



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0092998
(43) 공개일자 2021년07월27일

(51) 국제특허분류(Int. Cl.)

G05B 23/02 (2006.01)

(52) CPC특허분류

G05B 23/0221 (2013.01)

G05B 23/0243 (2013.01)

(21) 출원번호 10-2020-0006613

(22) 출원일자 2020년01월17일

심사청구일자 2020년01월17일

(71) 출원인

한림대학교 산학협력단

강원도 춘천시 한림대학길 1, 한림대학교(옥천동)

(72) 발명자

김의직

강원도 춘천시 지석로 29, 516동 1303호 (석사동, 휴먼타운아파트)

권정혁

강원도 춘천시 스무숲1길 32-9, 202호

(74) 대리인

김남혁

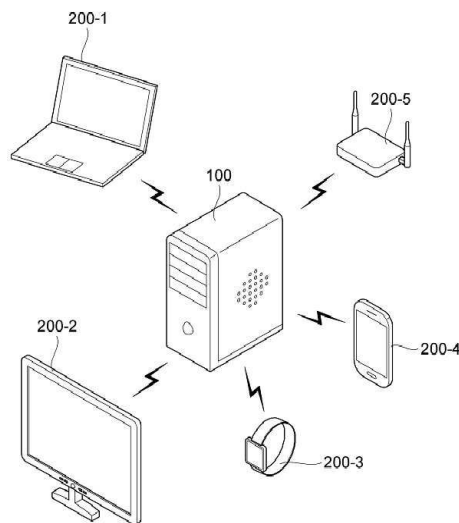
전체 청구항 수 : 총 9 항

(54) 발명의 명칭 산업용 사물인터넷(IIoT) 환경에서의 고장 예측 모델 획득을 위한 전처리 방법, 피처의 중요도 측정 방법 및 피처 선택 방법, 장치 및 프로그램

(57) 요약

산업용 사물인터넷(IIoT) 환경에서의 고장 예측 모델 획득 시스템의 제어 방법이 개시된다. 상기 제어 방법은, 서버가, 복수의 센서로부터 데이터를 획득하는 단계, 상기 서버가, 상기 획득된 데이터를 전처리하는 단계, 상기 서버가, 상기 전처리된 데이터를 바탕으로 피처의 중요도를 획득하는 단계 및 상기 서버가, 상기 획득된 중요도를 바탕으로 피처를 선택하는 단계를 포함한다.

대 표 도 - 도1



(52) CPC특허분류

G05B 23/0283 (2013.01)

G06N 20/00 (2019.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	NRF-2019R1I1A1A01059787
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	학문균형발전지원사업(창의도전연구)
연구과제명	전원 무구속 사물인터넷을 위한 저전력 고효율 무선 정보 및 전력 동시전송 프레임
워크 개발	
기 여 율	1/1
과제수행기관명	한림대학교 산학협력단
연구기간	2019.06.01 ~ 2020.02.29

명세서

청구범위

청구항 1

산업용 사물인터넷(IIoT) 환경에서의 고장 예측 모델 획득 시스템의 제어 방법에 있어서,
서버가, 복수의 센서로부터 데이터를 획득하는 단계;
상기 서버가, 상기 획득된 데이터를 전처리하는 단계;
상기 서버가, 상기 전처리된 데이터를 바탕으로 피치의 중요도를 획득하는 단계; 및
상기 서버가, 상기 획득된 중요도를 바탕으로 피치를 선택하는 단계; 를 포함하는 제어 방법.

청구항 2

제1항에 있어서,
상기 전처리하는 단계는,
상기 복수의 센서로부터 획득된 데이터 각각에 대응되는 복수의 피치 중 적어도 하나의 피치를 제거하는 단계;
상기 적어도 하나의 피치가 제거된 복수의 피치에 대한 데이터를 정규화 하는 단계; 및
상기 정규화된 데이터를 분할하는 단계; 를 포함하는 것을 특징으로 하는 제어 방법.

청구항 3

제2항에 있어서,
상기 피치를 제거하는 단계는,
상기 복수의 센서 각각으로부터 획득한 데이터 중, 특정 피치에 대한 비유효 데이터를 획득하는 단계;
상기 비유효 데이터의 비율이 [0,1]의 범위에서 기 설정된 유효성 계수보다 큰 경우, 상기 비유효 데이터에 대응되는 피치를 제거하는 단계; 를 포함하고,
상기 피치를 제거하는 단계는,
상기 복수의 센서 각각으로부터 획득한 데이터에 대한 각각의 분산값을 획득하는 단계;
상기 분산값이 기 설정된 값 이하인 경우, 상기 기 설정된 값 이하인 분산값에 대응되는 피치를 제거하는 단계; 를 포함하는 제어 방법.

청구항 4

제2항에 있어서,
상기 정규화 하는 단계는,
하기 수식 1을 바탕으로 정규화 되는 것을 특징으로 하는 제어 방법.

[수식 1]

$$xi'=(xi-m)/s$$

이때, xi' 는 $(i + 1)$ 번째 정규화 된 특성 데이터, xi 는 피치의 $(i + 1)$ 번째 데이터, m 은 피치의 평균, s 는 피치의 표준 편차.

청구항 5

제1항에 있어서,

상기 피처의 중요도를 획득하는 단계는,

상기 전처리 단계를 통해 획득된 데이터 중, 학습 데이터 세트를 획득하는 단계; 및

상기 학습 데이터를 랜덤 포레스트 기법을 바탕으로, 상기 학습 데이터에 대응되는 피처의 중요도를 획득하는 단계; 를 포함하는 제어 방법.

청구항 6

제1항에 있어서,

상기 피처를 선택하는 단계는,

하기 알고리즘 1을 바탕으로 피처를 선택하는 단계; 를 포함하는 제어 방법.

[알고리즘 1]

```

1: INITIALIZE  $sf_{cnt}$  to NULL,  $pa_{cnt}$  to 0, and  $cnt$  to 0 // Initialize variables
2: REPEAT // Start iteration
3:   /* ===== Feature Selection Step ===== */
4:    $maxImp \leftarrow \max(IMP)$  // Find the highest importance from IMP
5:   FOR each feature index,  $j$ ,  $j \in [0, nf - 1]$ 
6:     IF  $IMP[j] == maxImp$  // Find feature index with the highest importance
7:        $sf_{cnt} \leftarrow F[j]$  // Select a new feature
8:        $F[j], IMP[j] \leftarrow 0$  // Remove the selected feature from IMP and F
9:     ENDIF
10:  ENDFOR
11:   $SF[cnt] \leftarrow sf_{cnt}$  // Update the selected feature set

```

이때, cnt는 카운터 값, sf_{cnt} 는 cnt에서의 선택된 피처, 피처의 중요도 세트는 $IMP=[imp_0, \dots, imp_j, \dots, imp_{nf-1}]$, 피처의 세트는 $F=[f_0, \dots, f_j, \dots, f_{nf-1}]$, imp_j 는, (j+1)번째 피처의 중요도의 값, nf는 피처의 수.

청구항 7

제1항에 있어서,

상기 제어 방법은,

상기 선택된 피처를 바탕으로 복수의 모델을 생성하는 단계; 및

상기 생성된 복수의 모델 중, 적어도 하나의 모델을 선택하는 단계; 를 더 포함하고,

상기 복수의 모델을 생성하는 단계는, 하기 알고리즘 2에 의해 생성되는 것을 특징으로 하는 제어 방법.

[알고리즘 2]

```

1: INITIALIZE  $sf_{cnt}$  to NULL,  $pa_{cnt}$  to 0, and  $cnt$  to 0 // Initialize variables
2: REPEAT // Start iteration
3:   /*===== Feature Selection Step =====*/
4:    $maxImp \leftarrow \max(IMP)$  // Find the highest importance from IMP
5:   FOR each feature index,  $j$ ,  $j \in [0, nf - 1]$ 
6:     IF  $IMP[j] = maxImp$  // Find feature index with the highest importance
7:        $sf_{cnt} \leftarrow F[j]$  // Select a new feature
8:        $F[j], IMP[j] \leftarrow 0$  // Remove the selected feature from IMP and F
9:     ENDIF
10:  ENDFOR
11:   $SF[cnt] \leftarrow sf_{cnt}$  // Update the selected feature set
12:  /*===== Model Building Step =====*/
13:   $pa_{cnt} \leftarrow \text{modelBuildEvalFunction}(SF)$  // build and evaluate model
14:   $PA[cnt] \leftarrow pa_{cnt}$  // Update the prediction accuracy set
15:   $cnt \leftarrow cnt + 1$  // Increment iteration counter by one
16: UNTIL  $cnt \leq max$  // Terminate iteration if condition become FALSE
17: RETURN SF and PA // Return selected feature set and prediction accuracy set

```

이때, cnt 는 카운터 값, SF는 선택된 피쳐, PA는 예측 정확도, sf_{cnt} 는 cnt 에서의 선택된 피쳐, 피쳐의 중요도 세트는 $IMP=[imp_0, \dots, imp_j, \dots, imp_{nf-1}]$, 피쳐의 세트는 $F=[f_0, \dots, f_j, \dots, f_{nf-1}]$, imp_j 는, $(j+1)$ 번째 피쳐의 중요도의 값, nf 는 피쳐의 수, $\text{modelBuildEvalFunction}(SF)$ 는 예측 모델을 구축하고 평가하기 위한 임의의 함수.

