 포함하는 것을 특징으로 하는 데이터베이스 튜닝 방법.

#### 【청구항 14】

프로세서; 및

상기 프로세서와 결합되어 작동되는 메모리를 포함하는 데이터베이스 튜닝 장치로서, 상기 프로세서는,

기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계;

상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택하는 단계; 및

유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득하는 단계를 수행하도록 구성된,

데이터베이스 튜닝 장치.

#### 【청구항 15】

제14 항에 있어서,

상기 데이터베이스는 RocksDB인 것을 특징으로 하는 데이터베이스 튜닝 장치.

**【청구항 16】**

제14 항에 있어서, 상기 프로세서는,

분산 분석(ANOVA)을 이용하여, 상기 선택된 하나 이상의 파라미터들의 기여도를 검증하는 단계를 더 수행하도록 구성된,

데이터베이스 튜닝 장치.

**【청구항 17】**

제14 항에 있어서, 상기 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계는,

랜덤 포레스트 (random forest) 기법을 사용하여 상기 데이터베이스와 관련된 상기 복수의 파라미터의 기여도를 분석하는 단계를 포함하는 것을 특징으로 하는 데이터베이스 튜닝 장치.

**【요약서】****【요약】**

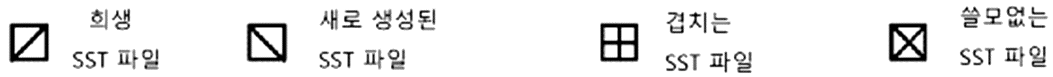
데이터베이스 튜닝 방법 및 장치가 제공된다. 데이터 베이스 튜닝 방법은 기계 학습 기반으로 데이터베이스와 관련된 복수의 파라미터의 기여도를 분석하는 단계, 상기 분석된 기여도에 기초하여 상기 복수의 파라미터들 중 하나 이상의 파라미터를 선택하는 단계, 및 유전자 알고리즘을 이용하여 상기 선택된 하나 이상의 파라미터들 각각에 대한 최적화된 값을 획득하는 단계를 포함하는 것을 특징으로 한다.

