관 인 생 략

출 원 번 호 통 지 서

출 원 일 자	2023.05.09
특 기 사 항	심사청구(유) 공개신청(무) 참조번호(IP230015N)
출 원 번 호	10-2023-0059859 (접수번호 1-1-2023-0513661-34) (DAS접근코드D1CE)
출원인 명칭	포항공과대학교 산학협력단(2-2004-043336-1)
대리인 성명	특허법인이상(9-2008-100021-0)
발명자 성명	홍원기 유재형 남석현
발명의 명칭	머신 러닝 기반 가상 머신 고장 예측 방법 및 장치

특 허 청 장

<< 안내 >>

1. 귀하의 출원은 위와 같이 정상적으로 접수되었으며, 이후의 심사 진행상황은 출원번호를 이용하여 특허로 홈페이지(www.patent.go.kr)에서 확인하실 수 있습니다.
2. 출원에 따른 수수료는 접수일로부터 다음날까지 동봉된 납입영수증에 성명, 납부자번호 등을 기재하여 가까운 은행 또는 우체국에 납부하여야 합니다.
※ 납부자번호 : 0131(기관코드) + 접수번호
3. 귀하의 주소, 연락처 등의 변경사항이 있을 경우, 즉시 [특허고객번호 정보변경(경정), 정정신고서]를 제출하여야 출원 이후의 각종 통지서를 정상적으로 받을 수 있습니다.
4. 기타 심사 절차(제도)에 관한 사항은 특허청 홈페이지를 참고하시거나 특허고객상담센터(☎ 1544-8080)에 문의하여 주시기 바랍니다.
※ 심사제도 안내 : https://www.kipo.go.kr-지식재산제도

【서지사항】

【서류명】 특허출원서

【참조번호】 IP230015N

【출원구분】 특허출원

【출원인】

【명칭】 포항공과대학교 산학협력단

【특허고객번호】 2-2004-043336-1

【대리인】

【명칭】 특허법인이상

【대리인번호】 9-2008-100021-0

【지정된변리사】 이재관, 전호진, 방영석, 권용남

【포괄위임등록번호】 2008-057306-7

【발명의 국문명칭】 머신 러닝 기반 가상 머신 고장 예측 방법 및 장치

【발명의 영문명칭】 METHOD AND APPARATUS OF VIRTUAL MACHINE FAILURE PREDICTION USING MACHINE LEARNING

【발명자】

【성명】 홍원기

【성명의 영문표기】 HONG, Won Ki

【주소】 경상북도 포항시 남구 청암로 77

【주소의 영문표기】 77, Cheongam-ro, Nam-gu, Pohang-si, Gyeongsangbuk-do

【발명자】

【성명】 유재형

【성명의 영문표기】 YOO, Jae Hyoung

【주민등록번호】 591118-1XXXXXX

【우편번호】 37673

【주소】 경상북도 포항시 남구 청암로 77

【발명자】

【성명】 남석현

【성명의 영문표기】 NAM, Suk Huyn

【주민등록번호】 961030-1XXXXXX

【우편번호】 37673

【주소】 경상북도 포항시 남구 청암로 77

【출원언어】 국어

【심사청구】 청구

【공지예외적용대상증명서류의 내용】

【공개형태】 논문

【공개일자】 2022.05.09

【공지예외적용대상증명서류의 내용】

【공개형태】 논문

【공개일자】 2022.10.31

【이 발명을 지원한 국가연구개발사업】

【과제고유번호】 1711193111

【과제번호】 2018-0-00749-006

【부처명】 과학기술정보통신부

【과제관리(전문)기관명】 정보통신기획평가원

【연구사업명】 방송통신산업기술개발(R&D,정보화)

【연구과제명】 인공지능 기반 가상 네트워크 관리기술 개발

【기여율】 1/1

【과제수행기관명】 포항공과대학교 산학협력단

【연구기간】 2023.01.01 ~ 2023.12.31

【취지】 위와 같이 특허청장에게 제출합니다.

대리인 특허법인이상 (서명 또는 인)

【수수료】

【출원료】	0	면	46,000 원
【가산출원료】	52	면	0 원
【우선권주장료】	0	건	0 원
【심사청구료】	20	항	1,023,000 원

【합계】 1,069,000 원

【감면사유】 전담조직(50%감면)[1]

【감면후 수수료】 534,500 원

【첨부서류】 1.공지예외적용대상(신규성상실의예외, 출원시의특례)규정을 적용받기 위한 증명서류[공지예외주장20220509]_1통 2.공지예외적용대상(신규성상실의예외, 출원시의특례)규정을

2023-05-09

적용받기 위한 증명서류[공지예외주장20221031]_1통

1 : 공지예외적용대상(신규성상실의예외,_출원시의특례)규정을_적용받기_위한_증명서류

PDF 파일 첨부

2 : 공지예외적용대상(신규성상실의예외,_출원시의특례)규정을_적용받기_위한_증명서류

PDF 파일 첨부

【발명의 설명】

【발명의 명칭】

머신 러닝 기반 가상 머신 고장 예측 방법 및 장치{METHOD AND APPARATUS OF VIRTUAL MACHINE FAILURE PREDICTION USING MACHINE LEARNING}

【기술분야】

<0001> 본 발명은 네트워크에 연결된 서버 및 서버에 설치된 가상머신의 고장 예측 방법에 관한 것으로, 보다 상세하게는 소프트웨어 정의 네트워크 및 네트워크 기능 가상화 환경에서 서버 별로 로그 분석을 통하여 가상 머신의 고장 발생 확률을 예측하는 방법 및 장치에 관한 것이다.

【발명의 배경이 되는 기술】

<0002> 소프트웨어 정의 네트워크(Software-Defined Network, SDN)는 네트워크 장치의 제어 평면(Control Plane)과 데이터 평면(Data Plane)을 분리하고, 복잡한 네트워크 제어 기능을 범용 컴퓨터에 소프트웨어로 구현하여 중앙 집중 방식으로 다수의 데이터 평면을 제어하도록 하는 기술이다.

<0003> 소프트웨어 정의 네트워크를 네트워크 기능 가상화(Network Function Virtualization, NFV) 기술과 함께 사용하면 값비싼 미들박스(Middle Box) 장비를 가상 서버 상에 소프트웨어로 구현함으로서 운용비용과 투자비용을 절감하고, 새로운 기능을 구현하는 시간을 단축할 수 있는 장점을 가진다. 이러한 기술은 5G 코어 네트워크와 공공 및 사설 클라우드 데이터 센터(Cloud Data Center)를 비롯한 다양한 환경에서 활발히 적용되고 있다.

<0004> 그러나, 이러한 SDN과 NFV 기술의 도입은 네트워크 구조를 더 복잡하게 만들었으며, 네트워크와 서버에 가상화 기술이 도입됨에 따라 네트워크와 서버의 장애를 진단하고 관리하는 것을 더욱 어렵게 만들었다. 특히 SDN과 NFV 환경에서 실제 사용자 응용 서비스를 처리하는 서버 및 가상 머신(Virtual Machine, VM)의 고장의 경우, 해당 고장을 즉시 대응하지 못하면 서비스 가용도(availability)의 저하로 이어져 고객의 불만을 초래하는 문제를 안고 있다.

<0005> 기존의 네트워크와 서버 관리 기술은 고장을 실시간으로 탐지하고 사후 조치를 빠르게 하는 것에 초점을 맞추고 있었지만, 최근의 5G 서비스나, 금융 서비스 등의 고가용도를 요구하는 서비스에서는 서비스 품질의 보장을 위해 고장을 미리 예측하고 서버 이전(server migration) 등의 사전 조치를 수행하는 방향으로 연구 개발이 진행되고 있다.

<0006> 대부분의 가상 머신의 고장은 가상 머신을 구동시키는 서버의 이상 상태에 기반한 것이며, 따라서 서버의 이상 상태와 관련된 고장 징후를 기반으로 예측해낼 수 있다. 특히, 서버 및 가상 머신의 고장 징후는 서버와 가상 머신이 만들어내는 다양한 로그(log) 데이터를 분석하여 찾아낼 수 있지만, 가상 머신과 서버가 출력하는 로그의 양이 매우 많기 때문에 네트워크 관리자가 이를 직접 분석하여 고장을 일일이 예측하기에는 어려움이 따른다. 또한 로그 데이터는 언어적 의미를 담은 텍스트 데이터이기 때문에 자동으로 고장 상태 분석을 하는 것에도 어려움이 따른다.

【발명의 내용】

【해결하고자 하는 과제】

<0007> 상기와 같은 종래 기술의 문제점을 해결하기 위한 본 발명의 목적은, 네트워크 기능 가상화 환경의 서버에 탑재된 가상 머신들의 로그를 머신러닝을 이용하여 자동으로 분석하여 서버 별로 가상 머신이 고장날 확률을 미리 예측하는 방법 및 장치를 제공하기 위한 것이다.

<0008> 본 발명의 다른 목적은, BERT(Bidirectional Encoder Representations from Transformers) 및 합성곱 신경망(Convolution Neural Network, CNN) 모델을 이용하여 서버에 탑재된 전체 가상 머신의 로그를 분석함으로써 학습된 적이 없는 가상 머신의 고장도 효과적으로 감지할 수 있는 서버별 가상 머신 고장 예측 방법 및 장치를 제공하는데 있다.

【과제의 해결 수단】

<0009> 상기 기술적 과제를 해결하기 위한 본 발명의 일 측면에 따른 가상 머신 고장 예측 방법은, 가상 머신을 서버 상에서 운용하고 가상 네트워크 기능(Virtual Network Function, VNF)을 가상 머신에서 운용하여 사용자 또는 클라이언트(client)에게 네트워크 기능을 제공하는 환경에서 사용되며, 다음의 구성을 가질 수 있다. 즉, 본 실시예의 가상 머신 고장 예측 방법은, 고장 여부가 태깅이 되어 있는 로그 데이터를 기반으로 사전 훈련을 하는 단계; 각 가상 머신(Virtual Machine, VM) 및 VMs를 동작시키는 서버로부터 발생하는 모든 로그를 수집하는 단계; 수집한 로그를 BERT(Bidirectional Encoder Representations from Transformers)의 입력으로 사용하여 로그의 특징을 나타내는 수치 행렬인 로그 임베딩 행렬로 변환하는 단계; 및 변환된 행렬을 기초로 사전 학습된 합성곱 신경

망(Convolution Neural Network, CNN)을 이용하여 서버별로 VM에 고장이 발생할 확률을 예측하는 단계를 포함한다.

<0010> 상기의 방법은, 수집한 로그를 전처리하는 단계를 더 포함할 수 있다.

<0011> 상기의 방법은, 상기 CNN을 이용할 때, CNN 모델과 BERT 모델의 역전파를 이용하여 BERT 모델과 CNN 모델을 미세튜닝하는 사전 훈련 단계를 더 포함할 수 있다.

<0012> 상기의 방법은, 상기 예측하는 단계에서 상기 일정 시간 내에 고장 발생이 예측된 가상머신에 대하여 고장 처리 방법을 결정하는 단계를 더 포함할 수 있다. 상기 고장 처리 방법을 결정하는 단계는, 고장 발생이 예측된 서버의 가상머신들을 고장 발생이 예측되지 않은 서버로 이전하는 단계를 포함할 수 있다. 즉, 상기의 방법은 가상머신 이전(migration) 단계를 더 포함할 수 있다.

<0013> 상기 기술적 과제를 해결하기 위한 본 발명의 다른 측면에 따른 가상 머신 고장 예측 방법은, 네트워크 기능 가상화 환경에서 동작하는 서버에 탑재되고 가상 네트워크 기능 또는 웹 서버를 운영하는 가상머신들 각각에서 발생하는 로그들을 상기 서버의 로그에 합한 로그 데이터를 획득하는 단계; 상기 로그 데이터를 사전 훈련된 언어 모델의 입력으로 사용하여 로그의 특징을 나타내는 수치 행렬인 로그 임베딩 행렬로 변환하는 단계; 및 상기 로그 임베딩 행렬을 입력으로 하는 합성곱 신경망 모델을 이용하여 서버별로 가상머신들에 일정 시간 이내에 고장이 발생할 확률을 예측하는 단계를 포함한다. 상기 로그 데이터는 상기 가상머신들의 로그들과 상기 가상머신들의 고장 이력들에 기초한 고장 관련 로그를 포함할 수 있다.

<0014> 상기 언어 모델은, 고장 여부가 태깅되어 있는 로그 데이터를 기반으로 사전 훈련된 후에, 상기 로그 데이터 내 문장의 맥락을 파악하여 같은 단어에 대하여 서로 다른 임베딩을 출력하는 모델일 수 있다.

<0015> 상기 언어 모델은, 상기 단어를 더 작은 하위 단어로 나누고 이를 토큰으로 사용하여 각 토큰에 대한 출력 임베딩을 생성하며, 상기 출력 임베딩이 생성을 위해 문장의 시작을 나타내는 특수 토큰과 위치 임베딩을 이용하는 모델일 수 있다.

<0016> 상기 언어 모델과 상기 합성곱 신경망 모델은, 사전 훈련 과정에서 역전파로 미세 튜닝되며, 상기 미세 튜닝에 의해 상기 언어 모델의 전체 하이퍼파라미터가 수정되도록 설치될 수 있다.

<0017> 상기 가상 머신 고장 예측 방법은, 상기 로그 데이터가 사전 훈련된 언어 모델에 입력되기 전에 상기 로그 데이터를 전처리하는 단계를 더 포함할 수 있다.