청구항 8

하나 이상의 인스트럭션을 저장하는 메모리; 및

상기 메모리에 저장된 상기 하나 이상의 인스트럭션을 실행하는 프로세서를 포함하고,

상기 프로세서는 상기 하나 이상의 인스트럭션을 실행함으로써,

제1 항의 방법을 수행하는, 장치.

청구항 9

하드웨어인 컴퓨터와 결합되어, 제1 항의 방법을 수행할 수 있도록 컴퓨터에서 독출가능한 기록매체에 저장된 컴퓨터프로그램.

발명의 설명

기술 분야

[0001]

본 발명은 산업용 사물인터넷(IIoT) 환경에서의 고장 예측 모델(Failure Prediction Model) 획득을 위한 전처리 방법, 피쳐의 중요도 측정 방법 및 피쳐 선택 방법, 장치 및 프로그램에 관한 것이다.

배경 기술

[0002]

최근, 산업용 사물인터넷(IIoT)은 고급 제조의 핵심 기술인 연결 및 분석 기능을 제공하기 때문에 다양한 회사에서 널리 채택되고 있다. IIoT에서는 머신 상태, 제조 공정, 산업 환경 등의 변화를 주기적으로 감지하기 위해 많은 수의 센서가 사용되고, 따라서 IIoT에 사용된 센서에서 많은 양의 데이터가 수집될 수 있다. 일반적으로 수집된 데이터는 생산성 향상에 유용한 정보를 제공하기 위해 분석된다. 특히, 데이터 분석을 통한 장애 예측은 IIoT에서 가장 중요한 문제 중 하나이다. 이는, IIoT 시스템이 고장날 경우, 고장으로 인해 치명적인 시스템

오작동과 유지 보수 비용이 발생하여 생산성이 저하되기 때문이다.

- [0003] 따라서, 근래에는 IIoT 시스템의 고장을 예측하기 위한 고장 예측 모델을 구축하기 위한 다양한 연구가 이뤄지고 있다. 그러나, 종래에는, 복수의 센서로부터 수집된 데이터를 바탕으로 고장 예측 모델을 구축하는 방법들이 연구되고 있으나, 복수의 센서로부터 수집된 데이터를 어떻게 선별할 것인지에 대한 연구가 미비한 실정이다.

선행기술문헌

특허문헌

- [0004] (특허문헌 0001) 등록특허공보 제10-1538709호, 2015.07.16

발명의 내용

해결하려는 과제

- [0005] 본 발명이 해결하고자 하는 과제는 산업용 사물인터넷(IIoT) 환경에서의 고장 예측 모델 획득을 위한 전처리 방법, 피처의 중요도 측정 방법 및 피처 선택 방법, 장치 및 프로그램을 제공하는 것이다.
- [0006] 본 발명이 해결하고자 하는 과제들은 이상에서 언급된 과제로 제한되지 않으며, 언급되지 않은 또 다른 과제들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

- [0007] 상술한 과제를 해결하기 위한 본 발명의 일 면에 따른 산업용 사물인터넷(IIoT) 환경에서의 고장 예측 모델 획득 시스템의 제어 방법은, 서버가, 복수의 센서로부터 데이터를 획득하는 단계; 상기 서버가, 상기 획득된 데이터를 전처리하는 단계; 상기 서버가, 상기 전처리된 데이터를 바탕으로 피처의 중요도를 획득하는 단계; 상기 서버가, 상기 획득된 중요도를 바탕으로 피처를 선택하는 단계; 를 포함한다.
- [0008] 이때, 상기 전처리하는 단계는, 상기 복수의 센서로부터 획득된 데이터 각각에 대응되는 복수의 피처 중 적어도 하나의 피처를 제거하는 단계; 상기 적어도 하나의 피처가 제거된 복수의 피처에 대한 데이터를 정규화 하는 단계; 및 상기 정규화된 데이터를 분할하는 단계; 를 포함할 수 있다.
- [0009] 이때, 상기 피처를 제거하는 단계는, 상기 복수의 센서 각각으로부터 획득한 데이터 중, 비유효 데이터를 획득하는 단계; 상기 비유효 데이터의 비율이 [0,1]의 범위에서 기 설정된 유효성 계수보다 큰 경우, 상기 비유효 데이터에 대응되는 피처를 제거하는 단계; 를 포함하고, 상기 피처를 제거하는 단계는, 상기 복수의 센서 각각으로부터 획득한 데이터에 대한 각각의 분산값을 획득하는 단계; 상기 분산값이 기 설정된 값 이하인 경우, 상기 기 설정된 값 이하인 분산값에 대응되는 피처를 제거하는 단계; 를 포함할 수 있다.
- [0010] 이때, 상기 정규화 하는 단계는, 하기 수식 1을 바탕으로 정규화 될 수 있다.
- [0011] [수식 1]
- [0012]
$$xi'=(xi-m)/s$$
- [0013] 이때, xi' 는 (i + 1) 번째 정규화 된 특성 데이터, xi 는 피처의 (i + 1) 번째 데이터, m 은 피처의 평균, s 는 피처의 표준 편차일 수 있다.
- [0014] 이때, 상기 피처의 중요도를 획득하는 단계는, 상기 전처리 단계를 통해 획득된 데이터 중, 학습 데이터 세트를 획득하는 단계; 및 상기 학습 데이터를 랜덤 포레스트 기법을 바탕으로, 상기 학습 데이터에 대응되는 피처의 중요도를 획득하는 단계; 를 포함할 수 있다.
- [0015] 이때, 상기 피처를 선택하는 단계는, 하기 알고리즘 1을 바탕으로 피처를 선택하는 단계; 를 포함할 수 있다.

[0016] [알고리즘 1]

```

1: INITIALIZE  $sf_{cnt}$  to NULL,  $pa_{cnt}$  to 0, and  $cnt$  to 0 // Initialize variables
2: REPEAT // Start iteration
3:   /* ===== Feature Selection Step ===== */
4:    $maxImp \leftarrow \max(IMP)$  // Find the highest importance from IMP
5:   FOR each feature index,  $j$ ,  $j \in [0, nf-1]$ 
6:     IF  $IMP[j] == maxImp$  // Find feature index with the highest importance
7:        $sf_{cnt} \leftarrow F[j]$  // Select a new feature
8:        $F[j], IMP[j] \leftarrow 0$  // Remove the selected feature from IMP and F
9:     ENDIF
10:  ENDFOR
11:   $SF[cnt] \leftarrow sf_{cnt}$  // Update the selected feature set

```

[0017]

[0018] 이때, cnt는 카운터 값, sf_{cnt} 는 cnt에서의 선택된 피쳐, 피쳐의 중요도 세트는 $IMP=[imp_0, \dots, imp_j, \dots, imp_{nf-1}]$, 피쳐의 세트는 $F=[f_0, \dots, f_j, \dots, f_{nf-1}]$, imp_j 는, (j+1)번째 피쳐의 중요도의 값, nf는 피쳐의 수일 수 있다.

[0019] 이때, 상기 제어 방법은, 상기 선택된 피쳐를 바탕으로 복수의 모델을 생성하는 단계; 및 상기 생성된 복수의 모델 중, 적어도 하나의 모델을 선택하는 단계;를 더 포함하고, 상기 복수의 모델을 생성하는 단계는, 하기 알고리즘 2에 의해 생성될 수 있다.

[0020] [알고리즘 2]

```

1: INITIALIZE  $sf_{cnt}$  to NULL,  $pa_{cnt}$  to 0, and  $cnt$  to 0 // Initialize variables
2: REPEAT // Start iteration
3:   /* ===== Feature Selection Step ===== */
4:    $maxImp \leftarrow \max(IMP)$  // Find the highest importance from IMP
5:   FOR each feature index,  $j$ ,  $j \in [0, nf-1]$ 
6:     IF  $IMP[j] == maxImp$  // Find feature index with the highest importance
7:        $sf_{cnt} \leftarrow F[j]$  // Select a new feature
8:        $F[j], IMP[j] \leftarrow 0$  // Remove the selected feature from IMP and F
9:     ENDIF
10:  ENDFOR
11:   $SF[cnt] \leftarrow sf_{cnt}$  // Update the selected feature set
12:  /* ===== Model Building Step ===== */
13:   $pa_{cnt} \leftarrow \text{modelBuildEvalFunction}(SF)$  // build and evaluate model
14:   $PA[cnt] \leftarrow pa_{cnt}$  // Update the prediction accuracy set
15:   $cnt \leftarrow cnt + 1$  // Increment iteration counter by one
16: UNTIL  $cnt \leq max$  // Terminate iteration if condition become FALSE
17: RETURN SF and PA // Return selected feature set and prediction accuracy set

```

[0021]

[0022] 이때, cnt는 카운터 값, SF는 선택된 피쳐, PA는 예측 정확도, sf_{cnt} 는 cnt에서의 선택된 피쳐, 피쳐의 중요도 세트는 $IMP=[imp_0, \dots, imp_j, \dots, imp_{nf-1}]$, 피쳐의 세트는 $F=[f_0, \dots, f_j, \dots, f_{nf-1}]$, imp_j 는, (j+1)번째 피쳐의 중요도의 값, nf는 피쳐의 수, $\text{modelBuildEvalFunction}(SF)$ 는 예측 모델을 구축하고 평가하기 위한 임의의 함수일 수 있다.