**【대표도】**

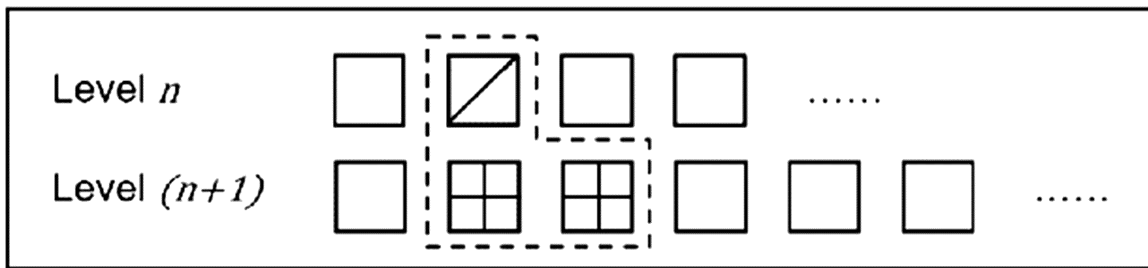
도 2

## 【도면】

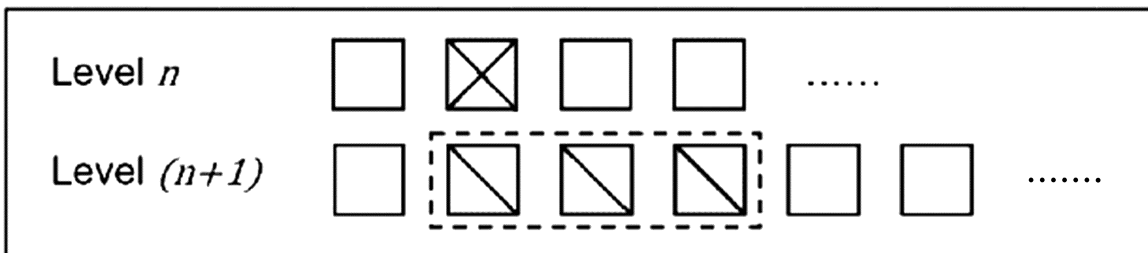
【도 1】



