

에지 네트워크 환경을 위한 딥 러닝 기반의 효율적인 IoT 데이터 처리 기법

정윤수

목원대학교 정보통신융합공학부 조교수

Efficient IoT data processing techniques based on deep learning for Edge Network Environments

Yoon-Su Jeong

Assistant Professor, Division of Information and Communication Convergence Engineering,
Mokwon University

요 약 에지 네트워크 환경에서 IoT 장치가 다양하게 활용되면서 IoT 장치에서 수집되는 정보들을 여러 응용 분야에서 활용하는 연구들이 다양하게 진행되고 있다. 그러나, 네트워크 환경(간섭, 전파방해 등)에 따라 수집되는 IoT 데이터들이 누락 또는 오류가 발생하는 상황이 빈번해지면서 정확한 IoT 데이터들을 바로 적용하기가 쉽지 않은 상황이다. 본 논문에서는 에지 네트워크 환경에서 수집되는 IoT 데이터들의 오류를 줄이기 위해서 IoT 데이터의 서명 값을 랜덤하게 생성하여 비트 형태로 보안 정보(Security Information, SI) 값만을 IoT 데이터들에 각각 할당함으로써 IoT 데이터의 신뢰성을 보장하는 관리 기법을 제안한다. 제안 기법은 IoT 장치로부터 수집되는 데이터들을 비대칭적으로 서로 연계 처리하도록 다중 해시 체인을 적용하여 IoT 데이터를 블록체인으로 묶는다. 이때, 블록 체인화된 IoT 데이터들은 딥 러닝 기반으로 상관관계 지수에 따라 가중치를 적용한 확률 함수를 사용한다. 또한, IoT 데이터의 무결성과 처리 비용을 낮추기 위해서 제안 기법은 그룹화된 IoT 데이터를 n-계층 구조로 확장 운영 가능하다.

주제어 : 딥러닝, 사물인터넷, 서명, 보안 정보, 블록체인

Abstract As IoT devices are used in various ways in an edge network environment, multiple studies are being conducted that utilizes the information collected from IoT devices in various applications. However, it is not easy to apply accurate IoT data immediately as IoT data collected according to network environment (interference, interference, etc.) are frequently missed or error occurs. In order to minimize mistakes in IoT data collected in an edge network environment, this paper proposes a management technique that ensures the reliability of IoT data by randomly generating signature values of IoT data and allocating only Security Information (SI) values to IoT data in bit form. The proposed technique binds IoT data into a blockchain by applying multiple hash chains to asymmetrically link and process data collected from IoT devices. In this case, the blockchainized IoT data uses a probability function to which a weight is applied according to a correlation index based on deep learning. In addition, the proposed technique can expand and operate grouped IoT data into an n-layer structure to lower the integrity and processing cost of IoT data.

Key Words : Deep Learning, Internet of Things, Signature, Security Information, Blockchain

*Corresponding Author : Yoon-Su Jeong(bukmunro@mokwon.ac.kr)

Received December 21, 2021
Accepted March 20, 2022

Revised January 5, 2022
Published March 28, 2022

1. 서론

최근 인공지능 관련 기술이 대두되면서 IT 기술 중 인공지능과 사물인터넷(intelligence and Internet of Things, IoT) 분야에서 핵심적인 기술로 활용되고 있다 [1-3]. IoT 장치는 센서가 장착되어 있어 IoT 장치가 수집한 데이터를 저장, 처리 및 제어하여 네트워크를 통해 전달한다.

에지 네트워크는 수집된 IoT 데이터를 에지 장치가 자체적으로 계산하거나 인접한 노드에서 계산 일부를 수행함으로써 클라우드로 전송해야 하는 데이터를 줄이는 역할을 수행한다[4]. 그러나, 네트워크의 상황에 따라 대용량의 IoT 데이터 손실을 최소화하기 위해서는 추출된 IoT 데이터의 오류를 해결할 필요가 있다[5, 6].

본 논문에서는 에지 네트워크에서 수집된 IoT 데이터의 투명성을 제공하기 위해서 IoT 데이터의 응답 지연 시간, 대역폭 소비, 스토리지 등의 최적화를 수행할 수 있는 딥러닝 기반의 효율적인 IoT 데이터 처리 기법을 제안한다. 제안 기법은 IoT 데이터들의 오류를 줄이기 위해서 IoT 데이터의 서명 값을 랜덤하게 생성하여 비트 형태로 보안 정보(Security Information, SI) 값만을 IoT 데이터들에 각각 할당함으로써 IoT 데이터의 신뢰성을 보장할 뿐만 아니라 IoT 데이터 선택도 에지 네트워크 시스템의 조건(공유 데이터, 응용 프로그램 요청, 데이터 볼륨 또는 기타 관련 정보 등)에 따라 동적으로 변경하도록 하였다. 그 이유는 제안 기법이 IoT 장치로부터 수집되는 데이터들을 상관관계 지수에 따라 가중치를 적용하여 비대칭적으로 서로 연계 처리하여 IoT 데이터를 블록체인으로 묶기 위해서이다. 또한, 제안 기법은 에지 장치에서 로드밸런스를 통해 IoT 정보를 수집하고 학습하기 때문에 네트워크 구성 요소들의 부하를 낮출 수 있다. 제안 기법은 IoT 데이터가 블록체인 형태로 전달되기 때문에 네트워크 트래픽은 최소화되지만, 확장성은 n-계층 형태로 응용할 수 있다.