[0023] 본 발명의 기타 구체적인 사항들은 상세한 설명 및 도면들에 포함되어 있다.

발명의 효과

[0024] 상술한 본 발명의 다양한 실시예에 따라, IIoT 환경에서 효율적이고 정확한 고장 예측 모델을 구축할 수 있는

새로운 효과가 존재한다.

[0025] 본 발명의 효과들은 이상에서 언급된 효과로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0026] 도 1은 본 발명의 일 실시예에 따른 IIoT 시스템을 도시한 시스템도이다.

도 2는 본 발명의 일 실시예에 따른 고장 예측 모델 획득 방법을 설명하기 위한 블록도이다.

도 3은 본 발명의 일 실시예에 따른 고장 예측 모델 획득을 위한 전처리, 피처의 중요도 측정 및 피처 선택 방법을 설명하기 위한 흐름도이다.

도 4는 본 발명의 일 실시예에 따른 전처리 방법을 설명하기 위한 흐름도이다.

도 5는 본 발명의 일 실시예에 따른 피처 제거 방법을 설명하기 위한 흐름도이다.

도 6은 본 발명의 일 실시예에 따른 피처의 중요도 획득 방법을 설명하기 위한 흐름도이다.

도 7 내지 도 11은 본 발명의 일 실시예에 따른 실험 결과를 설명하기 위한 예시도이다.

도 12는 본 발명의 일 실시예에 따른 장치의 구성도이다.

발명을 실시하기 위한 구체적인 내용

[0027] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나, 본 발명은 이하에서 개시되는 실시예들에 제한되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술 분야의 통상의 기술자에게 본 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다.

[0028] 본 명세서에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다. 명세서에서 사용되는 "포함한다(comprises)" 및/또는 "포함하는(comprising)"은 언급된 구성요소 외에 하나 이상의 다른 구성요소의 존재 또는 추가를 배제하지 않는다. 명세서 전체에 걸쳐 동일한 도면 부호는 동일한 구성 요소를 지칭하며, "및/또는"은 언급된 구성요소들의 각각 및 하나 이상의 모든 조합을 포함한다. 비록 "제1", "제2" 등이 다양한 구성요소들을 서술하기 위해서 사용되나, 이들 구성요소들은 이들 용어에 의해 제한되지 않음은 물론이다. 이들 용어들은 단지 하나의 구성요소를 다른 구성요소와 구별하기 위하여 사용하는 것이다. 따라서, 이하에서 언급되는 제1 구성요소는 본 발명의 기술적 사상 내에서 제2 구성요소일 수도 있음은 물론이다.

[0029] 다른 정의가 없다면, 본 명세서에서 사용되는 모든 용어(기술 및 과학적 용어를 포함)는 본 발명이 속하는 기술 분야의 통상의 기술자에게 공통적으로 이해될 수 있는 의미로 사용될 수 있을 것이다. 또한, 일반적으로 사용되는 사전에 정의되어 있는 용어들은 명백하게 특별히 정의되어 있지 않는 한 이상적으로 또는 과도하게 해석되지 않는다.

[0030] 명세서에서 사용되는 "부" 또는 "모듈"이라는 용어는 소프트웨어, FPGA 또는 ASIC과 같은 하드웨어 구성요소를 의미하며, "부" 또는 "모듈"은 어떤 역할들을 수행한다. 그렇지만 "부" 또는 "모듈"은 소프트웨어 또는 하드웨어에 한정되는 의미는 아니다. "부" 또는 "모듈"은 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 재생시키도록 구성될 수도 있다. 따라서, 일 예로서 "부" 또는 "모듈"은 소프트웨어 구성요소들, 객체지향 소프트웨어 구성요소들, 클래스 구성요소들 및 태스크 구성요소들과 같은 구성요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로 코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들 및 변수들을 포함한다. 구성요소들과 "부" 또는 "모듈"들 안에서 제공되는 기능은 더 작은 수의 구성요소들 및 "부" 또는 "모듈"들로 결합되거나 추가적인 구성요소들과 "부" 또는 "모듈"들로 더 분리될 수 있다.

[0031] 공간적으로 상대적인 용어인 "아래(below)", "아래(beneath)", "하부(lower)", "위(above)", "상부(upper)" 등은 도면에 도시되어 있는 바와 같이 하나의 구성요소와 다른 구성요소들과의 상관관계를 용이하게 기술하기 위해 사용될 수 있다. 공간적으로 상대적인 용어는 도면에 도시되어 있는 방향에 더하여 사용시 또는 동작시 구성

요소들의 서로 다른 방향을 포함하는 용어로 이해되어야 한다. 예를 들어, 도면에 도시되어 있는 구성요소를 뒤집을 경우, 다른 구성요소의 "아래(below)"또는 "아래(beneath)"로 기술된 구성요소는 다른 구성요소의 "위(above)"에 놓여질 수 있다. 따라서, 예시적인 용어인 "아래"는 아래와 위의 방향을 모두 포함할 수 있다. 구성요소는 다른 방향으로도 배향될 수 있으며, 이에 따라 공간적으로 상대적인 용어들은 배향에 따라 해석될 수 있다.

- [0032] 본 명세서에서, 컴퓨터는 적어도 하나의 프로세서를 포함하는 모든 종류의 하드웨어 장치를 의미하는 것이고, 실시 예에 따라 해당 하드웨어 장치에서 동작하는 소프트웨어적 구성도 포괄하는 의미로서 이해될 수 있다. 예를 들어, 컴퓨터는 스마트폰, 태블릿 PC, 데스크톱, 노트북 및 각 장치에서 구동되는 사용자 클라이언트 및 애플리케이션을 모두 포함하는 의미로서 이해될 수 있으며, 또한 이에 제한되는 것은 아니다.
- [0033] 이하, 첨부된 도면을 참조하여 본 발명의 실시예를 상세하게 설명한다.
- [0034] 본 명세서에서 설명되는 각 단계들은 컴퓨터에 의하여 수행되는 것으로 설명되나, 각 단계의 주체는 이에 제한되는 것은 아니며, 실시 예에 따라 각 단계들의 적어도 일부가 서로 다른 장치에서 수행될 수도 있다.
- [0035] 도 1은 본 발명의 일 실시예에 따른 IIoT 시스템을 도시한 시스템도이다.
- [0036] 도 1에 도시된 바와 같이, IIoT 시스템은 서버(100) 및 복수의 센서(200-1 내지 200-5)를 포함할 수 있다.
- [0037] 서버(100)는 복수의 센서(200-1 내지 200-5)로부터 센싱 데이터를 획득할 수 있다. 서버(100)는 획득된 센싱 데이터 및 센싱 데이터에 대응되는 복수의 센서(200-1 내지 200-5)를 바탕으로 시스템의 고장을 예측하기 위한 고장 예측 모델을 구축할 수 있다.
- [0038] 도 2는 본 발명의 일 실시예에 따른 고장 예측 모델 획득 방법을 설명하기 위한 블록도이다.
- [0039] 서버(100)는, 고장 예측 모델의 예측 정확도를 최대화 하기 위해 선택한 피처를 기반으로 고장 예측 모델을 획득할 수 있다. 이를 위해, 서버(100)는 피처 선택을 반복적으로 수행할 수 있으며, 반복해서 선택한 피처를 기반으로 복수의 고장 예측 모델을 획득할 수 있다. 이후, 서버(100)는, 복수의 고장 예측 모델 중 가장 높은 예측 정확도를 갖는 고장 예측 모델을 획득할 수 있다.
- [0040] 구체적으로, 도 2에 도시된 바와 같이, 서버(100) 고장 예측 모델을 획득하기 위해 1) 전처리 단계(Processing), 2) 중요도 측정 단계(Importance Measurement), 3) 피처 선택 단계(Feature Selection), 4) 모델 구축 단계(Model Building) 및 5) 모델 선택 단계(Model Selection)를 수행할 수 있다.
- [0041] 구체적으로, 전처리 단계에서, 서버(100)는 피처 제거, 누락 된 데이터 대체, 정규화 및 데이터 분할 단계를 수행할 수 있다. 이후, 서버(100)는 전처리 과정을 통해 획득된 피처의 중요도를 획득할 수 있다. 이후, 서버(100)는, 피처 선택 및 모델 구축 단계를 반복적으로 수행할 수 있다. 이후, 서버(100)는 모델 구축 단계에서, 선택된 피처에 기초한 SVM(Support Vector Machine)을 통해 고장 예측 모델을 구축할 수 있다.
- [0042] 이하에서는 도 3 내지 도 6을 바탕으로 고장 예측 모델을 생성하는 구체적인 방법에 대해 살펴본다.
- [0043] 도 3은 본 발명의 일 실시예에 따른 고장 예측 모델 획득을 위한 전처리, 피처의 중요도 측정 및 피처 선택 방법을 설명하기 위한 흐름도이다.
- [0044] 단계 S110에서, 서버(100)는 복수의 센서로부터 복수의 데이터를 획득할 수 있다.
- [0045] 일 실시예에 따라, 본 발명의 다양한 실시예에서, 피처의 용어와 센서의 용어는 동일하게 해석될 수 있다.
- [0046] 단계 S120에서, 서버(100)는, 획득된 데이터를 전처리할 수 있다.
- [0047] 단계 S130에서, 서버(100)는, 전처리된 데이터를 바탕으로 피처의 중요도를 획득할 수 있다.
- [0048] 단계 S140에서, 서버(100)는, 획득된 중요도를 바탕으로 피처를 선택할 수 있다.
- [0049] 도 4는 본 발명의 일 실시예에 따른 전처리 방법을 설명하기 위한 흐름도이다.
- [0050] 단계 S210에서, 서버(100)는, 복수의 센서로부터 획득된 데이터 각각에 대응되는 복수의 피처 중 적어도 하나의 피처를 제거할 수 있다.
- [0051] 단계 S220에서, 서버(100)는, 적어도 하나의 피처가 제거된 복수의 피처에 대한 데이터를 정규화할 수 있다.
- [0052] 구체적으로, 서버(100)는, 각 피처의 데이터 규모에 맞게 정규화 과정을 수행할 수 있다. 일 실시예에 따라, 서

버(100)는 하기 수학적 식 1을 바탕으로, 각각의 피처의 정규화 과정을 수행할 수 있다.