<0018> 상기 가상 네트워크 기능 또는 웹 서버를 운영하는 가상머신의 로그가 커널 로그일 때, 상기 전처리는 기설정된 규칙 기반으로 상기 커널 로그의 메이저 레지스터의 주소를 제거하는 것을 포함할 수 있다.

<0019> 상기 전처리는, 상기 로그들 중 미리 설정된 단어들 중 적어도 어느 하나 이상의 단어를 포함하는 로그들만을 남기는 것을 포함할 수 있다.

<0020> 상기 단어들은, 재설정(reset), 실패(fail), 아님(not), 고장(failure), 시간초과(timeout), 종료(kill), 오류(err), 메모리부족(out of memory, oom), 중지(stop), 종료(exit), 재시작(restart), 장시간(long) 및 경고(warn) 중 적어도 일부를 포함할 수 있다.

<0021> 상기 전처리는, 상기 로그들에서 시간 정보 및 애플리케이션 정보를 추출하는 과정, 상기 로그들에서 숫자와 기호를 제거하는 과정, 상기 로그들에서 대문자를 소문자로 변경하는 과정, 또는 이들의 조합 과정을 포함할 수 있다.

<0022> 상기 가상 머신 고장 예측 방법은, 상기 로그들에서 추출한 시간 정보에 기초하여 상기 서버의 로그와 상기 가상머신들의 로그들이 합해지는 입력 윈도우의 모든 로그들을 시간순으로 정렬하는 단계를 더 포함할 수 있다.

<0023> 상기 가상 머신 고장 예측 방법은, 상기 가상머신들 각각에서 발생하는 로그들과 상기 서버에서 발생하는 로그를 단일 입력 윈도우에서 합치는 단계; 및 상기 가상머신들 각각에서 발생하는 로그들과 상기 가상머신들 각각에 대한 네트워크 연결 상태의 확인 결과에 의해 생성되는 고장 이력들을 단일 고장 이력으로 합치는 단계;를 더 포함할 수 있다.

<0024> 상기 가상 머신 고장 예측 방법은, 상기 예측하는 단계에서 상기 일정 시간 내에 고장 발생 확률을 가진 가상머신에 대하여 고장 처리 방법을 결정하는 단계를 더 포함할 수 있다. 상기 고장 처리 방법은 상기 일정 시간 내에 고장 발생이 예측된 서버의 가상머신들을 상기 일정 시간 내에 고장이 예측되지 않은 서버로 이전하는 것을 포함할 수 있다.

<0025> 상기 가상머신들 중 적어도 일부는 서버의 종류와 트래픽 식별에 따라 동적으로 다른 서버와의 엔드투엔드 서비스를 제공하기 위해 서비스 기능 체이닝으로 구성된 복수의 가상 네트워크 기능들을 통과하도록 구성되고 상기 서비스 기능 체이닝을 위한 메시지 절차를 수행하도록 구성될 수 있다.

<0026> 상기 가상 머신 고장 예측 방법은, 상기 합성곱 신경망 모델을 학습시키기 위한 학습 데이터를 생성하는 단계를 더 포함할 수 있다. 상기 학습 데이터를 생성하는 단계는, 상기 가상머신들 각각에 일정 크기의 패킷 또는 테스트 신호를 보내고 상기 패킷 또는 테스트 신호에 대한 응답 메시지를 받고 상기 응답 메시지를 분석하여 상기 가상머신들 각각의 네트워크 연결 상태를 확인하는 것을 포함할 수 있다.

<0027> 상기 기술적 과제를 해결하기 위한 본 발명의 또 다른 측면에 따른 가상 머신 고장 예측 장치는, 머신 러닝 기반 가상 머신 고장 예측 방법을 수행하는 가상 머신 고장 예측 장치로서, 머신 러닝 기반으로 가상 머신의 고장을 예측하기 위해 메모리에 저장된 프로그램 명령들을 수행하는 적어도 하나 이상의 프로세서를 포함한다. 상기 적어도 하나 이상의 프로세서는, 네트워크 기능 가상화 환경에서 동작하는 서버에 탑재되고 가상 네트워크 기능 또는 웹 서버를 운영하는 가상머신들 각각에서 발생하는 로그들을 상기 서버의 로그에 합한 로그 데이터를 입력으로 사용하여 로그의 특징을 나타내는 수치 행렬인 로그 임베딩 행렬로 변환하는 사전 훈련된 언어 모델; 및 상기 로그 임베딩 행렬을 입력으로 하여 서버별로 가상머신들에 일정 시간 이내에 고장이 발생할 확률을 예측하는 합성곱 신경망 모델을 구비할 수 있다.

<0028> 상기 로그 데이터는 상기 가상머신들의 로그들과 상기 가상머신들의 고장 이력들에 기초한 고장 관련 로그를 포함할 수 있다.

<0029> 상기 사전 훈련된 언어 모델은, 고장 여부가 태깅되어 있는 로그 데이터를

기반으로 사전 훈련된 후에, 상기 로그 데이터 내 문장의 맥락을 파악하여 같은 단어에 대하여 서로 다른 임베딩을 출력하는 모델일 수 있다.

<0030> 상기 사전 훈련된 언어 모델과 상기 합성곱 신경망 모델은, 사전 훈련 과정에서 역전파로 미세 튜닝될 수 있다. 상기 미세 튜닝에 의해 상기 사전 훈련된 언어 모델의 전체 하이퍼파라미터가 수정될 수 있다.

<0031> 상기 가상 머신 고장 예측 장치는, 상기 로그 데이터가 상기 사전 훈련된 언어 모델에 입력되기 전에 상기 로그 데이터를 전처리하는 전처리 유닛을 더 포함할 수 있다. 상기 전처리 유닛은, 상기 가상 네트워크 기능 또는 웹 서버를 운영하는 가상머신의 로그가 커널 로그일 때, 기설정된 규칙 기반으로 상기 커널 로그의 메이저 레지스터의 주소를 제거하도록 구성될 수 있다. 상기 전처리 유닛은, 상기 로그들 중 미리 설정된 단어들 중 적어도 어느 하나 이상의 단어를 포함하는 로그들만을 남기도록 구성될 수 있다. 또한, 상기 전처리 유닛은, 상기 로그들에서 시간 정보 및 애플리케이션 정보를 추출하는 과정, 상기 로그들에서 숫자와 기호를 제거하는 과정, 상기 로그들에서 대문자를 소문자로 변경하는 과정, 또는 이들의 조합 과정을 수행하도록 구성될 수 있다.

<0032> 상기 가상 머신 고장 예측 장치는, 상기 로그 데이터를 생성하는 모니터링 노드를 더 포함할 수 있다. 상기 모니터링 노드는, 상기 가상머신들 각각에서 발생하는 로그들과 상기 서버에서 발생하는 로그를 단일 입력 윈도우로 합치고, 상기 가상머신들 각각에서 발생하는 로그들과 상기 가상머신들 각각에 대한 네트워크 연결 상태의 확인 결과에 의해 생성되는 상기 가상머신들 각각의 고장 이력들을 단일

고장 이력으로 합치도록 구성될 수 있다.

<0033> 상기 모니터링 노드는, 상기 로그들에서 추출한 시간 정보에 기초하여 상기 서버의 로그와 상기 가상머신들의 로그들이 합해지는 입력 윈도우의 모든 로그들을 시간순으로 정렬하도록 구성될 수 있다.

<0034> 상기 프로세서는, 상기 일정 시간 내에 고장 발생 확률을 가진 가상머신에 대하여 고장 처리 방법을 결정하도록 구성될 수 있다. 상기 고장 처리 방법은 상기 일정 시간 내에 고장 발생이 예측된 서버의 가상머신들을 상기 일정 시간 내에 고장 발생이 예측되지 않은 서버로 이전하는 것을 포함할 수 있다.

<0035> 상기 프로세서는, 상기 합성곱 신경망 모델의 훈련을 위한 훈련 데이터를 생성할 수 있다. 훈련 데이터를 생성하기 위해, 상기 프로세서는 상기 가상머신들 각각에 일정 크기의 패킷 또는 테스트 신호를 보내고 상기 패킷 또는 테스트 신호에 대한 응답 메시지를 받고, 상기 응답 메시지를 분석하여 상기 가상머신들 각각의 네트워크 연결 상태를 확인하는 상태 검사기를 더 구비할 수 있다.

【발명의 효과】

<0036> 전술한 본 발명에 의하면, 서버 별로 가상 머신에 고장이 발생하기 적어도 수 분 정도 내지 수십 분 정도 이전에 고장을 예측하여 가상 머신의 고장을 사전에 조치할 수 있는 효과가 있다.

<0037> 또한, 본 발명에 의하면, 자연 언어 처리 분야에서 문장의 의미를 파악하여 단어 임베딩을 생성하는 BERT와 함께 입력 정보를 손실하지 않으면서 판단하는 합성곱 신경망을 통해 서버 및 가상 머신에서 발생하는 대량의 로그 메시지를 자동으

로 효과적으로 분석하여 서버나 가상머신 오류와 관련된 메시지를 신속하게 파악하여 대처할 수 있는 장점이 있다.

【도면의 간단한 설명】

<0038> 도 1은 본 발명의 일실시예에 따른 가상 머신 고장 예측 시스템의 오픈 스택 기반 네트워크 기능 가상화 환경을 설명하기 위한 개략적인 블록도이다.

도 2는 도 1의 가상 머신 고장 예측 시스템에서 로그 데이터의 수집 과정을 설명하기 위한 예시도이다.

도 3은 도 1의 가상 머신 고장 예측 시스템의 모니터링 노드에 채용할 수 있는 구성을 설명하기 위한 블록도이다.

도 4는 도 1의 가상 머신 고장 예측 시스템에 채용할 수 있는 로그 데이터 생성 과정에서 서버 별로 고장 데이터를 태깅하는 과정을 설명하기 위한 예시도이다.

도 5는 도 1의 가상 머신 고장 예측 시스템의 고장 예측 모듈 내 BERT(Bidirectional Encoder Representations from Transformers) 모듈에 채용할 수 있는 구성을 설명하기 위한 블록도이다.

도 6은 도 5의 BERT 모듈에 채용할 수 있는 BERT 토큰화기 및 BERT 모델의 구조 및 고장 예측 기계 학습 알고리즘을 설명하기 위한 예시도이다.

도 7은 도 1의 가상 머신 고장 예측 시스템의 고장 예측 모듈 내 CNN 모델에 채용할 수 있는 구성과 BERT 모델과 결합관계를 설명하기 위한 블록도이다.

도 8은 도 7의 CNN 모델에 채용할 수 있는 구조 및 고장 예측 기계 학습 알

고리즘을 설명하기 위한 예시도이다.

도 9는 본 발명의 다른 실시예에 따른 가상 머신 고장 예측 방법을 설명하기 위한 흐름도이다.

도 10은 본 발명의 또 다른 실시예에 따른 가상 머신 고장 예측 장치에 채용할 수 있는 네트워크 기능 가상화 인프라(Network Function Virtualization Infrastructure, NFVI) 환경을 설명하기 위한 예시도이다.

도 11은 본 발명의 또 다른 실시예에 따른 가상 머신 고장 예측 장치에 채용할 수 있는 구성을 설명하기 위한 개략적인 블록도이다.

【발명을 실시하기 위한 구체적인 내용】

<0039> 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

<0040> 제1, 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 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 중

의 어느 항목을 포함한다.

<0041> 본 출원의 실시예들에서, 'A 및 B 중에서 적어도 하나'는 'A 또는 B 중에서 적어도 하나' 또는 'A 및 B 중 하나 이상의 조합들 중에서 적어도 하나'를 의미할 수 있다. 또한, 본 출원의 실시예들에서, 'A 및 B 중에서 하나 이상'은 'A 또는 B 중에서 하나 이상' 또는 'A 및 B 중 하나 이상의 조합들 중에서 하나 이상'을 의미할 수 있다.

<0042> 어떤 구성요소가 다른 구성요소에 '연결되어' 있다거나 '접속되어' 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 '직접 연결되어' 있다거나 '직접 접속되어'있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.

<0043> 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, '포함한다' 또는 '가진다' 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

<0044> 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서

사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가진 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

<0045> 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 본 발명을 설명함에 있어 전체적인 이해를 용이하게 하기 위하여 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.

<0046> 도 1은 본 발명의 일실시예에 따른 가상 머신 고장 예측 시스템의 오픈 스택 기반 네트워크 기능 가상화 환경을 설명하기 위한 개략적인 블록도이다. 도 2는 도 1의 가상 머신 고장 예측 시스템에서 로그 데이터의 수집 과정을 설명하기 위한 예시도이다. 그리고, 도 3은 도 1의 가상 머신 고장 예측 시스템의 모니터링 노드에 채용할 수 있는 구성을 설명하기 위한 블록도이다.

<0047> 도 1을 참조하면, 가상 머신 고장 예측 시스템은 컨트롤러 노드(Controller Node, 100), 컴퓨팅 노드(Computing Node 또는 Compute Node, 200), 모니터링 노드(Monitoring Node, 300) 및 고장 예측 모듈(Failure Prediction Module, FPM, 500)을 포함할 수 있다. 컨트롤러 노드(100), 컴퓨팅 노드(200), 모니터링 노드(300) 및 고장 예측 모듈(500) 각각은 서버 또는 컴퓨팅 장치에 의해 독립적으로 설치될 수 있고, 네트워크나 통신망을 통해 서로 연결되어 신호 및 데이터를 송수

신하도록 구성될 수 있다.