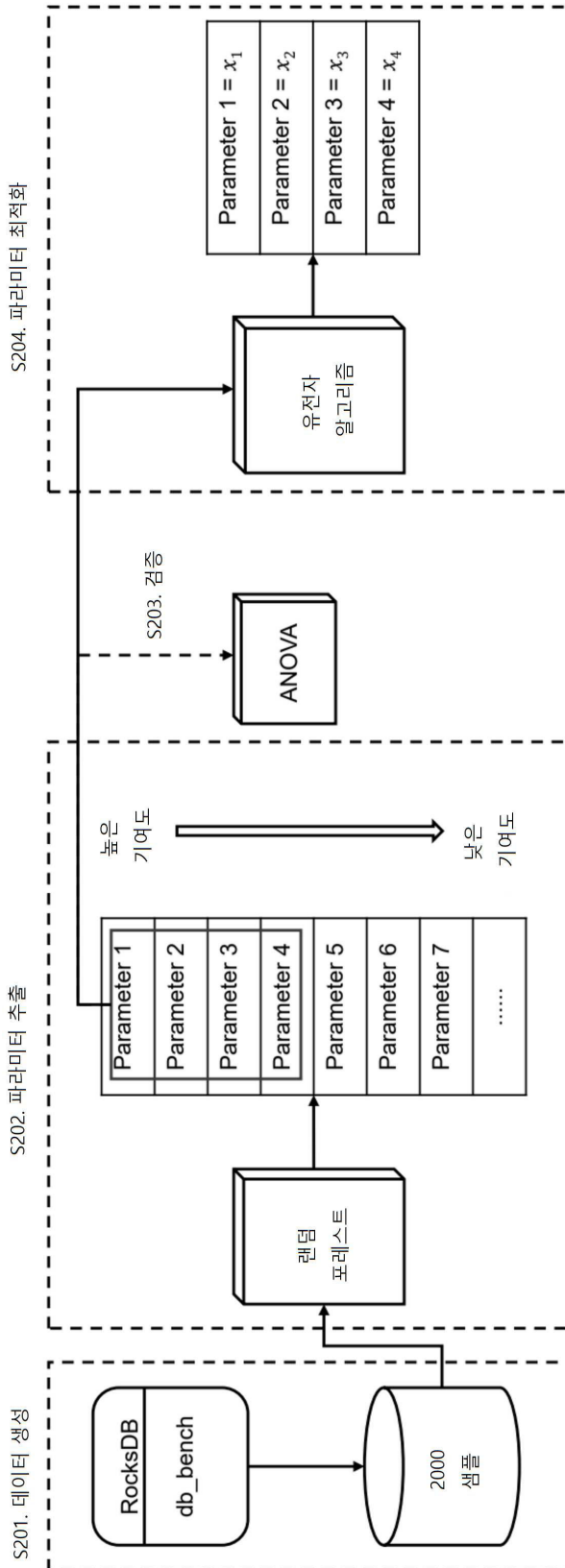
(a) 컴팩션 이전



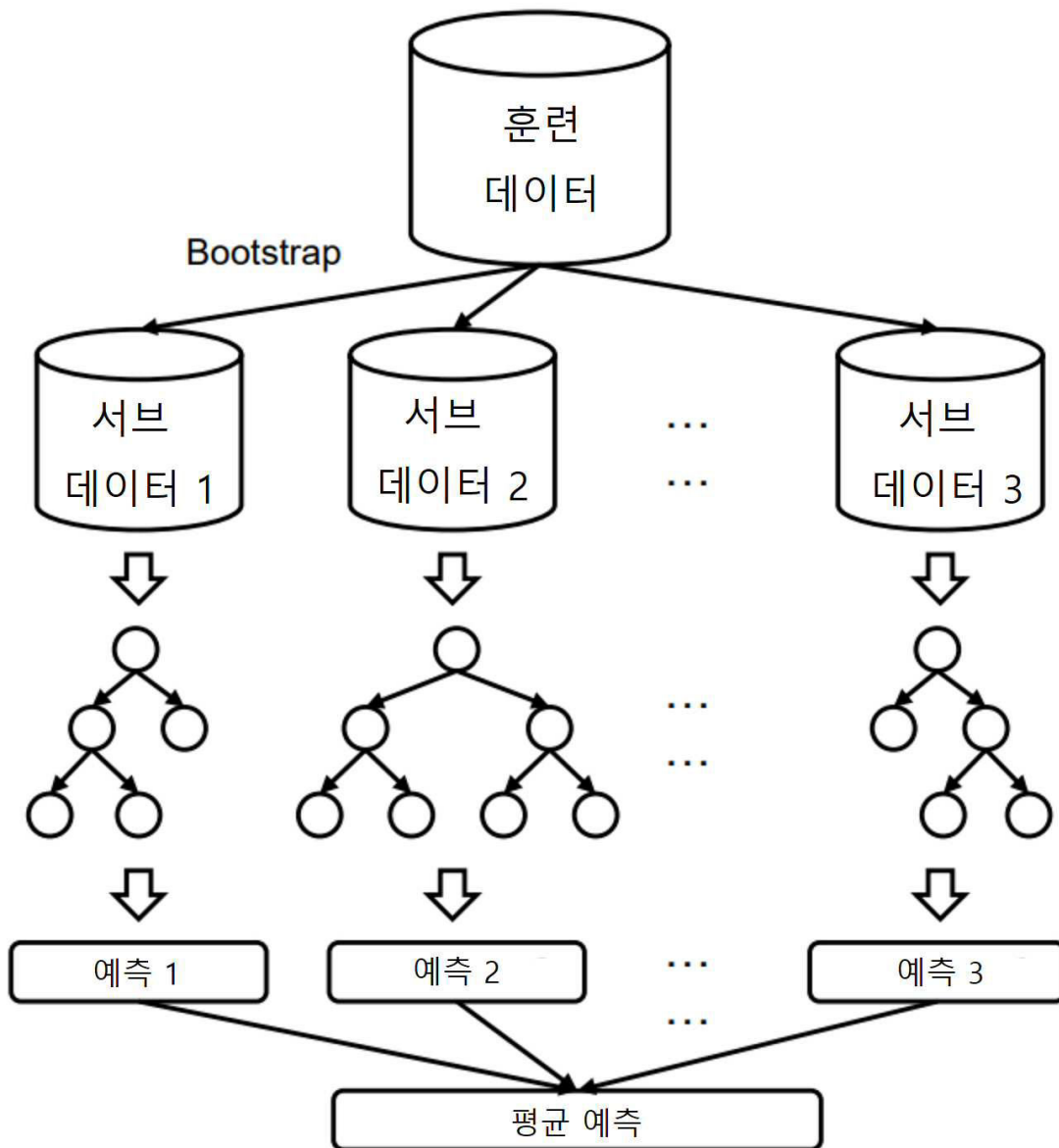
(b) 컴팩션 이후