[0053] [수학적 식 1]

[0054]
$$xi' = (xi - m) / s$$

[0055] 이때, xi' 는 $(i + 1)$ 번째 정규화 된 특성 데이터, xi 는 피처의 $(i + 1)$ 번째 데이터, m 은 피처의 평균, s 는 피처의 표준 편차일 수 있다.

[0056] 이때, 피처의 평균이라 함은 각각의 피처(예를 들어 센서)에서 수집된 데이터 값의 평균을 의미할 수 있다.

[0057] 일 실시예로, 피처가 온도 감지 센서인 경우, 피처의 평균은 온도 감지 센서에 의해 획득된 복수의 온도의 데이터에 대한 평균값을 의미할 수 있다. 피처의 표준편차 및 피처의 분산 또한, 각각의 피처에서 수집된 데이터 값의 표준편차 또는 분산임은 자명하다.

[0058] 단계 S230에서, 서버(100)는, 정규화된 데이터를 분할할 수 있다.

[0059] 구체적으로, 서버(100)는, 정규화 된 데이터 세트를 학습 데이터 세트(Training Dataset) 및 테스트 데이터 세트(Test Dataset)로 분류할 수 있다. 서버(100)는, 학습 데이터 세트는 고장 예측 모델을 획득하기 위해 사용하고, 테스트 데이터 세트는 예측 모델의 성능을 평가하는 데 사용할 수 있다.

[0060] 도 5는 본 발명의 일 실시예에 따른 피처 제거 방법을 설명하기 위한 흐름도이다.

[0061] 단계 S310에서, 서버(100)는, 복수의 센서 각각으로부터 획득한 데이터 중, 비유효 데이터를 획득할 수 있다.

[0062] 단계 S320에서, 서버(100)는, 상기 특정 피처에 대한 비유효 데이터의 비율이 $[0, 1]$ 의 범위에서 기 설정된 유효성 계수보다 큰 경우, 비유효 데이터에 대응되는 피처를 제거할 수 있다.

[0063] 이때, 비유효 데이터란, 피처로부터 수집된 데이터 중 값이 없는 데이터를 의미할 수 있다. 일 실시예로, 피처가 온도 센서인 경우, 온도가 감지되지 않은 데이터가 비유효 데이터일 수 있다. 비유효 데이터는 수집된 데이터 세트(훈련 데이터 세트 및 테스트 데이터 세트를 포함할 수 있다)에서 각 피처별로 값이 없는 행에 대응되는 데이터일 수 있다.

[0064] 단계 S330에서, 서버(100)는, 복수의 센서 각각으로부터 획득한 데이터에 대한 각각의 분산값을 획득할 수 있다.

[0065] 단계 S340에서, 서버(100)는, 분산값이 기 설정된 값 이하인 경우, 기 설정된 값 이하인 분산값에 대응되는 피처를 제거할 수 있다.

[0066] 구체적으로, 전처리 단계에서, 서버(100)는 피처 제거, 누락 된 데이터 대체, 정규화 및 데이터 분할 단계를 수행할 수 있다. 구체적으로, 서버(100) 입력 데이터 세트(즉, 복수의 센서로부터 수집 된 데이터)에서 유효하지 않은 피처를 제거하기 위하여 각 피처의 비유효 (NA) 데이터를 검색하고, 각 피처의 분산을 계산할 수 있다.

[0067] 일 실시예에 따라, 서버(100)는, 특정 피처의 NA 데이터의 비율이 $[0, 1]$ 범위에서의 기 설정된 유효성 계수보다 큰 경우, 해당 특정 피처를 입력 데이터 세트에서 제거할 수 있다.

[0068] 또 다른 실시예로, 서버(100)는 특정 피처의 분산이 기 설정된 값 이하인 경우, 해당 피처를 입력 데이터 세트에서 제거할 수 있다. 바람직하게는, 특정 피처의 분산이 0에 가까울수록 해당 피처가 입력 데이터 세트에서 제거될 확률이 높아질 수 있다.

[0069] 한편, 누락 된 데이터가 존재하는 경우 (즉, 피처 제거 후 남아있는 피처에 NA 데이터가 존재하는 경우), 서버(100)는 누락 된 데이터를 각 피처의 s 누락되지 않은 데이터의 평균값으로 대체할 수 있다.

[0070] 도 6은 본 발명의 일 실시예에 따른 피처의 중요도 획득 방법을 설명하기 위한 흐름도이다.

[0071] 단계 S410에서, 서버(100)는, 전처리 단계를 통해 획득된 데이터 중, 학습 데이터 세트를 획득할 수 있다.

[0072] 단계 S420에서, 서버(100)는, 학습 데이터를 랜덤 포레스트 기법을 바탕으로, 학습 데이터에 대응되는 피처의 중요도를 획득할 수 있다.

[0073] 구체적으로, 서버(100)는 전처리 과정을 통해 획득된 피처의 중요도를 획득할 수 있다. 서버(100)는, 랜덤 포레스트 기법을 바탕으로, 학습 데이터 세트를 분석하여, 각 피처의 중요도를 획득할 수 있다.

[0074] 구체적으로, 서버(100)는, 랜덤 포레스트 기법을 통해 복수개의 의사 결정 트리가 작성되고 작성된 의사 결정

트리를 종합적으로 고려하여 각 피처와 Fail 간의 관련성을 분석할 수 있다.

- [0075] 일 실시예에 따라, 서버(100)에 의해 수행되는 랜덤 포레스트 기법에서는 각각 의사 결정 트리를 다르게 획득하기 위해 학습 데이터 세트의 복수의 하위 세트가 생성될 수 있다. 생성된 복수의 하위 세트 각각은 서로 다른 데이터와 피처로 구성될 수 있다.
- [0076] 이를 위해, 서버(100)는, 학습 데이터 세트에서 n 개의 데이터(즉, n 개의 행) 및 m try개의 피처(즉, m try개의 열)가 무작위로 선택할 수 있다. 서버(100)는 획득된 서브 세트 수가 $ntree$ 개로 기 정의 된 의사 결정 트리 수에 도달 할 때까지 상기 무작위 선택을 반복할 수 있다.
- [0077] 이후, 각각의 의사 결정 트리는 획득된 서브 세트 중 하나를 바탕으로 별도로 구축될 수 있다. 즉, $ntree$ 개의 결정 트리를 구축 한 후 각 피처의 중요성은 Mean Decrease Gini(MDG, Gini importance score)를 통해 측정되며, 이는 각각 피처가 정확한 예측 결과에 얼마나 영향을 미치는지를 나타내는 지표가 될 수 있다.
- [0078] 보다 구체적으로, 서버(100)는, 각각의 의사 결정 트리에서, 특정 피처를 사용하는 부모 노드와 해당 부모 노드의 자식 노드 사이의 Gini impurities(Gini 불순도) 차이의 합을 계산할 수 있다. 이후, 서버(100)는 모든 의사 결정 트리 결과의 평균을 계산할 수 있다. 이때, 의사 결정 트리는 특정 피처의 임계 값을 사용하여 의사 결정을 내리는 여러 노드로 구성될 수 있다. 또한 각각의 노드에는 잘못된 결정 가능성을 측정하기 위한 Gini impurities을 포함할 수 있다.
- [0079] 한편, 랜덤 포레스트 기법이란, 다수의 의사 결정 트리들을 학습하는 앙상블 방법으로서, 랜덤 포레스트는 검출, 분류, 그리고 회귀 등 다양한 문제에 활용될 수 있다.
- [0080] 한편, 본 발명에 따른 랜덤 포레스트는 랜덤성(randomness)에 의해 트리들이 서로 조금씩 다른 특성을 가지게 되며, 이러한 특성은 각 트리들의 예측(prediction)들이 비상관화(decorrelation) 되게하며, 결과적으로 일반화(generalization) 성능을 향상시킬 수 있는 효과가 존재한다. 또한, 랜덤화(randomization)는 포레스트가 노이즈가 포함된 데이터에 대해서도 강인하게 만들어 줄 수 있다. 랜덤화는 각 트리들의 훈련 과정에서 진행되며, 랜덤 학습 데이터 추출 방법을 이용한 앙상블 학습법인 배깅(bagging)과 랜덤 노드 최적화(randomized node optimization)가 사용될 수 있다.
- [0081] 이후, 서버(100)는, 피처 선택 및 모델 구축 단계를 반복적으로 수행할 수 있다. 구체적으로, 하기 알고리즘 1은 피처 선택 단계 및 모델 구축 단계의 전반적인 동작을 나타낸다.