<0048> 가상 머신 고장 예측 시스템은 클라우드 컴퓨팅 관리 시스템으로 많이 사용되는 오픈스택(OpenStack)과 네트워크 기능 가상화 인프라(Network Function Virtualization Infrastructure, NFVI)를 사용하여 구축될 수 있다.

<0049> 본 명세서에서, 가상 머신 고장 예측 시스템에 포함되는 가상 머신 고장 예측 장치는, 고장 예측 모듈(500)을 포함하여 구성되거나, 고장 예측 모듈(500)과 모니터링 노드(300)를 포함하여 구성될 수 있다. 이러한 고장 예측 모듈(500) 또는 이를 포함하는 가상 머신 고장 예측 장치는, 학습이 완료된 이후에 특정 컴퓨팅 장치의 서버 및/또는 해당 서버의 가상 머신의 고장 예측에 적용하도록 구성될 수 있다.

<0050> 가상 머신 고장 예측 시스템의 각 구성요소를 좀더 구체적으로 설명하면 다음과 같다.

<0051> 먼저, 컨트롤러 노드(100)는 컴퓨팅 노드(200)의 각 가상 머신(Virtual Machine, VM)의 네트워크 연결을 관리하고, 컴퓨팅 노드(200)의 서비스 기능 체이닝(Service Function Chaining, SFC)을 설정할 수 있다.

<0052> 컨트롤러 노드(100)는 사전에 결정된 네트워크 관리 정책(policy)을 컴퓨팅 노드(200)로 전달하여 가상 네트워크 기능 및/또는 서비스 기능 체이닝의 구성 및 배치를 변화시킬 수 있다. 네트워크 관리 정책(policy)은 후술하는 데이터 수집 단계에서 컴퓨팅 노드(200)에 적용되지 않을 수 있다.

<0053> 컴퓨팅 노드(200)는 프로세싱 리소스, 메모리, 주변 장치, 네트워킹 모듈 등

을 구비하고, 프로세싱 리소스의 각 서버에서 독립적으로 운영되는 가상 머신들 상에서 다수의 가상 네트워크 기능(Virtualized Network Function, VNF)을 동작시키도록 구성될 수 있다.

<0054> 일례로, 컴퓨팅 노드(200)는 제1 서버(210), 제5 서버(250) 등의 복수의 서버들을 구비할 수 있다. 제1 서버(210)에는 제1 가상머신(VM A, 211), 제2 가상머신(VM B, 212), 제5 가상머신(VM E, 215) 등이 설치될 수 있다(도 2 참조). 이와 유사하게 제5 서버(250)에도 복수의 가상머신들(예컨대, VM F, VM G 등)이 설치될 수 있다.

<0055> 컴퓨팅 노드(200)는 도 2에 나타낸 바와 같이 상태 점검기(state checker, 217)를 더 구비할 수 있다. 상태 점검기(217)는 제1 서버(210)에 설치된 가상머신으로 구현될 수 있으나, 이에 한정되지 않고, 컴퓨팅 노드(200)의 다른 서버에 설치되거나, 컴퓨팅 노드(200)에서 해당 기능을 수행하는 독립적인 수단이나 구성부로 구현될 수 있다.

<0056> 컴퓨팅 노드(200)의 각 서버, 예를 들어 제1 서버(server 1, 210)에서 발생하는 로그(server 1 log), 제1 가상머신(211)에서 발생하는 로그(VM A log), 제2 가상머신(212)에서 발생하는 로그(VM B log), 제5 가상머신(215)에서 발생하는 로그(VM E log) 등의 로그(Log)는 모니터링 노드(300)에 의해 수집될 수 있다. 그리고, 상태 점검기(217)에 의해 확인된 가상머신들의 고장 이력(VMs failure history)도 모니터링 노드(300)에 의해 수집될 수 있다(도 3 참조).

<0057> 또한, 컴퓨팅 노드(200)는 가상 네트워크 기능들을 이용하여 서비스 기능 체

이닝(Service Function Chaining, SFC)을 배치하고, 이를 통해 가입자 또는 클라이언트에게 네트워크 기능 가상화(NFV) 기반의 네트워크 기능을 제공할 수 있다.

<0058> 모니터링 노드(300)는 네트워크 기능 가상화 환경에서 데이터를 수집하고 모니터링하는 역할을 수행한다. 모니터링 노드(300)는 로그 및 고장 이력에 대한 데이터를 저장할 수 있다. 이를 위해, 모니터링 노드(300)는 도 3에 나타낸 바와 같이 로그(Log)를 수집하는 알시스로그 서버(rsyslog server, 320)와 고장 이력 데이터베이스(Failure History DB, 340)를 구비할 수 있다. 용어 'rsyslog server'는 'The rocket-fast syslog server'의 약칭일 수 있다.

<0059> 즉, 모니터링 노드(300)는 서버 별 가상 머신의 고장 예측에 로그 데이터를 이용하기 위하여 로그 수집 유틸리티인 알시스로그 서버(320)를 활용하여 로그를 수집할 수 있다. 그리고 수집된 로그는 모니터링 노드(300)의 저장부에 저장될 수 있다. 일례로, 모니터링 노드(300)는 컴퓨팅 노드(200)의 각 서버(210, 250) 및 각 서버(210, 250)에서 동작하는 가상머신들(예컨대, 211, 212, 215)로부터 로그(log)를 수집할 수 있다.

<0060> 모니터링 노드(300)에 수집되는 로그는 가상 머신에서 발생하는 커널(Kernel) 로그, 애플리케이션(Application) 로그, 시스템 데몬(Systemd) 로그 등을 포함할 수 있다.

<0061> 또한, 모니터링 노드(300)에 수집되는 로그는, 서버의 NFVI 구축에 사용된 클라우드 컴퓨팅 관리 시스템에서 발생하는 서버 애플리케이션 로그를 포함할 수 있다. 예를 들어, 초당 거의 10 라인들의 로그를 생성하는 서버에서 동작하는 오픈

스택 관련 애플리케이션들의 경우, 수집되는 로그는, 네트워크 관리에 사용되는 소프트웨어인 뉴트론(Neutron)의 로그, 노바(Nova)의 로그, 오픈 브이스위치(Open vSwitch, OVS)의 로그 등을 포함할 수 있다.

<0062> 다시 도 1 내지 도 3을 참조하면, 고장 예측 모듈(500)은 모니터링 노드(300)로부터 로그 데이터를 받을 수 있다. 로그 데이터는 가상머신들(211, 215)의 로그들과 가상머신들(211, 212, 215)의 고장 이력들에 기초한 고장 관련 로그를 포함할 수 있다. 그리고, 고장 예측 모듈(500)은 서버 별 가상 머신의 고장 발생 확률을 예측한 후, 예측 결과로 얻어지는 가상 머신의 고장 정보(Failure info)를 컨트롤러 노드(100)에 제공할 수 있다.

<0063> 전술한 기능들을 수행하기 위해, 고장 예측 모듈(500)은 BERT 모듈(520) 및 CNN 모델(530)을 구비할 수 있다. 또한, 고장 예측 모듈(500)은 전처리(Preprocessing) 유닛(510)을 더 구비할 수 있다. 전처리 유닛(510), BERT 모듈(520) 및 CNN 모델(530)의 구조 및 기능에 대하여는 아래에서 상세히 설명하기로 한다.

<0064> 이와 같이, 본 실시예의 가상 머신 고장 예측 시스템에 채용되는 가상 머신 고장 예측 방법을 학습하기 위해서는, 기계 학습(Mechine leanrnig) 모델의 입력으로 사용되는 각 가상 머신 및 서버의 로그들 뿐 아니라 고장 발생 여부에 대한 데이터 또는 정보를 수집하는 것이 필요하다. 이를 위해, 본 실시예의 학습 과정에서는 단일 서버 내 가상 머신들 중 적어도 하나를 상태 점검기(State Checker)로 설치하고, 나머지 가상 머신들에 대해 일정 주기 예를 들어, 수십 초에서 수 분 사이

서 선택되는 시간(바람직하게는 1분)마다 핑(ping) 메시지를 보내 각 가상 머신의 네트워크 기능에 고장이 발생하는지 않았는지 확인할 수 있다. 핑 메시지는 일정 크기의 패킷 또는 테스트 신호에 대응될 수 있다.

<0065> 만약, 핑 메시지에 응답이 없으면, 상태 점검기(217)은 해당 가상 머신을 고장으로 분류하고 고장이 난 시각을 저장할 수 있다. 각 가상 머신별로 실시간 고장 여부를 기록한 가상머신 고장 이력 데이터(이하 간략히 '고장 이력 데이터'라고도 한다)는 상태 점검기(217)로부터 모니터링 노드(300)로 전달될 수 있다. 이때, 고장 이력 데이터는 해당 가상 머신에 장애가 발생한 시간을 기록하기 위해 yml 또는 yaml 확장자를 사용하고 특정 구문 규칙에 따르는 YAML 파일 등의 미리 설정된 형식으로 저장될 수 있다.

<0066> 가상 머신의 고장 여부를 기록하는 기능이나 이러한 고장 여부 기록 기능을 구현하는 구성부는, 사전 훈련 과정 이후에 본 실시예의 가상 머신 고장 예측 방법을 실행하여 서버 별 가상 머신의 고장 발생 확률을 예측하는 장치의 테스트 동작 시 생략될 수 있다.

<0067> 또한, 본 실시예의 가상 머신 고장 예측 시스템은 각 서버의 가상 머신의 고장을 가상 머신 별로 예측하지 않고, 서버 별로 예측하도록 구성된다. 그 이유는 가상 머신 환경이 워낙 복잡하고 민감한 시스템을 이루고 있어 가상 머신의 고장 원인이 다양하기 때문이다.

<0068> 예를 들어, 특정 가상 머신의 커널 이상 상태로 인해 서버에 이상 상황이 발생할 수 있으며, 이로 인해 해당 가상 머신이 아닌 같은 서버에서 동작하는 다른

가상 머신의 고장으로 발전할 수 있다. 이러한 종류의 고장은, 각 가상 머신 별로 가상 머신의 로그를 기반으로 예측될 수 없기 때문에, 본 실시예에서는 같은 서버에서 동작하는 모든 가상 머신들의 로그를 이용하여서 서버 별로 가상 머신에 고장이 발생할 확률을 예측한다.

<0069> 수집된 로그와 고장 이력 데이터는 본 실시예의 고장 예측 기계 학습 모델의 입력 로그 데이터(이하 간략히 '입력 데이터'라고도 한다)를 생성하는데 이용되고, 입력 데이터에 대응하는 출력 데이터와 함께 고장 예측 기계 학습 모델을 학습시키는데 이용될 수 있다. 특히, 본 실시예의 가상 머신 고장 예측 시스템은, 서버 별로 가상 머신의 고장 확률을 예측하기 때문에, 서버 별 가상 머신의 로그의 기록 시간과 고장 이력 데이터의 기록 시간을 토대로 서버 별 가상 머신의 로그를 미리 설정된 시간 동안 즉, 개별 가상 머신의 로그들이 기록되는 개별 윈도우의 미리 설정된 윈도우 크기(window size) 동안 수집된 각 가상 머신의 로그들을 서버 별로 합치도록 구성될 수 있다.

<0070> 수집된 서버 로그와 가상 머신 로그를 서버 별로 합치는 과정은, 미리 설정된 가상 머신들의 일련 순서에 따라 해당 서버 로그와 함께 가상 머신들의 로그들을 순차적으로 합치도록 구성될 수 있다.

<0071> 또한, 로그를 서버 별로 합치는 과정은, 각 가상 머신의 로그가 기록된 시간이나 타임스탬프를 기준으로 가상 머신들의 로그들을 단일 입력 윈도우로 합치도록 구성될 수 있다. 즉, 로그를 서버 별로 합치는 과정은, 로그들에서 추출한 시간 정보에 기초하여 서버의 로그와 가상머신들의 로그들이 합해지는 입력 윈도우에서 모

든 로그들이 시간순으로 정렬하도록 구성될 수 있다.

<0072> 도 4는 도 1의 가상 머신 고장 예측 시스템에 채용할 수 있는 로그 데이터 생성 과정에서 서버 별로 고장 데이터를 태깅하는 과정을 설명하기 위한 예시도이다.

<0073> 도 4를 참조하면, 가상 머신 고장 예측 시스템에서는, 기계 학습 모델을 학습시키기 위해 입력 윈도우(input window)에 들어온 로그 데이터를 기계 학습 모델의 입력으로 사용하고, 이로부터 일정 시간 차(gap) 이후의 고장 여부를 기계 학습 모델의 출력으로 사용하도록, 기계 학습 모델의 입력 데이터와 출력 데이터를 생성할 수 있다.

<0074> 특히, 가상 머신 고장 예측 시스템은, 서버 별 고장 데이터를 만들기 위해 특정 서버에서 동작하는 모든 가상 머신의 로그들 및 해당 서버의 로그를 각 로그의 타임스탬프를 기반으로 입력 윈도우 동안 합쳐 입력 데이터를 생성할 수 있다.

<0075> 이때, 입력 윈도우는 사전에 설정한 입력 윈도우 크기(input window size)를 갖는 버퍼 메모리이며, 입력 윈도우 크기와 가상머신들의 네트워크 상태가 기록되는 고장 이력과의 예측 시간 차이(gap)는 사용자가 학습 과정에서 조정가능한 인자(parameter)에 해당할 수 있다. 즉, 입력 윈도우는 로그 데이터를 슬라이드하여 생성되므로, 슬라이딩 윈도우 크기를 조절하여 로그의 양을 조절할 수 있다.