【도 2】

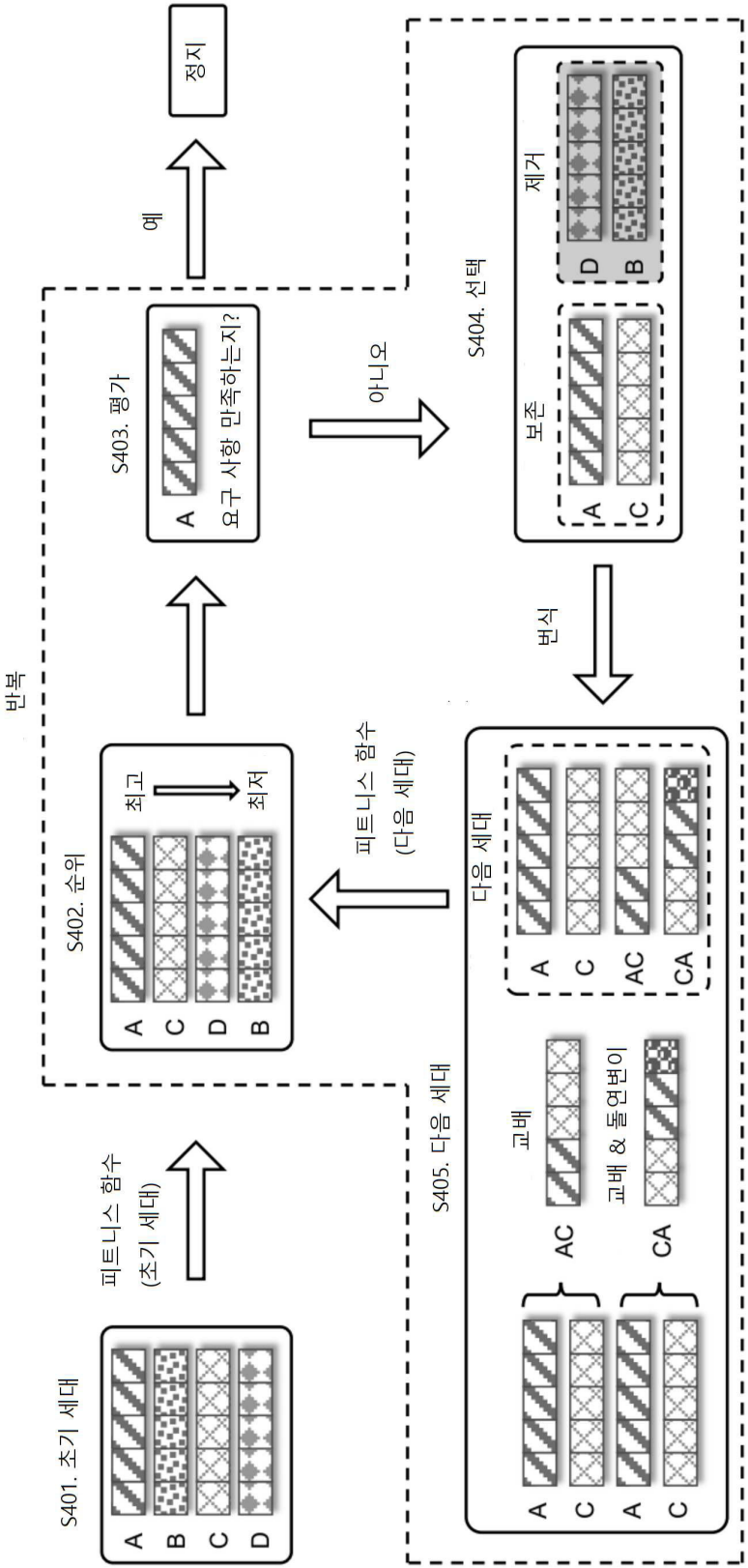


【도 3】

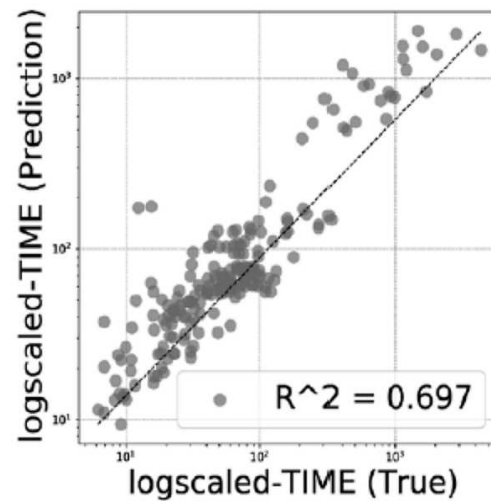