Algorithm 1. Operation of feature selection and model building steps

```

1: INITIALIZE  $sf_{cnt}$  to NULL,  $pa_{cnt}$  to 0, and  $cnt$  to 0 // Initialize variables
2: REPEAT // Start iteration
3:   /* ===== Feature Selection Step ===== */
4:    $maxImp \leftarrow \max(IMP)$  // Find the highest importance from IMP
5:   FOR each feature index,  $j$ ,  $j \in [0, nf - 1]$ 
6:     IF  $IMP[j] == maxImp$  // Find feature index with the highest importance
7:        $sf_{cnt} \leftarrow F[j]$  // Select a new feature
8:        $F[j], IMP[j] \leftarrow 0$  // Remove the selected feature from IMP and F
9:     ENDIF
10:  ENDFOR
11:   $SF[cnt] \leftarrow sf_{cnt}$  // Update the selected feature set
12:  /* ===== Model Building Step ===== */
13:   $pa_{cnt} \leftarrow \text{modelBuildEvalFunction}(SF)$  // build and evaluate model
14:   $PA[cnt] \leftarrow pa_{cnt}$  // Update the prediction accuracy set
15:   $cnt \leftarrow cnt + 1$  // Increment iteration counter by one
16: UNTIL  $cnt \leq max$  // Terminate iteration if condition become FALSE
17: RETURN SF and PA // Return selected feature set and prediction accuracy set

```

- [0082]
- [0083] 한편, 알고리즘 1에서, 피처의 중요도 세트는 $IMP = [imp_0, \dots, imp_j, \dots, imp_{nf-1}]$ 로 정의될 수 있으며, 피처의 세트는 $F = [f_0, \dots, f_j, \dots, f_{nf-1}]$ 로 정의될 수 있다.

- [0084] 이때, nf 는 피처의 수, imp_j 는 $(j + 1)$ 번째 피처의 중요도일 수 있다.
- [0085] 상기 알고리즘 1은 시작 시 변수가 초기화 되며, sf_{cnt} 는 $(cnt + 1)$ 번째 반복으로 선택된 피처를 의미하며, pa_{cnt} 는, $(cnt + 1)$ 번째 반복으로 구축된 모델의 고장 예측 정확도를 의미하며, cnt 는 0에서 1씩 증가하는 카운터 값을 의미할 수 있다.
- [0086] 서버(100)는, 상기 알고리즘 1에서 새로 선택한 피처의 중요도가 중요도 임계 값보다 작을 때까지 상기 알고리즘 1을 반복할 수 있다. 즉, 서버(100)는 새로 선택한 피처의 중요도가 중요도 임계 값보다 작을 때까지 피처 선택 단계 및 모델 작성 단계를 반복할 수 있다.
- [0087] 이때, 알고리즘 1의 최대 반복 횟수(max)는 중요도가 중요도 임계 값보다 큰 피처 수와 같게 설정될 수 있다. 따라서 cnt 가 최대 반복 횟수(max)값에 도달하면 서버(100)는 알고리즘 1의 반복을 종료시킬 수 있다. 피처 선택 단계가 반복되는 동안, 서버(100)는, 중요도가 가장 높은 새 피처를 선택하고 선택한 피처를 피처 세트(SF)에 추가할 수 있다. 결론적으로, SF에 포함 된 피처의 수는 반복 횟수가 증가함에 따라 1씩 증가할 수 있다.
- [0088] 이후, SF가 업데이트되면, 서버(100)는 모델 구축 단계에서, SF에 기초한 SVM을 통해 고장 예측 모델을 구축할 수 있다.
- [0089] 이후, 서버(100)는 구축된 고장 예측 모델의 예측 정확도를 평가하여 예측 정확도 세트 (PA)를 업데이트할 수 있다. 이때, 알고리즘 1에서 $modelBuildEvalFunction(SF)$ 은 예측 모델을 구축하고 평가하기 위한 함수로, 종래의 예측 평가 함수가 사용될 수 있음은 물론이다.
- [0090] 예를 들어, cnt 가 2이고(3번째 루프) 선택된 피처 세트 $SF = [F39, F12, F19]$ 이면, 서버(100)는, 해당 3개의 피처의 훈련용 데이터 세트만을 가지고 SVM 기반의 예측 모델을 생성할 수 있다. 이후 서버(100)는, 해당 3개의 피처에 대한 테스트 데이터 세트를 가지고 pa_2 를 산출하고 PA 세트를 업데이트할 수 있다. (즉, PA세트 내의 pa 수가 2개에서 3개로 증가한다. $PA=[pa_0, pa_1, pa_2]$)
- [0091] 한편, 본 발명의 일 실시예에 따라, SF는 선택된 피처의 집합일 수 있으며, 중요도가 큰 순서로 정렬될 수 있다. 즉, 알고리즘 1의 Repeat 루프에서 cnt 가 1씩 증가할때 마다 포함되는 중요도가 큰 피처가 1개씩 추가될 수 있다.
- [0092] 서버(100)는, SF를 입력으로 하여 예측 정확도를 도출할 수 있다. 반복이 종료되면 서버(100)는 최종 SF 및 PA를 작업의 결과로 도출할 수 있다.
- [0093] 즉, 서버(100)는 SF를 입력 값으로 하여 예측 정확도(PA)를 획득할 수 있으며, 알고리즘 1의 반복이 종료됨에 따라 최종적으로 도출된 SF 및 PA를 결과값으로 산출할 수 있다.
- [0094] 이후, 서버(100)는, 모델 선택 단계에서, PA를 참조하여 예측 정확도가 가장 높은 고장 예측 모델을 선택할 수 있다. 구체적으로, 서버(100)는 $\max(PA)$ 를 통해 가장 높은 예측 정확도를 선택하고, $\max(PA)$ 의 인덱스를 도출하여 예측 정확도가 가장 높은 반복 횟수(즉, 선택된 피처 수)를 산출할 수 있다. 서버(100)는 $\max(PA)$ 의 인덱스와 SF를 고려하여 고장 예측 모델을 선택할 수 있다. 이때, $\max(PA)$ 는 알고리즘 1의 반복을 통해 획득된 PA(즉, PA 세트) 중, 가장 큰 값을 의미할 수 있음은 물론이다.
- [0095] 이때, $\max(PA)$ 의 인덱스는 PA 세트에서 최대 정확도가 몇번째 순번에 포함되어 있는지를 의미할 수 있다. 즉, 인덱스는 cnt 의 값이 몇일때 (즉, 반복(iteration) 횟수가 몇번째 일때) 가장 큰 정확도를 갖는지를 나타내는 지표이다.
- [0096] 인덱스는 PA 세트 내 pa 값들과 $\max(PA)$ 를 비교하여, $\max(PA)$ 와 동일한 pa 의 순번(index= cnt)을 PA 세트에서 획득될 수 있으며, 서버(100)는, PA세트 내에서 $\max(PA)$ 의 인덱스(즉, cnt 값)를 획득하고, 해당 반복시행에서의 피처들을 최종적으로 선택하여 예측모델을 구축할 수 있다.
- [0097] 예를 들어, 도 8에 도시된 바와 같이, $\max(PA)$ 의 인덱스는 7 이고, 이때 선택되는 피처 세트는 $SF = [F60, F349, F41, F289, F427, F65, F66, F154]$ (즉, 중요도 순 8개 피처)이며 해당 SF 내 포함된 피처의 데이터들로 만든 SVM 기반 예측모델이 최종적으로 선정 및 사용될 수 있다.
- [0098] 도 7 내지 도 11은 본 발명의 일 실시예에 따른 실험 결과를 설명하기 위한 예시도이다.
- [0099] 본 발명에서는, 오픈 소스 R 버전 3.4.3을 사용하여 제안 된 고장 예측 모델의 타당성을 검증하기 위해 실험 구

현이 수행되었다. 이를 위해 UCI 저장소에서 제공 한 SECOM 데이터 세트가 사용되었으며, 해당 데이터 세트는 1567 개의 데이터와 591 개의 피처로 구성되고 센서 및 프로세스 측정 지점을 모니터링하여 반도체 제조 프로세스에서 데이터를 수집하였다. 590 개의 피처 데이터는 여러 센서에서 측정되었으며, 나머지 피처의 데이터는 Pass 및 Fail로 표시되는 가계도 테스트 결과이다.