<0076> 예를 들어, 입력 윈도우 크기로 10분, 슬라이딩 크기로 5분을 사용하는 경우, 10분의 입력 윈도우 내의 로그는 입력 데이터가 된다. 각 입력 로그는 입력 윈도우가 종료된 시점부터 '갭' 기간 이후에 고장 이력이 'failure'로 표시되면 고장

으로 라벨링되고, 그 간격 이후에 정상이면 정상으로 라벨링된다.

<0077> 또 다른 예로써, 입력 윈도우 크기로 20분을, 예측 시간 차이로 10분을 각각 설정하여 사용하는 경우, 12시 10분부터 12시 30분까지의 서버의 각 가상 머신 별 로그 및 서버 로그를 모두 입력 데이터로 사용하여 12시 40분에 고장이 발생할지 여부를 예측할 수 있다.

<0078> 즉, 기계 학습의 출력 데이터는 해당 서버에서 동작하는 가상 머신들 중 하나라도 고장이 발생했는지를 기반으로 새롭게 태깅(tagging)되는 데이터를 포함하도록 생성될 수 있다. 새로 태깅되는 데이터는 예를 들어 '고장(failure)'으로서 라벨링되는 데이터를 포함할 수 있다.

<0079> 또 예를 들면, 도 4에 나타낸 바와 같이, 제1 입력 윈도우(211g)에 수집되는 제1 가상 머신의 로그들과, 제5 입력 윈도우(215g)에 수집되는 제5 가상 머신의 로그들을 포함한 하나의 특정 서버 내 모든 가상 머신의 로그들과, 서버 로그의 입력 윈도우(210g)에 수집되는 서버 로그가 단일 입력 윈도우(WSa, 216g)(이하 '통합 입력 윈도우'라고 한다)로 합쳐질 수 있다.

<0080> 또한, 제1 가상 머신의 고장 이력(211f)과 제5 가상 머신의 고장 이력(215f)을 포함한 모든 가상 머신의 고장 이력들은 단일 고장 이력(218f)(이하 '통합 고장 이력'이라고 한다)으로 합쳐질 수 있다. 이때, 각 가상 머신의 로그들은 해당 가상 머신들의 고장 이력들(211f, 215f)을 기반으로 서버의 고장 관련 로그로 새로 생성된 것을 볼 수 있다. 특정 서버의 고장 관련 로그는 'failure'로 라벨링된 고장 히스토리(failure history)가 해당 타임스탬프의 시간에 따라 순차적으로

기록될 수 있다.

<0081> 한편, 입력 데이터로 생성된 입력 로그 데이터는 고장 예측 모듈(500)의 기계 학습의 입력으로 이용하기 전에 전처리 과정(pre-processing)을 거칠 수 있다. 전처리 과정은 전처리 유닛에 의해 수행될 수 있다. 전처리 유닛의 기능이 모니터링 노드의 후단이나 BERT 모듈의 전단에 통합되는 경우, 전처리 유닛은 생략될 수 있다.

<0082> 본 실시예의 고장 예측 과정에서는, 단일 가상 머신의 로그뿐만 아니라 다수의 가상 머신들의 로그들과 다수의 가상 머신들의 동작하는 서버의 로그도 함께 입력으로 사용하므로, 입력 로그의 개수가 상대적으로 매우 많다. 특히, 서버에서 동작하는 오픈 스택(openstack) 관련 어플리케이션(eg. Neutron, Nova, OVS)은 초당 거의 10줄의 로그를 생성한다. 이 경우, 기계 학습 모델의 성능은 차치하고 로그를 읽는 것만으로도 시간이 오래 걸리므로 로그 수를 줄이기 위한 전처리가 요구된다.

<0083> 이러한 요구에 부응하기 위해, 본 실시예의 전처리 과정에서는 로그 데이터에서 불필요한 로그를 제거하도록 구성될 수 있다. 전처리 과정은, 날짜 및 로그를 발생시킨 어플리케이션 이름(Application Name), 프로세스 이름(Process Name) 등을 미리 설정된 규칙 기반(Rule-based)으로 제거하는 것이나, 미리 설정된 규칙 기반으로 숫자 및/또는 기호를 제거하는 것을 포함할 수 있다.

<0084> 또한, 전처리 과정은, 가상 머신의 로그가 커널 로그인 경우, 주기적으로 출력되는 커널 로그의 메이저 레지스터의 주소(addresses of major registers)를 제거하도록 구성될 수 있다. 이러한 커널 로그의 전처리 과정은 별도의 배경 지식이

없어도 로그에서 불필요한 패턴을 찾는 간단한 과정이기 때문에 새로운 유형의 로그가 있어도 쉽게 적용할 수 있다.

<0085> 게다가, 가상 머신 고장 예측 시스템에서는 고장 예측에 사용되는 서버의 컴퓨팅 자원이 부족한 경우, 효율을 높이기 위하여 일부 로그를 제거하여 입력 데이터의 양을 줄이는 작업을 진행할 수 있다. 로그 제거 작업은 미리 설정된 규칙 기반으로 수행될 수 있다. 예를 들어, 고장과 전혀 관련 없는 로그, 예를 들어, 서버 원격로그인(Secure Shell, ssh) 로그인 로그, 패키지 업데이트 로그 등을 미리 설정된 규칙 기반으로 제거하도록 구성될 수 있다.

<0086> 전술한 과정을 거친 후에도 로그의 양이 너무 많아 학습이 잘 되지 않는 경우에는 고장과 관련될 확률이 높은 로그만 규칙 기반으로 남길 수 있다. 예를 들어, 수집한 데이터 중 고장이 발생했을 때의 로그 데이터에서 가장 빈번하게 출력되는 단어들의 목록을 미리 뽑을 수 있고, 이를 기반으로 해당 단어가 포함된 로그들만 남길 수 있다. 이러한 단어들에는 일례로 재설정(reset), 실패(fail), 아님(not), 고장(failure), 시간초과(timeout), 종료(kill), 오류(err), 메모리부족(out of memory, oom), 중지(stop), 종료(exit), 재시작(restart), 장시간(long), 및 경고(warn)가 포함될 수 있다.

<0087> 전술한 다양한 형태의 전처리 과정은 고장 예측 모듈을 사전학습시키는 단계에서 가상 머신 고장 예측 시스템에 접속가능한 네트워크 관리자에 의해 해당 모듈의 성능에 따라 자동으로 선택되도록 설정될 수 있다.

<0088> 도 5는 도 1의 가상 머신 고장 예측 시스템의 고장 예측 모듈 내

BERT(Bidirectional Encoder Representations from Transformers) 모듈에 채용할 수 있는 구성을 설명하기 위한 블록도이다. 그리고 도 6은 도 5의 BERT 모듈에 채용할 수 있는 BERT 토큰화기 및 BERT 모델의 구조 및 고장 예측 기계 학습 알고리즘을 설명하기 위한 예시도이다.

<0089> 본 일실시예에서는, 고장 예측 모듈에 채용할 수 있는 BERT(Bidirectional Encoder Representations from Transformers) 모듈의 구조를 예시한다. 그리고, 구글(google)에서 학습시킨 후 공개한 두 BERT 모델 중 BERT-Base 모델의 구조를 이용하는 BERT 모듈에 대하여 설명한다.

<0090> BERT-Base 모델은 공개 모델로서, 본 실시예에서 제안하는 BERT-CNN 모델의 일부로서 중요한 역할을 하는 모델이다. 본 실시예에서 BERT 모듈(520)는 언어 모델을 포함하기 때문에 사전 훈련 이후에도 얼마든지 사용하고자 하는 태스크에 맞게 추가 학습이 가능하며, 이를 통해 사전 훈련용 말뭉치에 편향되지 않은 수치 벡터를 생성하도록 구성될 수 있다. 즉, 본 실시예에서는 사전 훈련 과정에서 역전파를 통해 미세튜닝을 수행함으로써 사전 훈련용 대량의 말뭉치와 로그에서 사용하는 단어의 종류나 의미가 다른 문제를 해결하여 로그 분석 기반의 가상 머신 고장 예측에 적합하도록 기계 학습 모델을 훈련시킬 수 있다.

<0091> 도 5 및 도 6을 참조하면, BERT 모듈(520)은 BERT 토큰화기(BERT tokenizer, 522) 및 BERT 모델(524)을 구비할 수 있다.

<0092> BERT 모듈(520)은 시퀀스-투-시퀀스(sequence-to-sequence, seq2seq) 모델 중 하나인 트랜스포머(Transformer)에서 인코더(Encoder)만을 사용하여 설계될 수

있다. 본 실시예의 가상 머신의 고장 예측을 위한 BERT 모델(524)은, 12층의 트랜스포머 인코더(Transformer Encoder)를 중첩시켜 구성될 수 있다.

<0093> 더 많은 트랜스포머 인코더를 사용하여 성능을 높인 모델을 사용할 수 있으나, 그 경우 학습에 더 오랜 시간이 걸리기 때문에, 본 실시예의 BERT 모델(524)은, 네트워크 관리자가 학습 시간과 모델의 성능 중 더 중요한 요소에 맞게 BERT 모델(524)의 트랜스포머 인코더의 개수를 선택하도록 구성될 수도 있다.

<0094> BERT 모듈(520)은 단어를 더 작은 서브워드(Sub-word)로 쪼개어 입력으로 사용하기 때문에 BERT 토큰화기(522)와 함께 사용한다. 본 실시예에서, BERT 토큰화기(522)는 각 단어의 어간과 어미를 구별하여 단어를 원형으로 변형시킨다. 즉, 도 6에서와 같이 'failed'를 'fail'과 '-ed'로 분리할 수 있다. 그리고 각 문장의 맨 처음에 문장의 시작을 알리는 특수 토큰 즉, [CLS] 토큰을 추가할 수 있다.

<0095> [CLS] 토큰은 문장 전체의 단어의 의미를 통합한 벡터 즉, 문장 전체가 하나의 벡터로 표현된 특수 토큰으로서, 그 출력 단어 벡터는 문장의 의미를 나타내는 문장 벡터로 사용될 수 있다. 따라서, 본 실시예의 BERT-CNN 모델(도 7 참조)을 사용할 때, 가상 머신 고장 예측 시스템은, 학습 시간을 줄이려고 하는 경우, 나머지 단어 벡터들을 합쳐서 사용하는 대신 [CLS] 토큰의 출력 벡터만 사용하도록 구성될 수 있다.

<0096> 또한, 본 실시예의 BERT 모듈(520)은 문장 내에서 각 서브워드(Sub-word) 토큰들의 위치를 나타내는 위치 임베딩(Position Embedding, E2)을 각 토큰 임베딩(Token Embedding, E1)과 합쳐서 중첩된 트랜스포머 인코더들의 입력으로 사용하

여 각 단어의 순서가 보전되도록 하면서 해당 문맥을 분석하도록 동작할 수 있다.

<0097> 이와 같이, 전처리 과정을 거친 입력 윈도우 크기 동안의 같은 서버 내의 가상 머신들의 로그들과 서버 로그들을 포함한 로그 파일(log file)은 BERT-CNN 모델의 입력으로 사용될 수 있다. 로그 파일 내 각 로그는 BERT 토큰화기(522)에 의해 서브워드 토큰들(sub-word tokens)로 분리되고 토큰 집합의 임베딩으로 변환되어 사전 학습된 BERT 모델(524)에 입력으로 사용될 수 있다.

<0098> 본 실시예의 BERT 모듈(520)은 최종적으로 각 서브워드 토큰 별로 일정한 크기의 벡터를 출력 임베딩(Output Embedding, E5)으로서 출력할 수 있다. BERT-Base 모델의 경우, 출력 임베딩(E5)은 각 서브워드 토큰 별로 768 차원의 벡터를 생성할 수 있다.

<0099> 도 7은 도 1의 가상 머신 고장 예측 시스템의 고장 예측 모듈 내 CNN 모델에 채용할 수 있는 구성과 BERT 모델과 결합관계를 설명하기 위한 블록도이다. 그리고 도 8은 도 7의 CNN 모델에 채용할 수 있는 구조 및 고장 예측 기계 학습 알고리즘을 설명하기 위한 예시도이다.

<0100> 본 일실시예에서는 가상 머신 고장 예측 시스템에 채용할 수 있는 CNN 모델의 구조와 이 구조에 적용되는 고장 예측 기계 학습 알고리즘을 설명한다.

<0101> 도 7을 참조하면, 고장 예측 모듈은 BERT 모델(524)와 CNN 모델(530)을 합친 구조를 구비할 수 있다. 고장 예측 모듈은 고장 예측 모델로 지칭될 수 있다. 고장 예측 모델은, 로그 데이터를 수치 행렬 즉, 로그 임베딩으로 변환하기 위해 BERT 모델(524)를 사용하고, 수치 행렬을 분석하여 고장 확률을 계산하기 위해 합성곱

신경망 모델(530)을 사용할 수 있다.

<0102> 일반적인 BERT 모델은 2018년에 구글(Google)이 공개한 언어 모델(language model)로서, 워드 임베딩(word embedding)과 마찬가지로 단어를 입력으로 받아서 단어의 의미를 나타내는 수치 벡터(numeric vector)를 생성하기 위해 사용되지만, 본 실시예의 BERT 모델(524)은, 워드 임베딩 기법과 다르게, 같은 단어에 대해서도 문맥에 따라 다른 벡터를 생성하는 방법론을 적용한다. 즉, BERT 모델(524)은 고장 여부가 태깅되어 있는 로그 데이터를 기반으로 사전 훈련된 후에, 입력 로그 데이터 내 문장의 맥락을 파악하여 같은 단어에 대해서도 서로 다른 임베딩을 출력하도록 구성된다.