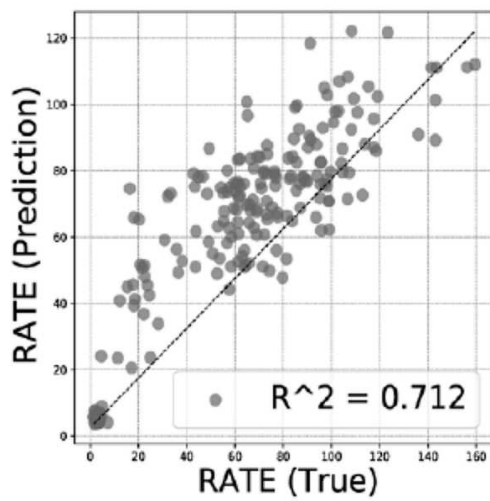
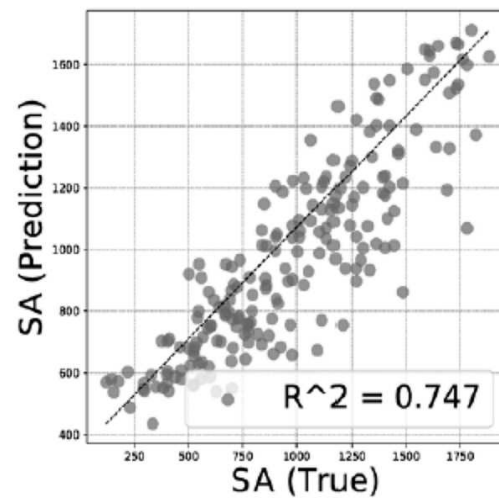
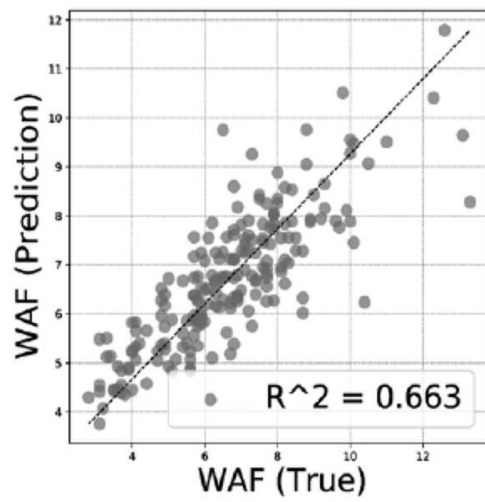