- [0100] 본 실험에서는, 피처 제거를 위해 유효성 계수를 0.1로 설정하였다. 즉, 10 % 이상의 NA 데이터를 갖는 특징 및 분산이 없는 특징은 데이터 세트에서 제거되었다. 상술한 피처 제거 과정을 통해, 피처 수는 591에서 393으로 감소하였음을 확인할 수 있다.
- [0101] 학습 데이터 세트와 테스트 데이터 세트의 비율은 7:3으로 설정되었다. 각각의 피처의 중요도를 측정하기 위해 랜덤 포레스트 기법 및 caret 패키지를 사용하였다. n, mtry 및 ntree를 각각 1000, 19 및 500으로 설정하였다. 이 설정을 통해 임의로 생성 된 1000 * 19 매트릭스를 사용하여 500 개의 의사 결정 트리가 작성되었다. 각각의 특징의 중요성은 MDG(mean Decrease Gini)를 통해 측정되었다.
- [0102] 도 7은 상위 30 개 피처의 중요성을 나타낸다. x 축 및 y 축은 각각 MDG 및 특징을 나타낸다. 그림에서 F60은 모든 피처 중에서 MDG (1.52)가 가장 높음을 확인할 수 있다.
- [0103] 한편, 본 실험에서는, 반복적인 피처 선택 과정 수행을 위하여, 중요도 임계 값을 0.7로 설정하였다. 따라서 최대 반복 횟수(max)는 24로 결정되었으며, 이는 반복 횟수가 24임을 의미한다.
- [0104] 학습 데이터 세트에는 70 개의 Fail과 1038 개의 Pass가 포함된다. 이러한 학습 데이터 세트의 불균형은 고장 예측 모델을 고장 사례를 예측하기 어렵게 만들고, 이러한 문제를 해결하기 위해 고장 예측 모델을 작성하기 전에 샘플링을 수행하였다 이를 통해, 일부 Pass 데이터가 제거되어 모델 작성에 미치는 영향을 감소시켰다. 상술한 바와 같이, 피처 선택 결과와 샘플링 된 학습 데이터 세트를 통해 SVM을 사용하여 고장 예측 모델을 구축하였으며, 고장 예측 모델은, e1071 package in R을 사용하여 구축되었다. 도 8에 도시된 표는 획득 한 SF 및 PA를 나타낸다. 도 8의 표에서 max(PA)는 0.72이고 인덱스는 7임을 확인할 수 있다 결론적으로 8 가지 피처(예: F60, F349, F41, F289, F427, F65, F66 및 F154)가 선택된다.
- [0105] 성능 평가를 위해 제안 된 모델의 예측 정확도를 기존 모델의 예측 정확도와 비교하였다. 도 9 내지 도 11은 서로 다른 수의 피처를 사용하는 세 가지 고장 예측 모델에 대한 ROC(수신기 작동 특성) 곡선을 나타낸다. ROC 곡선은 예측 모델의 성능 측정으로, TPR (true positive rate)과 FPR (false positive rate) 간의 관계를 나타낸다 ROC 곡선에서는 곡선 아래 면적 (AUC)을 사용하여 모형의 예측 정확도를 평가한다. 구체적으로, AUC가 클수록 예측 정확도가 높아진다. 도 9에서, 반복적 특징 선택에 기초하여 구축 된 고장 예측 모델은 다른 모델에 비해 가장 큰 AUC를 가지는 것을 확인할 수 있다. 이는, 모델이 다른 수의 피처를 사용하여 반복적으로 구축되고 예측 정확도가 가장 높은 모델 중 하나가 선택되기 때문이다.
- [0106] 즉, 관련 없는 피처로 인해 정확한 예측 모델을 작성하기가 어렵 기 때문에, 고정 된 개수의 피처가 모델을 구축하는 데 사용되는 경우 예측 정확도가 상대적으로 저하 될 수 있다.
- [0107] 즉, 고장과 관련이 없는 더 많은 피처를 사용하여 모델을 작성하면 모델의 예측 정확도가 떨어질 수 있다. 따라서, 데이터 세트의 모든 특징이 도 11에 도시 된 바와 같이 사용되는 경우, AUC는 상당히 감소한다. 정량적으로, 제안된 모델은 각각 고정된 수의 특징 및 모든 특징 경우에 비해 14.3 및 22.0 % 더 높은 AUC를 얻는다.
- [0108] 그러나, 본 발명은 다른 수의 피처를 사용하여 반복적으로 고장 예측 모델을 구축하며, 나아가, 고장과 관련된 피처를 선택적으로 획득하여 고장 예측 모델을 구축하므로 도 11에 도시된 바와 같이, 기존의 모델에 비해 더 나은 효과가 존재하게 된다.
- [0109] 도 12는 본 발명의 일 실시예에 따른 장치의 구성도이다.
- [0110] 프로세서(102)는 하나 이상의 코어(core, 미도시) 및 그래픽 처리부(미도시) 및/또는 다른 구성 요소와 신호를 송수신하는 연결 통로(예를 들어, 버스(bus) 등)를 포함할 수 있다.
- [0111] 일 실시예에 따른 프로세서(102)는 메모리(104)에 저장된 하나 이상의 인스트럭션을 실행함으로써, 도 3 내지 도 6과 관련하여 설명된 방법을 수행한다.
- [0112] 한편, 프로세서(102)는 프로세서(102) 내부에서 처리되는 신호(또는, 데이터)를 일시적 및/또는 영구적으로 저장하는 램(RAM: Random Access Memory, 미도시) 및 롬(ROM: Read-Only Memory, 미도시)을 더 포함할 수 있다.

또한, 프로세서(102)는 그래픽 처리부, 램 및 롬 중 적어도 하나를 포함하는 시스템온칩(SoC: system on chip) 형태로 구현될 수 있다.

[0113] 메모리(104)에는 프로세서(102)의 처리 및 제어를 위한 프로그램들(하나 이상의 인스트럭션들)을 저장할 수 있다. 메모리(104)에 저장된 프로그램들은 기능에 따라 복수 개의 모듈들로 구분될 수 있다.

[0114] 본 발명의 실시예와 관련하여 설명된 방법 또는 알고리즘의 단계들은 하드웨어로 직접 구현되거나, 하드웨어에 의해 실행되는 소프트웨어 모듈로 구현되거나, 또는 이들의 결합에 의해 구현될 수 있다. 소프트웨어 모듈은 RAM(Random Access Memory), ROM(Read Only Memory), EPROM(Erasable Programmable ROM), EEPROM(Electrically Erasable Programmable ROM), 플래시 메모리(Flash Memory), 하드 디스크, 착탈형 디스크, CD-ROM, 또는 본 발명이 속하는 기술 분야에서 잘 알려진 임의의 형태의 컴퓨터 판독가능 기록매체에 상주할 수도 있다.

[0115] 본 발명의 구성 요소들은 하드웨어인 컴퓨터와 결합되어 실행되기 위해 프로그램(또는 애플리케이션)으로 구현되어 매체에 저장될 수 있다. 본 발명의 구성 요소들은 소프트웨어 프로그래밍 또는 소프트웨어 요소들로 실행될 수 있으며, 이와 유사하게, 실시 예는 데이터 구조, 프로세스들, 루틴들 또는 다른 프로그래밍 구성들의 조합으로 구현되는 다양한 알고리즘을 포함하여, C, C++, 자바(Java), 어셈블러(assembler) 등과 같은 프로그래밍 또는 스크립팅 언어로 구현될 수 있다. 기능적인 측면들은 하나 이상의 프로세서들에서 실행되는 알고리즘으로 구현될 수 있다.

[0116] 이상, 첨부된 도면을 참조로 하여 본 발명의 실시예를 설명하였지만, 본 발명이 속하는 기술분야의 통상의 기술자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로, 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며, 제한적이지 않은 것으로 이해해야만 한다.