<0103> 본 실시예의 BERT 모델(524)은 레이블이 없는 대량의 말뭉치로부터 사전 학습된 언어모델일 수 있으며, 본 실시예에서 해결하고자 하는 태스크에 맞게 추가 학습 즉, 미세 튜닝(fine-tuning)을 적용함으로서 해당 태스크에 적합한 수치 벡터를 생성해 낼 수 있고, 이에 의해, 기존의 워드 임베딩 기반 모델들보다 높은 성능을 나타낸다.

<0104> BERT 모델(524)는 각 서브워드 토큰에 대해 수치 벡터를 생성한다. BERT-CNN 모델의 사전 학습 과정에서, CNN 모델(530)과 BERT 모델(524)에 역전파(backpropagation)가 발생하고, 이러한 역전파를 통해 BERT 모델(524)에서 미세 튜닝(fine-tuning)을 통한 추가 학습이 수행되고, 그에 의해 BERT 모델(524)은 각 토큰에 대해 고장 예측에 적합한 단어 벡터를 생성해 내도록 학습될 수 있다.

<0105> BERT 모델(524)에서 생성된 수치 벡터는 CNN 모델(530)의 입력으로 사용되기

위해 합쳐질 수 있다(concatenate). 이 때 합치는 과정은 벡터를 더하는(add) 개념이 아니라, 각 벡터를 중첩시킴으로서 행렬을 만드는 것을 나타낸다. 이것은 BERT 모델(524)이 각 서브워드 토큰에 대해 같은 차원의 벡터를 출력시키기 때문에 이렇게 행렬로 변환시키는 것이 가능하다. 이러한 행렬 변환에 의해 생성되는 행렬은 BERT 모델(524)의 출력 층의 차원에 입력 로그에 포함된 서브워드 토큰의 개수들의 곱에 의해 결정될 수 있다.

<0106> CNN 모델(530)은 커널들(kernels)이 행렬 안을 이동하면서 합성곱 연산(convolution operation)을 수행하고 대량의 데이터로부터 핵심 데이터를 추출하는 기계 학습 알고리즘이다. 커널은 행렬 안에서 아래로 이동하도록 하기 위해 행렬과 열 수는 같고, 행 수는 다르게 설정될 수 있다.

<0107> 예를 들어, CNN 모델(530)의 입력 행렬은 BERT 모델(524)의 출력 층의 차원(D1)과 입력 로그에 포함된 서브워드 토큰 개수(N1)로 이루어진 2차원 행렬이 된다. 즉, 각 문장으로부터의 벡터들에 대응하는 커널은 BERT 모델(524)의 출력 층 또는 은닉 층의 차원(D1)과 미리 설정된 커널 크기의 2차원 행렬이 될 수 있다. BERT 모델(524)이 BERT-Base 모델인 경우, 출력 층 또는 은닉 층의 차원(D1)은 768일 수 있다. 이때 커널 크기는 입력 로그에 포함된 서브워드 토큰 개수(N1)와 다른 커널 행렬의 행 수로서 네트워크 관리자가 자유로이 설정할 수 있다. 이러한 CNN 모델(530)을 사용하는 경우, 예를 들어 3, 4, 5의 커널 크기를 가지는 커널들을 각각 100개씩 사용하는 식으로 합성곱 연산을 구성할 수 있다.

<0108> 한편, 입력 로그의 양이 방대한 경우, 데이터를 더 압축시키기 위해서 더 큰

커널이 사용될 수 있고, 입력 로그의 양이 적다면 커널의 종류를 줄여 적은 종류의 커널이 사용될 수 있다. 커널의 크기에 대해서는 정답이 존재하지 않기 때문에, 고장 예측 모듈은 사전 학습 단계에서 사용하고자 하는 네트워크의 로그를 기반으로 적합한 커널 크기나 커널 개수를 사용하도록 설정될 수 있다.

<0109> 사전 학습 단계에서, 역전파를 통해서 CNN 모델(530)의 커널이 고장과 관련된 데이터를 추출해낼 수 있도록 학습될 수 있다. 즉, CNN 모델(530)에는, 커널들이 합성곱 연산을 수행하는 컨볼루션 레이어(Convolutional Layer, 532)가 구비될 수 있다. 그리고, 컨볼루션 레이어(532)의 다음에는 커널의 출력들 별로 가장 높은 값을 남기는 맥스-풀링 레이어(Max-pooling Layer, 534)가 배치될 수 있다. 맥스-풀링 레이어(534)를 통해 가장 높은 데이터를 남김으로서, CNN 모델(530)에서 고장과 관련된 데이터로서 가장 중요한 값만 남길 수 있다.

<0110> 맥스-풀링 레이어를 통해 최종적으로 커널의 개수만큼의 차원을 가지는 벡터가 생성되면, 이 벡터를 완전 연결 계층(Fully-connected Layer, FC Layer, 536)의 입력으로 사용하여 0에서 1 사이의 고장 예측 확률을 계산할 수 있다. 완전 연결 계층(536)은 모든 뉴런이 다음 층의 모든 뉴런과 연결된 층을 의미하며, 입력은 고차원의 벡터이지만, 출력은 단일 값이기 때문에 맥스-풀링 레이어의 출력 값에 가중치(weight)를 곱해서 더하는 단순한 형태로 구현될 수 있다.

<0111> 이러한 완전 연결 계층(536)은 기계 학습 모델의 마지막에 연결되어 최종적으로 해결하고자 하는 태스크에 맞게 출력(550)을 분류하도록 사용될 수 있다. 출력은 학습 결과 또는 예측 결과(prediction result)로 지칭될 수 있다. CNN 모

델(536)에서 커널들과 완전 연결 계층(536)의 가중치들은 사전 훈련 단계에서 역전파를 통해서 학습되는 학습 대상이 될 수 있다.

<0112> 또한, 사전 훈련 단계에서의 학습을 위해 도 1 내지 도 3에 예시한 데이터 수집 환경에서 로그 데이터 뿐만 아니라 고장 이력 데이터를 수집할 수 있다. 이 경우, 고장 예측 모델은 일정 시간 뒤에 고장이 발생할 확률을 계산하는 것을 목적으로 하기 때문에, 예컨대 학습 단계에서 입력 로그로부터 일정 시간 뒤에 서버에 포함된 가상 머신 중 하나라도 고장으로 태깅되면 1을 출력 값으로 가지고, 모든 가상 머신이 정상 상태였으면 0을 출력 값으로 갖도록 구성될 수 있다.

<0113> 전술한 사전 학습 단계에서 만일 최종 출력 값이 0에 가깝지만 실제로는 고장이었으면 역전파를 통해 BERT 모델(524)과 CNN 모델(530)의 각 요소가 수정될 것이고, 마찬가지로 출력 값이 1과 가깝지만 실제로는 정상이었으면 역전파를 통해 각 요소가 수정될 수 있다. 이와 같이, BERT-CNN 모델은 입력 로그를 통해 고장과 정상을 효과적으로 분류하도록 학습될 수 있다. 이러한 사전 훈련이 완료된 BERT-CNN 모델은 같은 네트워크에서 로그 기반 고장 예측 모델에 사용될 수 있다.

<0114> 한편, 장애 관련 로그가 장애 가상 머신이 아닌 서버 로그나 다른 가상 머신에서 발생하는 경우가 있다. 이러한 경우, 장애라고 표시된 입력창에는 장애와 관련된 로그 시퀀스가 포함되어 있지 않고, 장애와 관련된 로그가 포함된 입력 윈도우에는 정상으로 표시되어 기계 학습 모델이 제대로 학습되지 않을 수 있다. 이에 본 실시예에서는 서버의 로그와 해당 서버의 모든 가상 머신의 로그들을 함께 사용하고, 서버에서 작동하는 가상 머신들 중 하나라도 실패하면 입력 윈도우의 해당

로그를 고장으로 표시하도록 구성한다.

<0115> 또한 본 실시예에서 고장 예측 모듈은, 기계 학습 모델을 훈련시키기 위한 훈련 데이터를 생성할 때, 고장으로 표시된 특정 가상 머신의 윈도우에 대해서만 입력 윈도우 크기를 변경하도록 구성될 수 있다.

<0116> 즉, 사전 훈련 과정에서, 고장 예측 모듈은, 기본적으로 입력 윈도우의 윈도우 크기는 예를 들어 10분으로 설정하고, 고장 관련 로그가 발생한 입력 윈도우의 윈도우 크기만 28분의 입력 윈도우 크기로 변경할 수 있다. 이 경우, 고장 발생 시각 30분 전부터 2분 전까지의 모든 로그를 입력으로 사용할 수 있다. 즉, 고장 예측 후 조치를 취하기 위해서는 고장 예측 시점과의 최소 시간 간격이 필요할 수 있고, 본 실시예의 경우, 그 간격을 2분으로 최소화할 수 있다. 여기서, 윈도우 크기는 그 최대값이 30분인 것을 가정한 것으로, 윈도우 크기의 최대값이 변동하는 경우, 그에 따라 고장 관련 로그가 발생한 입력 윈도우의 윈도우 크기를 적절하게 가변시킬 수 있다.

<0117> 한편, 입력 윈도우의 윈도우 크기가 다르고, 윈도우 크기가 다름에 따라 기계 학습 모델의 입력 로그의 양이 달라지는 문제가 발생할 수 있다. 그러나 CNN 모델(530)은 많은 양의 데이터에서 중요한 데이터만 추출하는 방법으로 학습한다. 즉, 본 실시예에서는 CNN 모델(530)의 풀링 레이어로 맥스 풀링 레이어를 사용하기 때문에 CNN 모델(530)이 정상적으로 학습되면 입력 데이터의 양에 관계없이 가상 머신의 장애 또는 고장과 관련된 메시지만 찾을 수 있다. 따라서, CNN 모델(530)에서 입력 데이터의 크기 변화는 성능에 영향을 미치지 않는다. 이와 같이, 전술한

고장 관련 로그가 발생한 입력 윈도우의 윈도우 크기를 변경하는 라벨링 방식은 CNN 모델(530)을 이용하는 기계 학습 모델의 성능을 향상시킬 수 있다.

<0118> 도 9는 본 발명의 다른 실시예에 따른 가상 머신 고장 예측 방법을 설명하기 위한 흐름도이다.

<0119> 도 9를 참조하면, 가상 머신 고장 예측 방법은 고장 여부가 태깅이 되어 있는 로그 데이터를 기반으로 사전 훈련하는 단계(S910)를 포함할 수 있다. 사전 훈련 단계(S910)에서는 CNN 모델과 BERT 모델의 역전파를 이용하여 BERT 모델과 CNN 모델을 미세 튜닝(fine-tuning)하는 과정을 포함할 수 있다. 미세 튜닝에 의하면, 로그 기반의 가상 머신 고장 예측에 좀더 적합하도록 사전 훈련된 언어 모델의 전체 하이퍼파라미터가 수정될 수 있다.

<0120> 또한, 가상 머신 고장 예측 방법은, 사전 훈련 단계(S910) 전에 학습 데이터를 생성하는 단계를 더 포함할 수 있다. 학습 데이터 생성 단계에서는 상기 가상머신들 각각에 일정 크기의 패킷 또는 테스트 신호를 보내고 상기 패킷 또는 테스트 신호에 대한 응답 메시지를 받고 상기 응답 메시지를 분석하여 상기 가상머신들 각각의 네트워크 연결 상태를 확인하는 과정을 포함할 수 있다.

<0121> 다음, 각 가상 머신 및 가상 머신을 동작시키는 서버로부터 발생하는 로그를 수집하는 단계(S920)를 포함할 수 있다. 로그 수집 단계(S920)는 컴퓨팅 노드의 각 서버에서 각 가상 머신의 입력 윈도우에 기록되는 로그들을 모니터링 노드의 통합 입력 윈도우로 합치고, 각 가상 머신의 고장 이력들을 모니터링 노드의 통합 고장 이력으로 합치는 과정을 포함할 수 있다. 통합 입력 윈도우와 통합 고장 이력을 포

함한 입력 로그 데이터는 모니터링 노드로부터 고장 예측 모듈로 전달될 수 있다.

<0122> 다음, 수집한 로그 즉, 입력 로그 데이터를 BERT 모듈의 입력으로 사용하여 로그의 특징을 나타내는 수치 행렬인 로그 임베딩 행렬로 변환하는 단계(S930)를 포함할 수 있다. 행렬 변환 단계(S930)에서는 입력 로그 데이터를 서브워드 토큰으로 변환하는 BETR 토큰화기와 서브워드 토큰을 임베딩을 변환하는 BERT 모델이 이용될 수 있다.

<0123> 다음, 변환된 로그 임베딩 행렬을 기초로 사전 학습된 CNN 모델을 이용하여 서버별로 가상 머신에 고장이 발생할 확률을 예측하는 단계(S940)를 포함할 수 있다. CNN 모델에서 커널들과 완전 연결 계층의 가중치들은 사전 훈련 단계에서 역전파를 통해서 학습될 수 있다. 이러한 CNN 모델에서는 입력 로그 데이터 내 각 문장으로부터의 벡터들에 대해 합성곱 연산을 수행하고, 맥스-풀링 레이어를 통해 커널의 합성곱 연산 결과들 중 가장 큰 값을 가진 데이터만을 남기도록 동작하고, 완결연결 계층을 통해 커널 개수만큼의 차원을 가진 가장 큰 값의 벡터 데이터를 단일 값의 출력으로 고장 예측 확률을 계산할 수 있다.