【도 4】

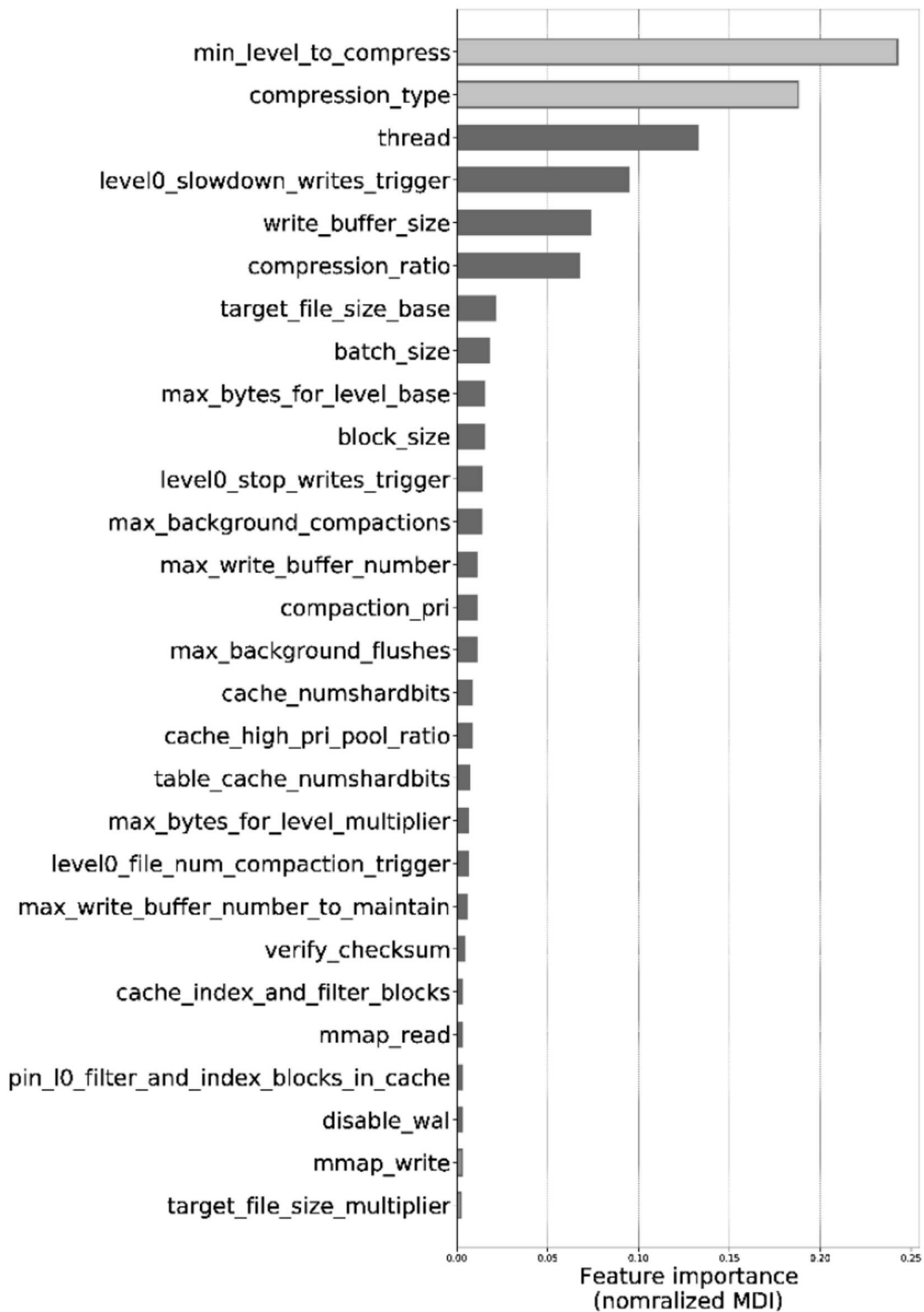




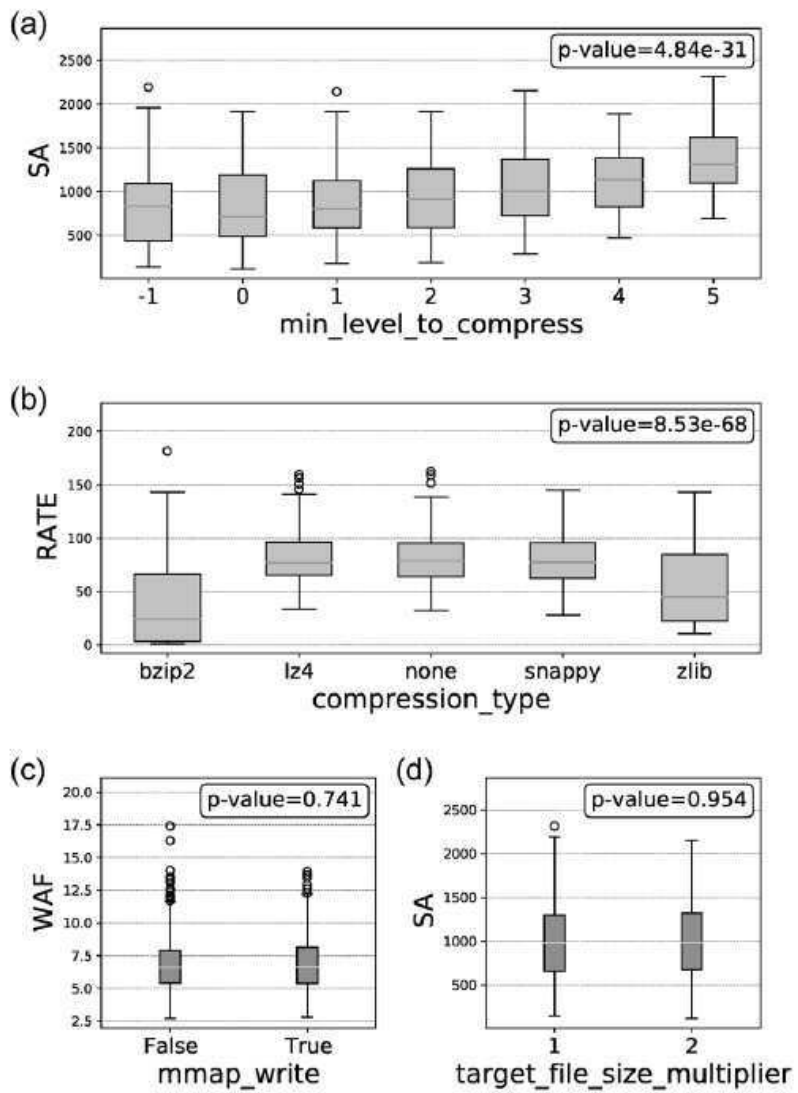