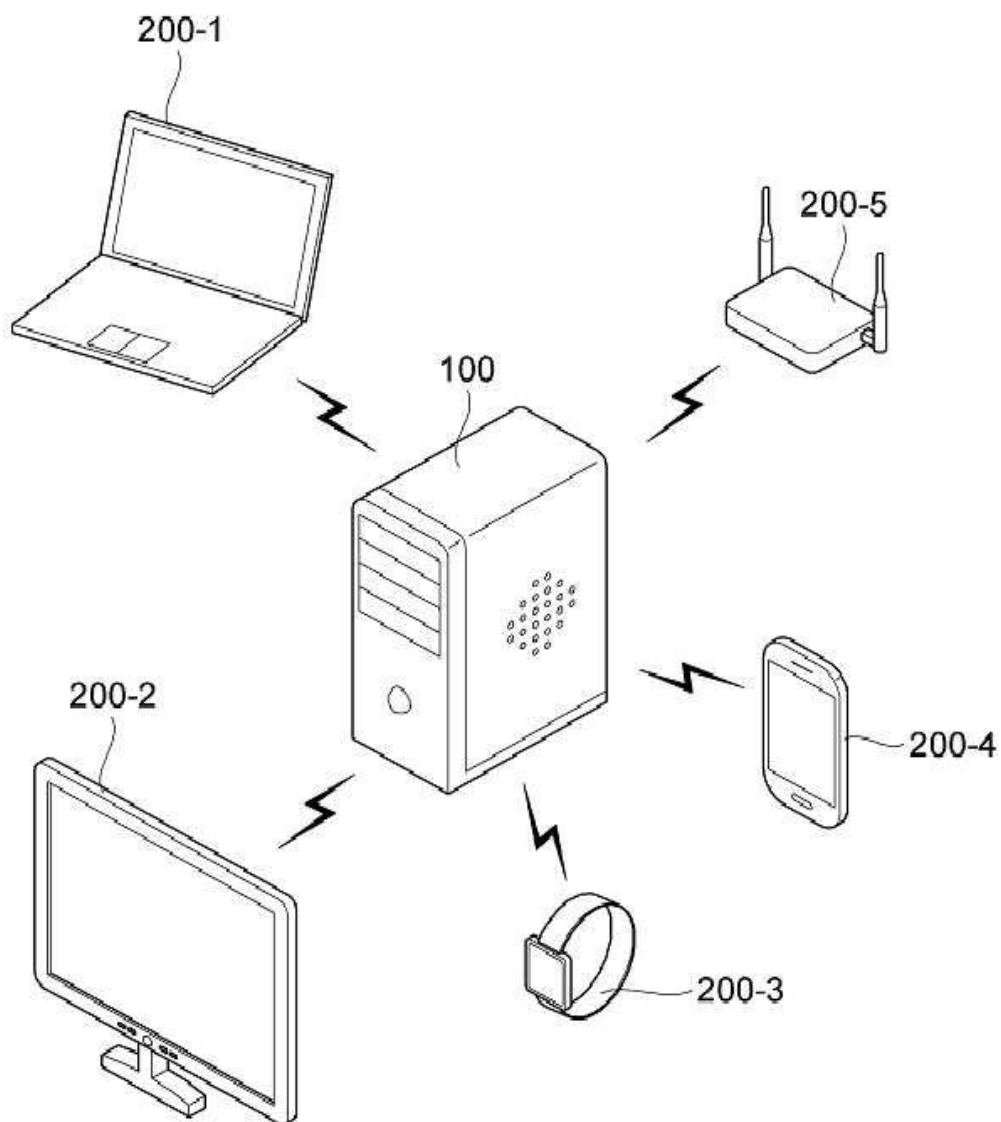
부호의 설명

[0117] 100 : 서버

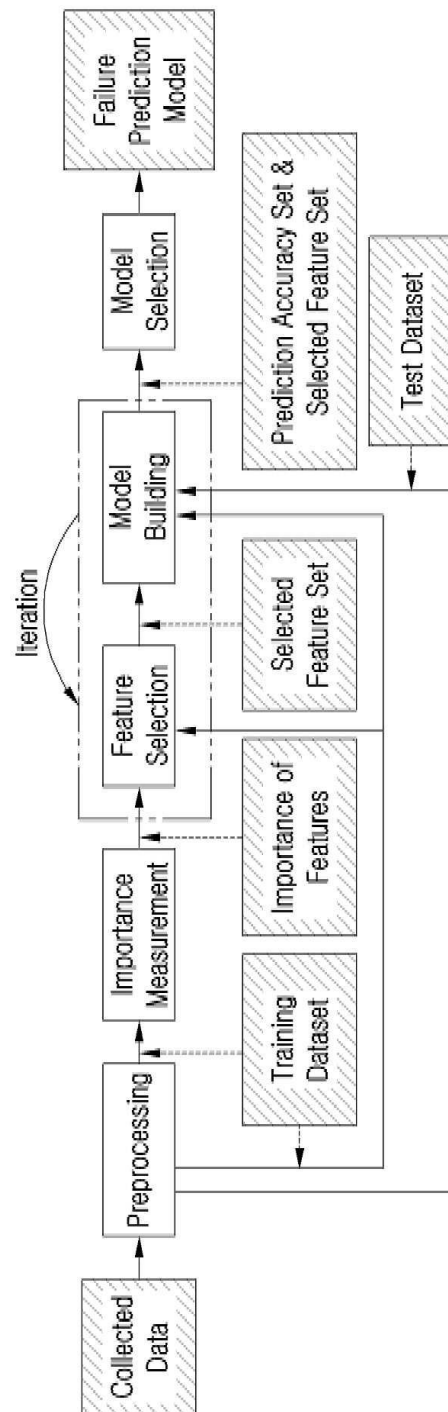
200-1 내지 200-5 : 복수의 IoT 장치

도면

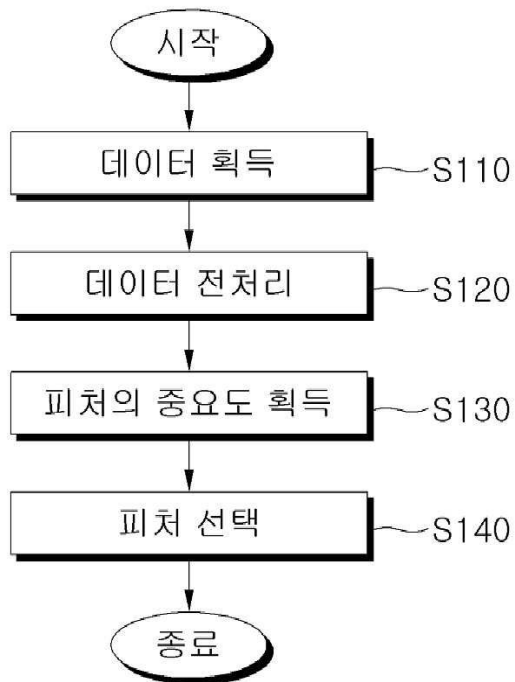
도면1



도면2



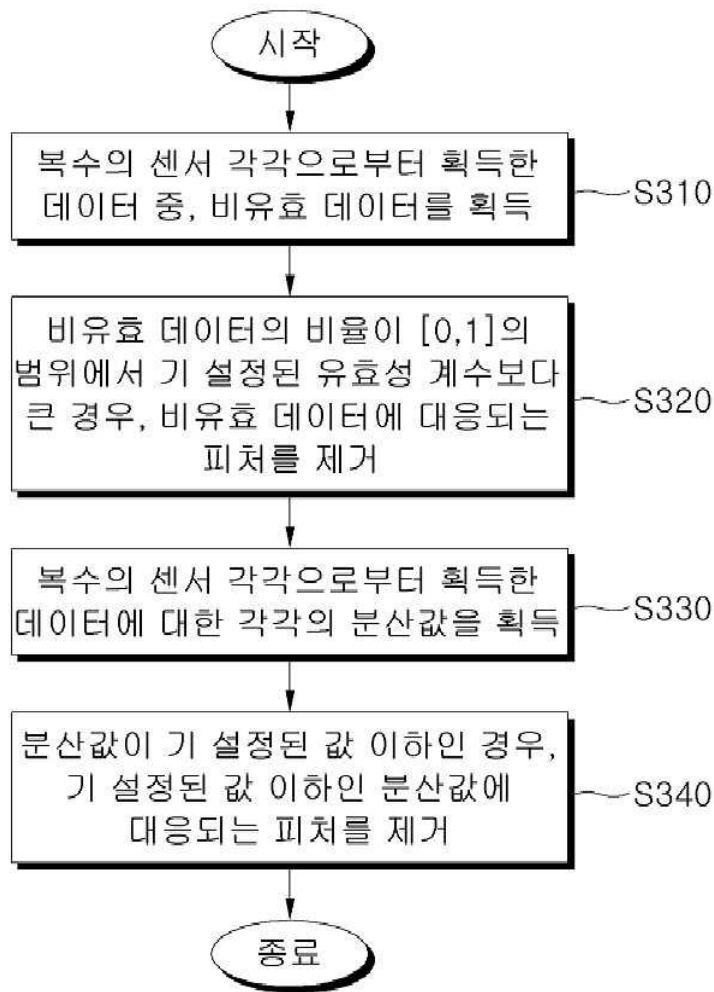
도면3



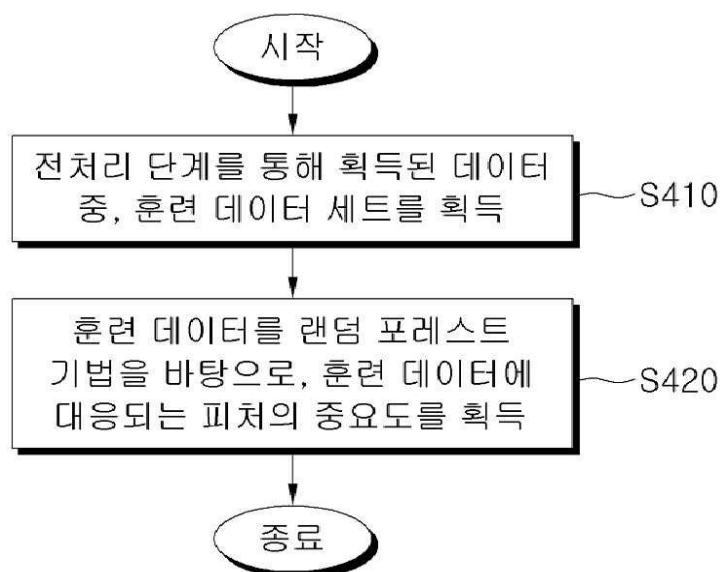
도면4



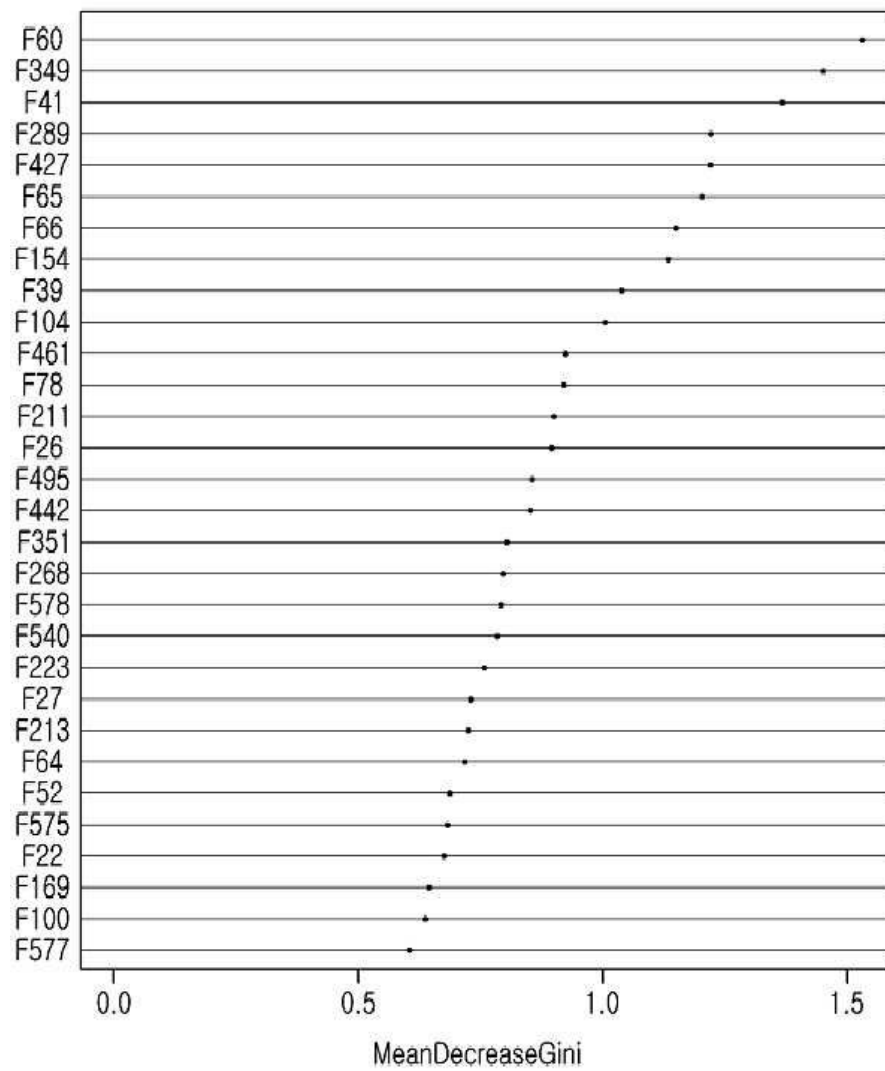
도면5



도면6



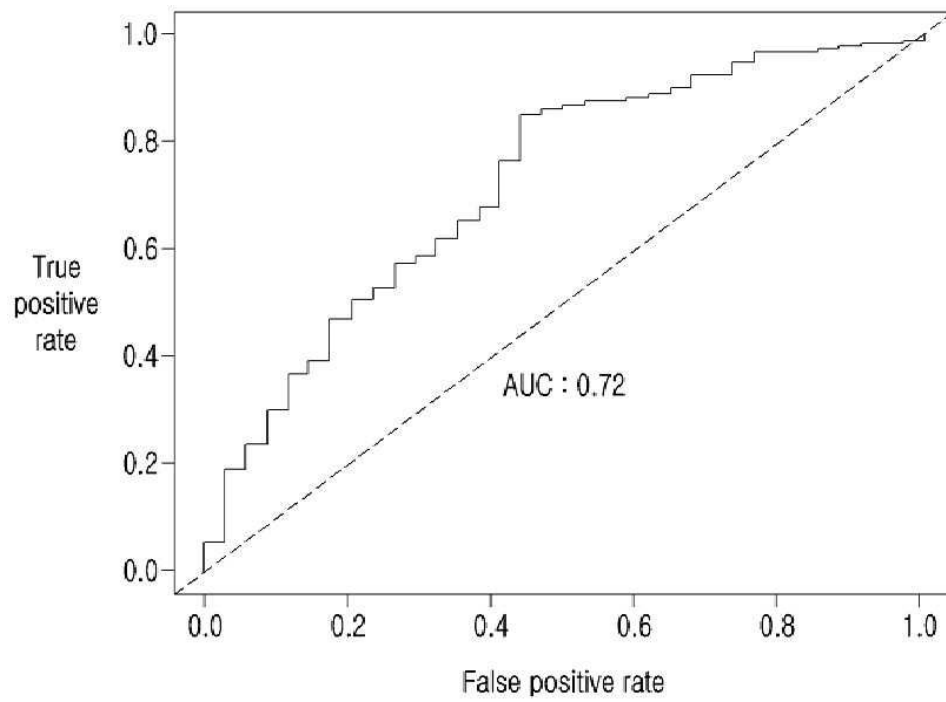
도면7