<0124> 다음, 예측된 서버별 가상 머신 고장 발생 확률에 따라 고장 처리 방법을 결정하는 단계(S950)를 포함할 수 있다. 고장 처리 방법은 미리 설정된 일정 시간 내에 고장 발생이 예측된 서버의 가상머신들을 일정 시간 내에 고장이 예측되지 않은 서버로 이전하는 것을 포함할 수 있다.

<0125> 도 10은 본 발명의 또 다른 실시예에 따른 가상 머신 고장 예측 장치에 채용할 수 있는 네트워크 기능 가상화 인프라(Network Function Virtualization

Infrastructure, NFVI) 환경을 설명하기 위한 예시도이다. 본 실시예의 NFVI 환경은 BERT-CNN 모델 기반으로 고장 예측 모듈에 적용될 수 있다.

<0126> 도 10을 참조하면, BERT-CNN 모델은 고장 예측 모듈(Failure Prediction Module)의 형태로서 네트워크에 별개의 서버 또는 노드에 설치될 수 있다. NFVI 환경에서는 데이터 수집 단계에서와 마찬가지로 모니터링 노드(300)가 컴퓨팅 노드(200)의 서버에 설치된 각 가상 머신로부터의 로그와 해당 서버로부터의 로그를 수집할 수 있다.

<0127> 모니터링 노드(300)에서는 데이터 수집 단계에서와 마찬가지로 입력 윈도우 시간 동안의 각 가상 머신 및 서버의 로그들을 모두 합쳐서 입력 윈도우를 생성할 수 있다. 생성된 입력 윈도우 동안의 로그 데이터는 고장 예측 모듈(Faulure Prediction Module, 500)의 입력으로서 전달될 수 있다.

<0128> 고장 예측 모듈(500)은 사전 훈련 단계에서와 마찬가지로 BERT 토큰화기가 로그 데이터의 각 문장을 서브워드 토큰으로 변환시키고 BERT 모델의 입력으로 사용하여 수치 행렬로 변환시킨 후, CNN 모델의 입력으로 사용하여 최종적으로 일정 시간 뒤의 고장 발생 확률을 계산하도록 구성될 수 있다. 이 때, 입력 윈도우의 크기(input window size)와 고장 예측 시각과의 시간 차(gap)는 사전 훈련 단계에서 네트워크 관리자가 긴급성과 예측 정확도 중 더 중요한 것을 기반으로 적절한 값을 설정할 수 있다.

<0129> 계산된 값이 일정 임계 값(threshold)을 넘는 경우, 고장이 발생할 확률이 높은 것으로 판단되기 때문에 컨트롤러 노드(controller node, 100)로 예측 결과가

전달이 된다. 컨트롤러 노드(100)는 전달된 예측 결과에 따라 사전에 네트워크 관리자가 설정해놓은 고장 완화 작업을 수행하도록 컴퓨팅 노드(200)에 명령을 전달할 수 있다.

<0130> 본 실시예의 가상 머신 고장 예측 장치(1000)는, 모니터링 노드(300)와 고장 예측 모듈(500)을 포함한 형태로 구성될 수 있고, 서버 별로 가상 머신 고장이 발생할 확률을 계산하도록 구현될 수 있다. 서버 별로 가상 머신의 고장 발생 확률을 계산하기 때문에, 가상 머신 고장 예측 장치(1000)로부터 예측 결과를 전달받은 컨트롤러 노드(100)는 고장 발생 확률이 높을 것으로 예측되는 서버로부터 고장 발생 확률이 낮을 것으로 예측되는 서버로 가상 머신을 이전(migration)시킬 수 있다.

<0131> 본 실시예에서 여러 파라미터들에 대해서 네트워크 관리자가 직접 설정할 수 있다고 표현하였으나, 이에 한정되지는 않는다. 본 실시예에서 정확하게 지정하지 않은 것으로서 BERT-Base 모델을 사용할 것인가, BERT-Large 모델을 사용할 것인가, 커널의 크기 및 개수를 몇 개를 사용할 것인가, 입력 윈도우의 크기를 얼마로 사용할 것인가, 고장 예측 시점과의 시간 차는 얼마로 할 것인가 등의 파라미터들은, 훈련에 얼마나 오랜 시간이 걸리느냐, 예측 결과를 계산하는 데 얼마나 오랜 시간이 걸리느냐, 예측 정확도가 얼마가 되느냐를 결정짓는 요소이기 때문에, 네트워크 관리자가 고장 예측 모듈이 설치될 서버의 성능에 따라, 고장 예측 정확도의 중요도 등에 따라 적절한 값을 설정하도록 구성될 수 있다. 각 파라미터 값이 커질수록 고장 예측 정확도는 높아지나, 훈련 및 고장 확률 값을 계산해내는 데 더 많은 컴퓨팅 자원을 사용하여야 하며, 학습에 더 오랜 시간이 걸릴 수 있으므로, 적

절한 트레이드 오프가 적용될 수 있다.

<0132> 도 11은 본 발명의 또 다른 실시예에 따른 가상 머신 고장 예측 장치에 채용할 수 있는 구성을 설명하기 위한 개략적인 블록도이다.

<0133> 도 11을 참조하면, 가상 서버 고장 예측 장치(1000)은, 고장 예측 모듈을 포함하거나, 고장 예측 모듈과 모니터링 노드를 포함하도록 구성될 수 있다. 이러한 가상 서버 고장 예측 장치(1000)는 적어도 하나의 프로세서(1100) 및 메모리(1200)를 포함할 수 있다.

<0134> 또한, 가상 서버 고장 예측 장치(1000)은 네트워크와 연결되어 통신을 수행하는 송수신 장치(1300)를 더 포함할 수 있다. 또한, 가상 서버 고장 예측 장치(1000)는 입력 인터페이스 장치(1400), 출력 인터페이스 장치(1500), 저장 장치(1600) 등을 더 포함하도록 구성될 수 있다. 가상 서버 고장 예측 장치(1000)에 포함된 각각의 구성 요소들은 버스(bus, 1700)에 의해 연결되어 서로 통신을 수행하도록 구성될 수 있다.

<0135> 프로세서(1100)는 메모리(1200) 및 저장 장치(1600) 중에서 적어도 하나에 저장된 프로그램 명령(program command)을 실행할 수 있다. 프로세서(1100)는 중앙 처리 장치(central processing unit, CPU), 그래픽 처리 장치(graphics processing unit, GPU), 또는 본 발명의 실시예들에 따른 방법들이 수행되는 전용의 프로세서를 의미할 수 있다.

<0136> 메모리(1200) 및 저장 장치(1600) 각각은 휘발성 저장 매체 및 비휘발성 저장 매체 중에서 적어도 하나로 구성될 수 있다. 예를 들어, 메모리(1200)는 읽기

전용 메모리(read only memory, ROM) 및 랜덤 액세스 메모리(random access memory, RAM) 중에서 적어도 하나로 구성될 수 있다.

<0137> 송수신 장치(1300)는 근거리 무선 네트워크나 케이블 연결, 위성과의 통신, 범용 기지국과의 유선 또는 무선 통신, 모바일 에지 코아 네트워크나 코아 네트워크(core network)와의 아이디얼 백홀 링크(ideal backhaul link) 또는 넌(non)-아이디얼 백홀 링크의 연결 등을 위한 통신인터페이스나 서브통신시스템을 포함할 수 있다.

<0138> 입력 인터페이스 장치(1400)는 키보드, 마이크, 터치패드, 터치스크린 등의 입력 수단들에서 선택되는 적어도 하나와 적어도 하나의 입력 수단을 통해 입력되는 신호를 기저장된 명령과 매핑하거나 처리하는 입력 신호 처리부를 포함할 수 있다.

<0139> 출력 인터페이스 장치(1500)는 프로세서(1100)의 제어에 따라 출력되는 신호를 기저장된 신호 형태나 레벨로 매핑하거나 처리하는 출력 신호 처리부와, 출력 신호 처리부의 신호에 따라 진동, 빛 등의 형태로 신호나 정보를 출력하는 적어도 하나의 출력 수단을 포함할 수 있다. 적어도 하나의 출력 수단은 스피커, 디스플레이 장치, 프린터, 광 출력 장치, 진동 출력 장치 등의 출력 수단들에서 선택되는 적어도 하나를 포함할 수 있다.

<0140> 전술한 본 실시예에 따른 방법들은 다양한 컴퓨터 수단을 통해 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또

는 조합하여 포함할 수 있다. 컴퓨터 판독 가능 매체에 기록되는 프로그램 명령은 본 발명을 위해 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다.

<0141> 컴퓨터 판독 가능 매체의 예에는 롬(rom), 램(ram), 플래시 메모리(flash memory) 등과 같이 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러(compiler)에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터(interpreter) 등을 사용해서 컴퓨터에 의해 실행될 수 있는 고급 언어 코드를 포함한다. 상술한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 적어도 하나의 소프트웨어 모듈로 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

<0142> 이상 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

【청구범위】

【청구항 1】

머신 러닝 기반 가상 머신 고장 예측 방법에 있어서,

네트워크 기능 가상화 환경에서 동작하는 서버에 탑재되고 가상 네트워크 기능 또는 웹 서버를 운영하는 가상머신들 각각에서 발생하는 로그들을 상기 서버의 로그에 합한 로그 데이터를 획득하는 단계;

상기 로그 데이터를 사전 훈련된 언어 모델의 입력으로 사용하여 로그의 특징을 나타내는 수치 행렬인 로그 임베딩 행렬로 변환하는 단계; 및

상기 로그 임베딩 행렬을 입력으로 하는 합성곱 신경망 모델을 이용하여 서버별로 가상머신들에 일정 시간 이내에 고장이 발생할 확률을 예측하는 단계;

를 포함하고,

상기 로그 데이터는 상기 가상머신들의 로그들과 상기 가상머신들의 고장 이력들에 기초한 고장 관련 로그를 포함하는, 가상 머신 고장 예측 방법.

【청구항 2】

청구항 1에 있어서,

상기 언어 모델은, 고장 여부가 태깅되어 있는 로그 데이터를 기반으로 사전 훈련된 후에, 상기 로그 데이터 내 문장의 맥락을 파악하여 같은 단어에 대하여 서로 다른 임베딩을 출력하는 모델인, 가상 머신 고장 예측 방법.

【청구항 3】

청구항 2에 있어서,

상기 언어 모델은, 상기 단어를 더 작은 하위 단어로 나누고 이를 토큰으로 사용하여 각 토큰에 대한 출력 임베딩을 생성하며, 상기 출력 임베딩이 생성을 위해 문장의 시작을 나타내는 특수 토큰과 위치 임베딩을 이용하는, 가상 머신 고장 예측 방법.

【청구항 4】

청구항 3에 있어서,

상기 언어 모델과 상기 합성곱 신경망 모델은, 사전 훈련 과정에서 역전파로 미세 튜닝되며, 상기 미세 튜닝에 의해 상기 언어 모델의 전체 하이퍼파라미터가 수정되는, 가상 머신 고장 예측 방법.

【청구항 5】

청구항 1에 있어서,

상기 로그 데이터가 사전 훈련된 언어 모델에 입력되기 전에 상기 로그 데이터를 전처리하는 단계를 더 포함하는, 가상 머신 고장 예측 방법.

【청구항 6】

청구항 5에 있어서,

상기 가상 네트워크 기능 또는 웹 서버를 운영하는 가상머신의 로그가 커널 로그일 때, 상기 전처리는 기설정된 규칙 기반으로 상기 커널 로그의 메이저 레지스터의 주소를 제거하는 것을 포함하는, 가상 머신 고장 예측 방법.

【청구항 7】

청구항 5에 있어서,

상기 전처리는, 상기 로그들 중 미리 설정된 단어들 중 적어도 어느 하나 이상의 단어를 포함하는 로그들만을 남기는 것을 포함하는, 가상 머신 고장 예측 방법.

【청구항 8】

청구항 7에 있어서,

상기 단어들은, 재설정(reset), 실패(fail), 아님(not), 고장(failure), 시간초과(timeout), 종료(kill), 오류(err), 메모리부족(out of memory, oom), 중지(stop), 종료(exit), 재시작(restart), 장시간(long) 및 경고(warn) 중 적어도 일부를 포함하는, 가상 머신 고장 예측 방법.

【청구항 9】

청구항 5에 있어서,

상기 전처리는, 상기 로그들에서 시간 정보 및 애플리케이션 정보를 추출하는 과정, 상기 로그들에서 숫자와 기호를 제거하는 과정, 상기 로그들에서 대문자를 소문자로 변경하는 과정, 또는 이들의 조합 과정을 포함하는, 가상 머신 고장 예측 방법.

【청구항 10】

청구항 9에 있어서,

상기 로그들에서 추출한 시간 정보에 기초하여 상기 서버의 로그와 상기 가상머신들의 로그들이 합해지는 입력 윈도우의 모든 로그들을 시간순으로 정렬하는 단계를 더 포함하는, 가상 머신 고장 예측 방법.

【청구항 11】

청구항 1에 있어서,

상기 가상머신들 각각에서 발생하는 로그들과 상기 서버에서 발생하는 로그를 단일 입력 윈도우에서 합치는 단계; 및

상기 가상머신들 각각에서 발생하는 로그들과 상기 가상머신들 각각에 대한 네트워크 연결 상태의 확인 결과에 의해 생성되는 고장 이력들을 단일 고장 이력으로 합치는 단계;를 더 포함하는, 가상 머신 고장 예측 방법.