【도 5】



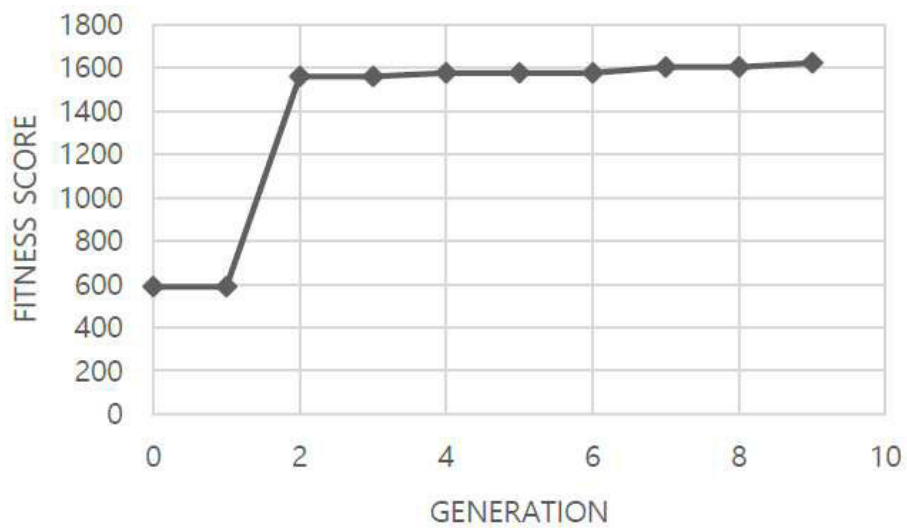
【도 6】



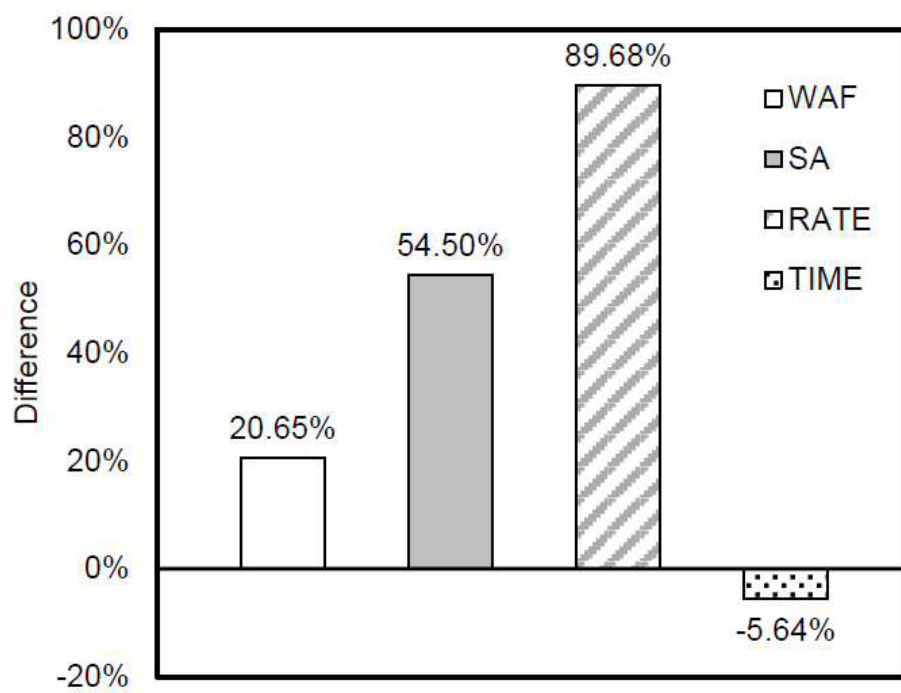