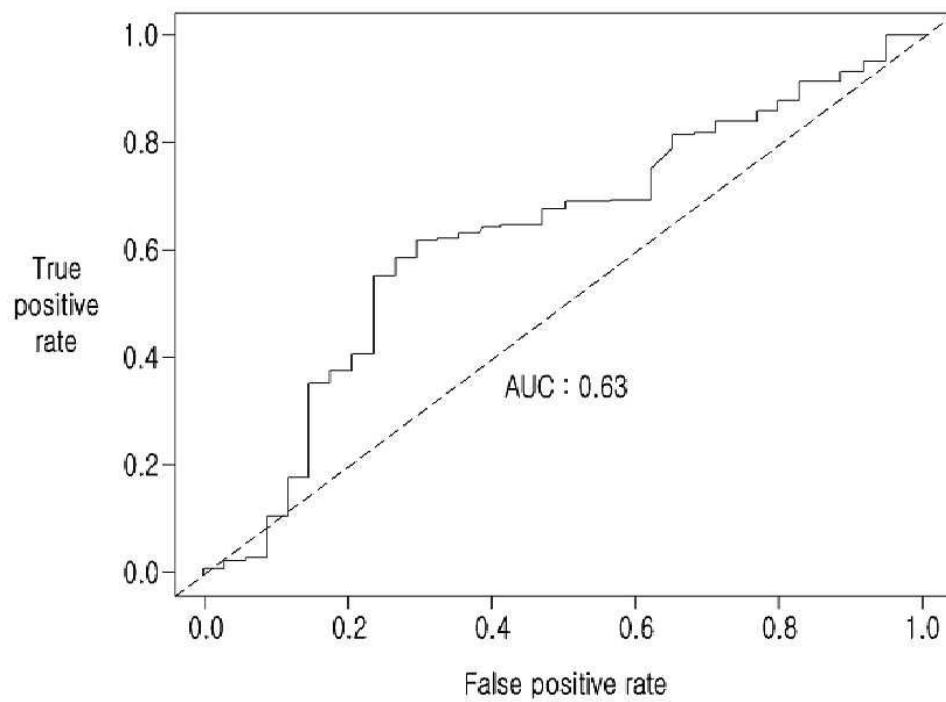
도면8

Index	0	1	2	3	4	5	6	7	8	9	10	11
SF	F60	F349	F41	F289	F427	F65	F66	F154	F39	F104	F461	F78
PA	0.69	0.64	0.66	0.7	0.7	0.69	0.71	0.72	0.71	0.7	0.67	0.71
Index	12	13	14	15	16	17	18	19	20	21	22	23
SF	F211	F26	F495	F442	F351	F268	F578	F540	F223	F27	F213	F64
PA	0.66	0.7	0.69	0.64	0.68	0.65	0.64	0.6	0.56	0.64	0.65	0.63

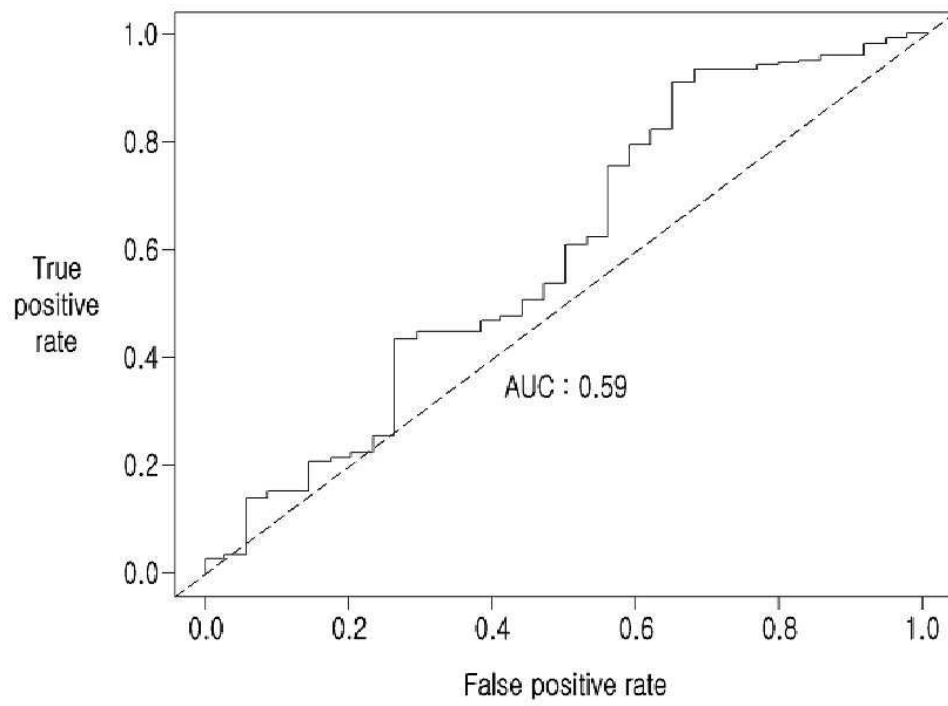
도면9



도면10



도면11



도면12