【청구항 12】

청구항 1에 있어서,

상기 예측하는 단계에서 예측된 상기 일정 시간 내에 고장 발생 확률을 가진 가상머신에 대하여 고장 처리 방법을 결정하는 단계를 더 포함하며,

상기 고장 처리 방법은 상기 일정 시간 내에 고장 발생이 예측된 서버의 가상머신들을 상기 일정 시간 내에 고장이 예측되지 않은 서버로 이전하는 것을 포함하는, 가상 머신 고장 예측 방법.

【청구항 13】

청구항 1에 있어서,

상기 가상머신들 중 적어도 일부는 서버의 종류와 트래픽 식별에 따라 동적으로 다른 서버와의 엔드투엔드 서비스를 제공하기 위해 서비스 기능 체이닝으로 구성된 복수의 가상 네트워크 기능들을 통과하도록 구성되고 상기 서비스 기능 체이닝을 위한 메시지 절차를 수행하도록 구성되는, 가상 머신 고장 예측 방법.

【청구항 14】

청구항 1에 있어서,

상기 합성곱 신경망 모델을 학습시키기 위한 학습 데이터를 생성하는 단계를 더 포함하며,

상기 학습 데이터를 생성하는 단계는, 상기 가상머신들 각각에 일정 크기의 패킷 또는 테스트 신호를 보내고 상기 패킷 또는 테스트 신호에 대한 응답 메시지를 받고 상기 응답 메시지를 분석하여 상기 가상머신들 각각의 네트워크 연결 상태를 확인하는 것을 포함하는, 가상 머신 고장 예측 방법.

【청구항 15】

머신 러닝 기반 가상 머신 고장 예측 방법을 수행하는 가상 머신 고장 예측 장치로서,

머신 러닝 기반으로 가상 머신의 고장을 예측하기 위해 메모리에 저장된 프로그램 명령들을 수행하는 프로세서를 포함하고, 상기 프로세서는,

네트워크 기능 가상화 환경에서 동작하는 서버에 탑재되고 가상 네트워크 기능 또는 웹 서버를 운영하는 가상머신들 각각에서 발생하는 로그들을 상기 서버의 로그에 합한 로그 데이터를 입력으로 사용하여 로그의 특징을 나타내는 수치 행렬인 로그 임베딩 행렬로 변환하는 사전 훈련된 언어 모델; 및

상기 로그 임베딩 행렬을 입력으로 하여 서버별로 가상머신들에 일정 시간 이내에 고장이 발생할 확률을 예측하는 합성곱 신경망 모델;을 구비하며,

상기 로그 데이터는 상기 가상머신들의 로그들과 상기 가상머신들의 고장 이

력들에 기초한 고장 관련 로그를 포함하는, 가상 머신 고장 예측 장치.

【청구항 16】

청구항 15에 있어서,

상기 사전 훈련된 언어 모델은, 고장 여부가 태깅되어 있는 로그 데이터를 기반으로 사전 훈련된 후에, 상기 로그 데이터 내 문장의 맥락을 파악하여 같은 단어에 대하여 서로 다른 임베딩을 출력하는 모델인, 가상 머신 고장 예측 장치.

【청구항 17】

청구항 15에 있어서,

상기 사전 훈련된 언어 모델과 상기 합성곱 신경망 모델은, 사전 훈련 과정에서 역전파로 미세 튜닝되며, 상기 미세 튜닝에 의해 상기 사전 훈련된 언어 모델의 전체 하이퍼파라미터가 수정되는, 가상 머신 고장 예측 장치.

【청구항 18】

청구항 15에 있어서,

상기 로그 데이터가 상기 사전 훈련된 언어 모델에 입력되기 전에 상기 로그 데이터를 전처리하는 전처리 유닛을 더 포함하는, 가상 머신 고장 예측 장치.

【청구항 19】

청구항 15에 있어서,

상기 로그 데이터를 생성하는 모니터링 노드를 더 포함하고,

상기 모니터링 노드는, 상기 가상머신들 각각에서 발생하는 로그들과 상기 서버에서 발생하는 로그를 단일 입력 윈도우로 합치고, 상기 가상머신들 각각에서

발생하는 로그들과 상기 가상머신들 각각에 대한 네트워크 연결 상태의 확인 결과에 의해 생성되는 상기 가상머신들 각각의 고장 이력들을 단일 고장 이력으로 합치는, 가상 머신 고장 예측 장치.

【청구항 20】

청구항 19에 있어서,

상기 모니터링 노드는, 상기 로그들에서 추출한 시간 정보에 기초하여 상기 서버의 로그와 상기 가상머신들의 로그들이 합해지는 입력 윈도우의 모든 로그들을 시간순으로 정렬하는, 가상 머신 고장 예측 장치.

【요약서】

【요약】

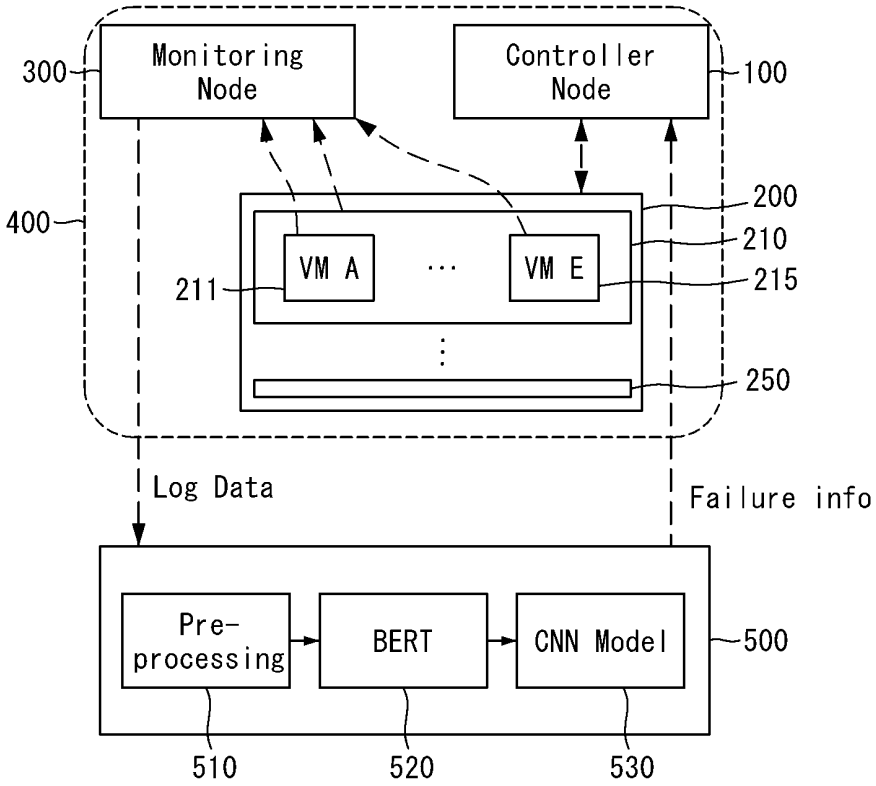
소프트웨어 정의 네트워크 및 네트워크 기능 가상화 환경에서 서버 별로 로그 분석을 통하여 가상 머신의 고장 발생 확률을 예측하는 방법 및 장치가 개시된다. 이 방법은, 가상 네트워크 기능 또는 웹 서버를 운영하는 가상머신들 각각에서 발생하는 로그들을 가상머신들이 탑재된 서버의 로그에 합한 로그 데이터를 획득하고, 로그 데이터를 사전 훈련된 언어 모델의 입력으로 사용하여 로그의 특징을 나타내는 수치 행렬인 로그 임베딩 행렬로 변환하고, 로그 임베딩 행렬을 입력으로 하는 합성곱 신경망 모델을 이용하여 서버별로 가상머신들에 일정 시간 이내에 고장이 발생할 확률을 예측하는 단계를 포함한다.

【대표도】

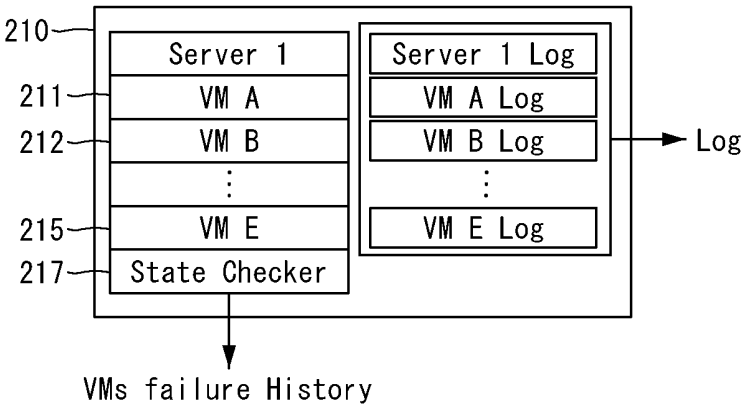
도 1

【도면】

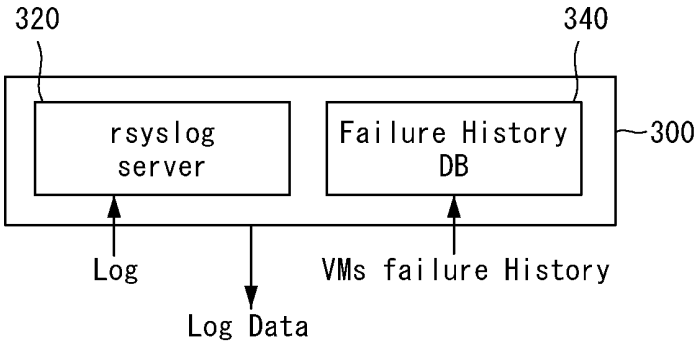
【도 1】


300
Monitoring Node
Controller Node
100
400
200
210
211
VM A
VM E
215
250
Log Data
Failure info
Pre-processing
BERT
CNN Model
500
510
520
530

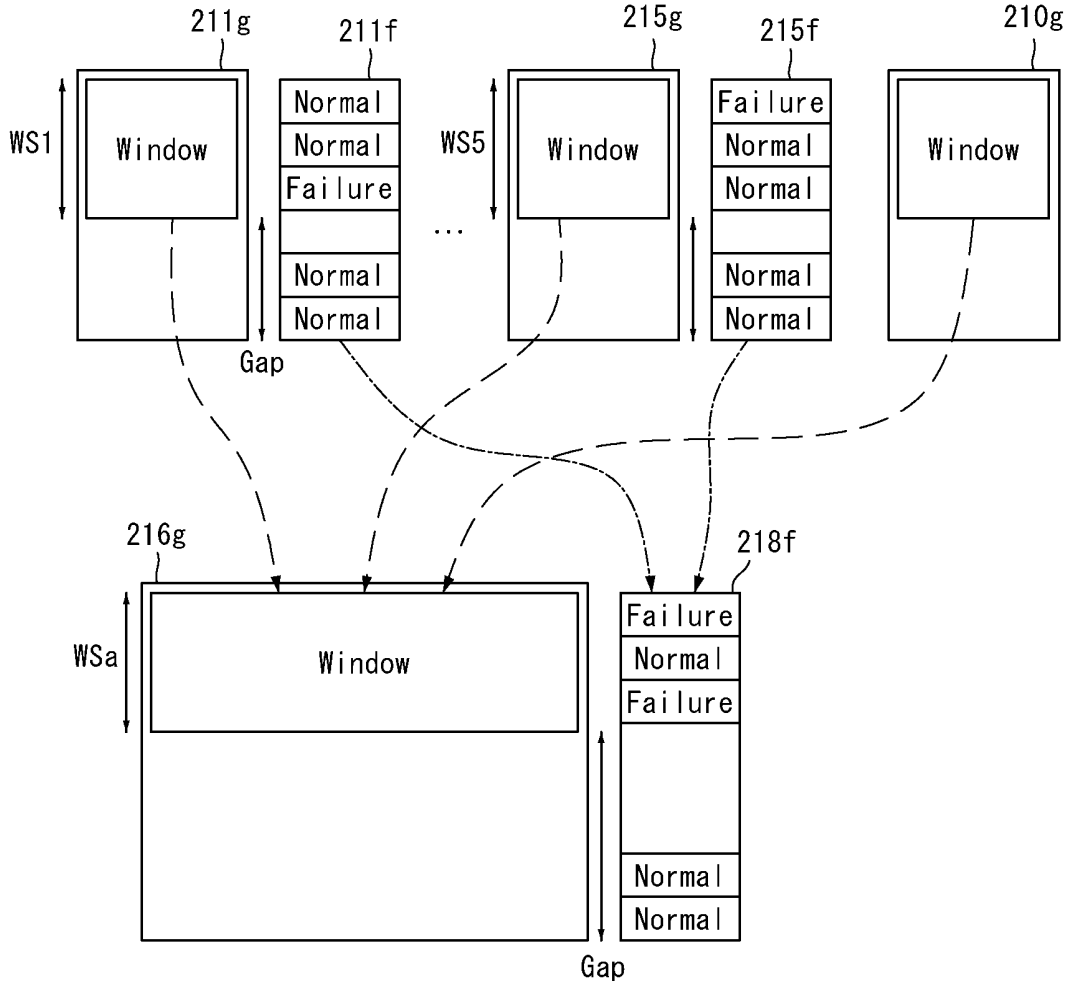
【도 2】


210
Server 1
Server 1 Log
211
VM A
VM A Log
212
VM B
VM B Log
Log
215
VM E
VM E Log
217
State Checker
VMs failure History