【도 7】



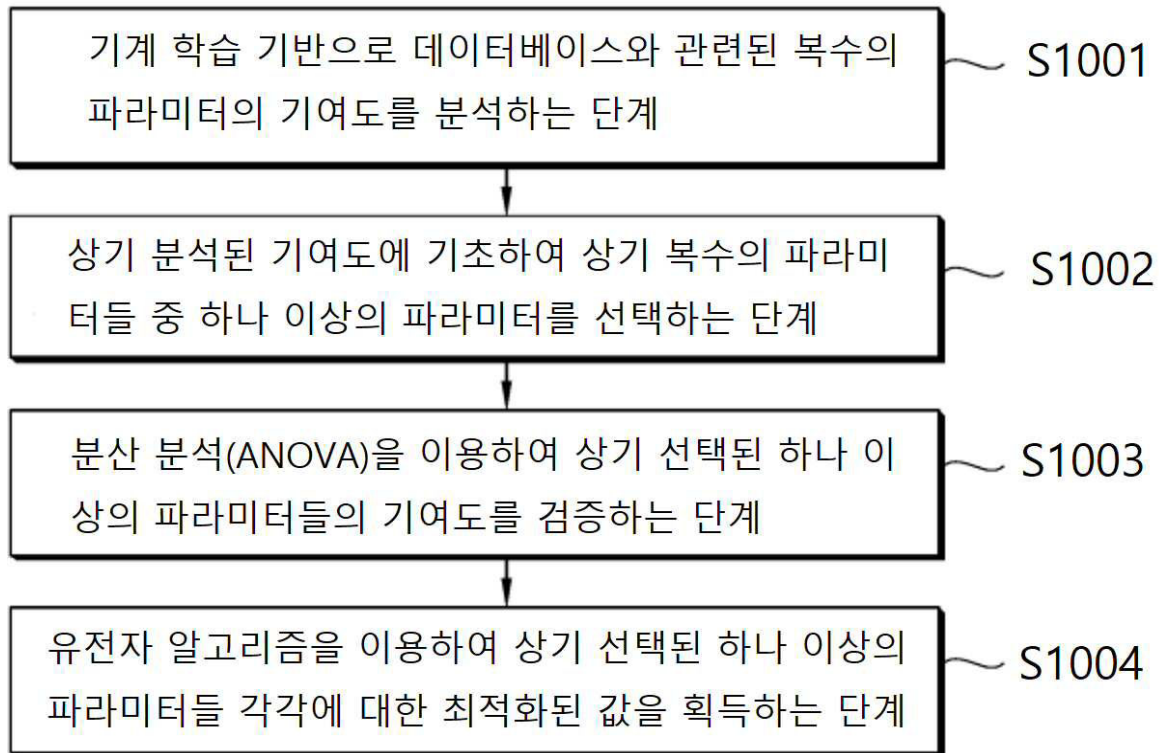
【도 8】



【도 9】



【도 10】



【도 11】