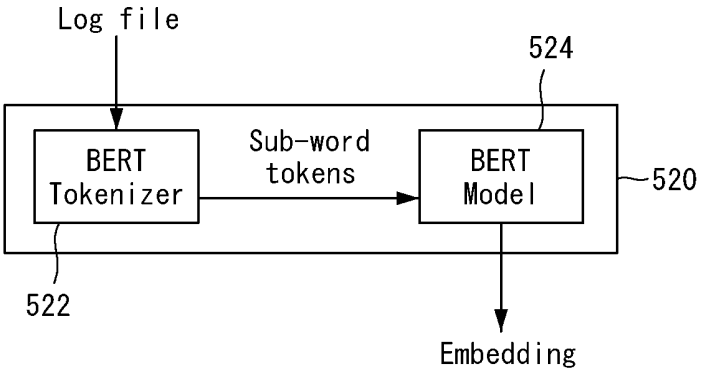
【도 3】


320
340
rsyslog server
Failure History DB
300
Log
VMs failure History
Log Data

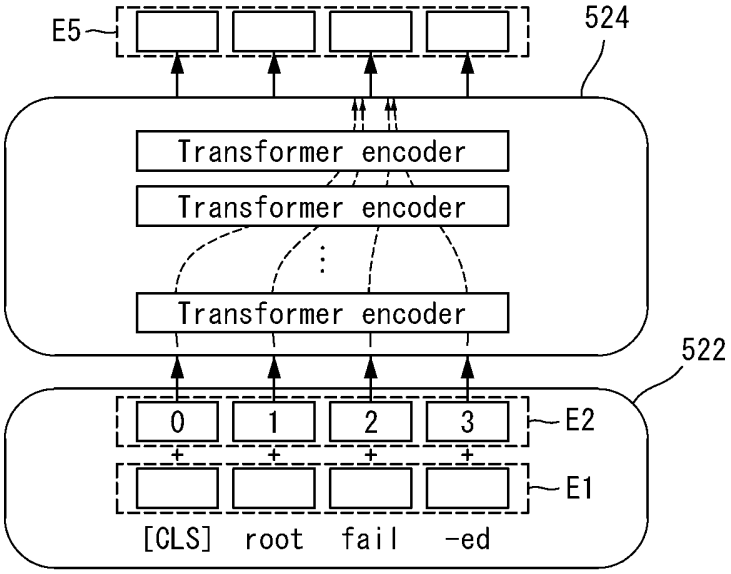
【도 4】


211g
211f
215g
215f
210g
WS1
Window
Normal
Normal
Failure
Normal
Normal
Gap
WS5
...
Window
Failure
Normal
Normal
Normal
Normal
Window
216g
218f
WSa
Window
Failure
Normal
Failure
Normal
Normal
Gap

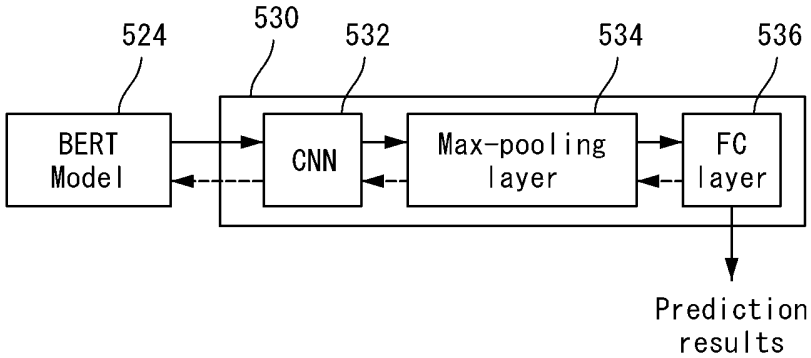
【도 5】


Log file
524
BERT Tokenizer
Sub-word tokens
BERT Model
520
522
Embedding

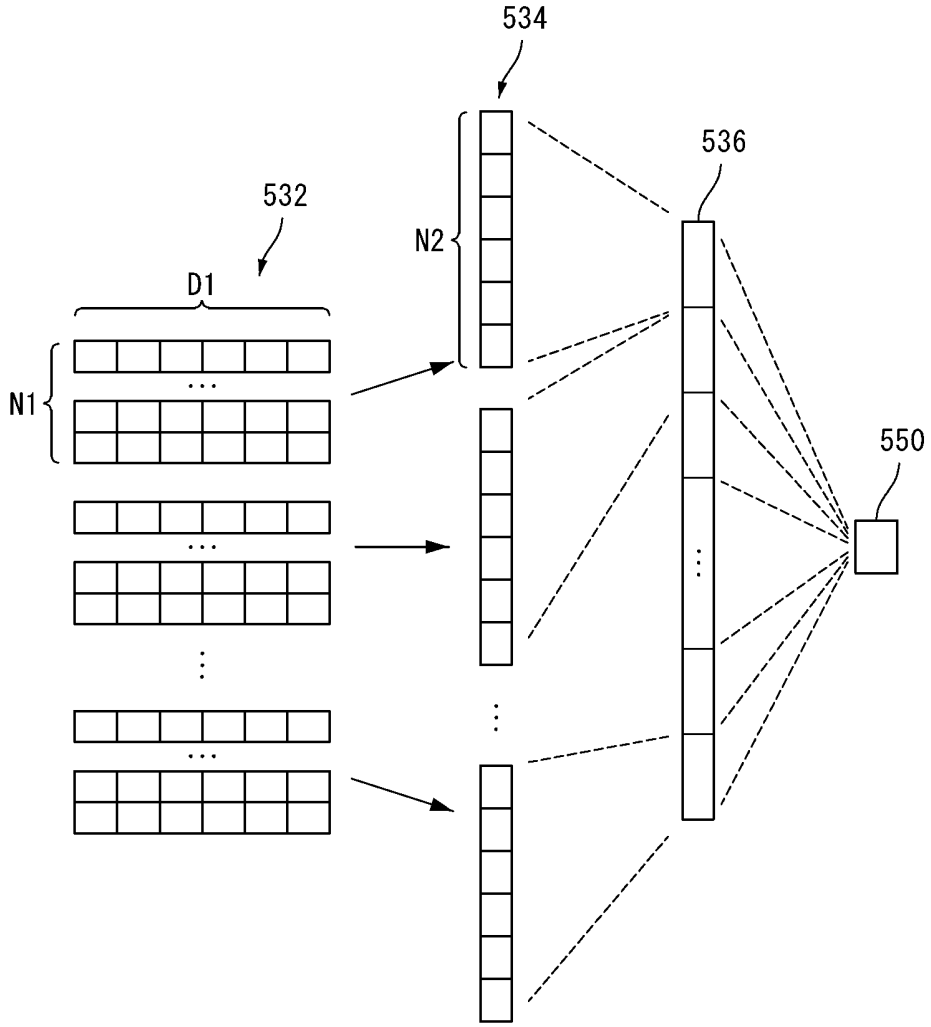
【도 6】


E5
524
Transformer encoder
Transformer encoder
Transformer encoder
522
0
1
2
3
E2
E1
[CLS] root fail -ed

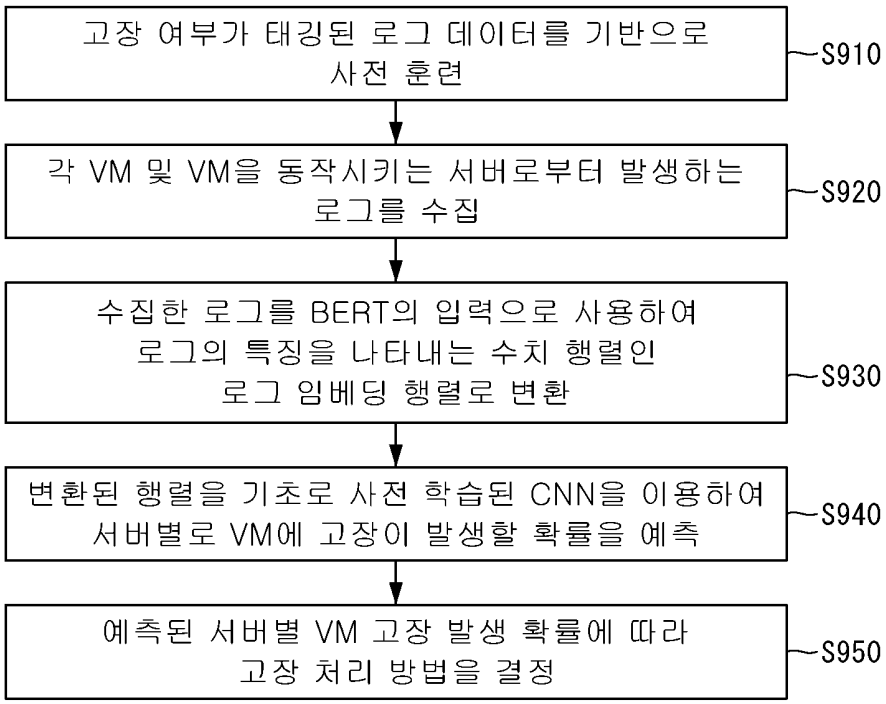
【도 7】


524
530
532
534
536
BERT Model
CNN
Max-pooling layer
FC layer
Prediction results

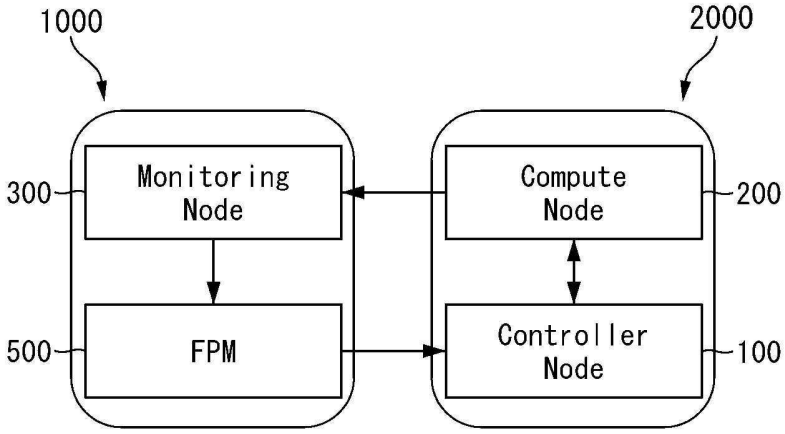
【도 8】



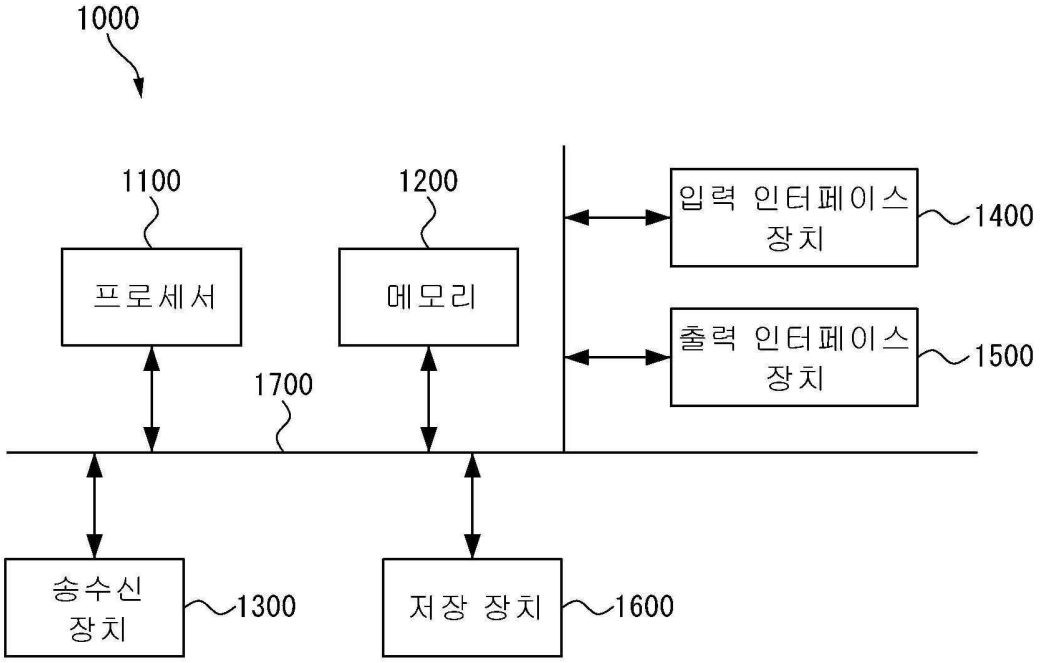
【도 9】


고장 여부가 태깅된 로그 데이터를 기반으로 사전 훈련
S910
각 VM 및 VM을 동작시키는 서버로부터 발생하는 로그를 수집
S920
수집한 로그를 BERT의 입력으로 사용하여 로그의 특징을 나타내는 수치 행렬인 로그 임베딩 행렬로 변환
S930
변환된 행렬을 기초로 사전 학습된 CNN을 이용하여 서버별로 VM에 고장이 발생할 확률을 예측
S940
예측된 서버별 VM 고장 발생 확률에 따라 고장 처리 방법을 결정
S950

【도 10】


1000
2000
Monitoring Node
300
Compute Node
200
FPM
500
Controller Node
100

【도 11】


1000
1100
프로세서
1200
메모리
입력 인터페이스 장치
1400
출력 인터페이스 장치
1500
1700
송수신 장치
1300
저장 장치
1600