이 논문의 구성은 다음과 같다. 2장은 딥러닝 개요 및 딥러닝 기반의 IoT 데이터 연구들에 관해서 설명한다. 3장에서는 딥러닝 기반의 IoT 데이터 처리 기법을 제안하고, 4장에서는 시뮬레이션을 통한 성능평가를 수행한다. 마지막으로 5장은 결론을 맺는다.

2. 관련 연구

2.1 딥러닝

최근 딥러닝은 사회적으로 큰 이슈가 되고 있으며, 다중 레이어를 사용하는 기계학습 알고리즘의 한 종류이다 [7,8]. 딥러닝은 명시적 데이터가 아닌 데이터 표현을 사용하여 학습을 수행한다. 데이터는 학습할 수 있도록 추상적 표현으로 변환하여 사용한다. 딥러닝은 지도 학습과 비지도 학습 모두 사용할 수 있지만 클라우드로 전송하는 데이터를 줄이기 위해서는 비지도 학습 중 자동 인코더를 주로 사용한다. 자동 인코더는 데이터를 먼저 인코딩한 후 학습하게 함으로써 입력 데이터와 가장 유사하게 생성하는 기법이다. 오토인코더는 Fig. 1처럼 동작하며, 인코더 과정을 통해 중간 은닉층(hidden layer) h 을 형성한다. Fig. 1의 은닉층의 결과는 디코더 과정을 통해 기본과정을 hidden 계층에 따라 반복한다.

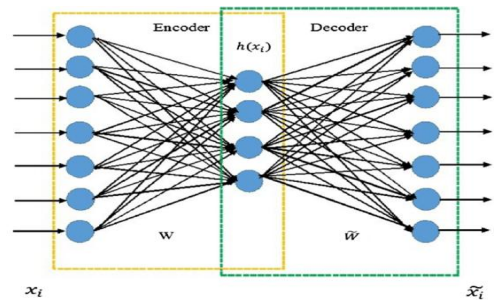


Fig. 1. Autoencoder Operation

2.2 기존 연구

H. El-Sayed et. al은 분산 클라우드 환경에서 IoT 장치, 에지 장치, 클라우드를 통합한 모델을 제시하였다 [9]. 이 모델에서는 분산 클라우드에 맞는 장치 선택, 통신 오버헤드, 보안 및 개인 정보 보호와 관련된 내용을 언급함으로써 중앙 집중식 처리의 단점을 개선할 수 있는 IoT 분산 데이터 처리 제공 방안을 제시하고 있다.

A. Sinaeepourfard et. al은 빠른 데이터 액세스를 제공하기 위해서 데이터 보존 블록을 통합한 포그 대 클라우드(Fog to Cloud) 데이터 관리 아키텍처를 제안하였다 [10]. 이 아키텍처는 데이터 전송량 및 지연 시간을 감소하는 방안은 제시하였지만, 검증을 위한 실제 실험을 하지 않은 단점이 있다.

Y. Jararweh et. al은 네트워크 에지 근처에 있는 클라우드렛의 계층적 모델을 제안하였다 [11]. 이 모델은 전력 소비와 발생 지연이 오프로드를 통해 어떻게 영향을 미치는지를 시뮬레이션을 통해 검증하였다.

L. Li et. al은 빅 데이터의 결함 탐지를 위해 딥러닝

과 포그 컴퓨팅 기반의 제조 검사 시스템을 제안하였다 [12]. 이 시스템은 기존 CNN과 달리 높은 결합 감지 정확도, 중앙 서버의 부하 감소, 전체 계산 시간 단축의 장점은 있지만, 지연 시간과 통신 오버헤드는 분석되지 못한 단점이 있다.

G. G. Jia et. al 은 에너지 소비 절감, 제어 기능 제공 및 세분화된 제어 기능 제공을 목적으로 스마트 시티를 위한 가로등 시스템을 제안하였다 [13]. 이 시스템은 SSL 시스템이 사전 정의된 다양한 정적 상태를 이해함으로써 문제 감지 및 유지보수를 개선할 수 있음을 입증하고 있다. 그러나, 클라우드 서버와 비교하면 포그 서버의 이점을 구체적으로 고려하지 않고 비교하지 못한 단점을 가지고 있다.

J. He et. al 은 스마트 시티 애플리케이션을 위한 대규모 데이터 분석 서비스를 갖춘 다 계층 포그 컴퓨팅 모델을 제안하였다 [14], 이 모델은 데이터 분석을 지원하고 분석 서비스 유틸리티를 극대화하기 위한 오프로드, 리소스 할당 및 서비스 품질 인식 작업이 허용되고 있다.

X. Wang et. al은 이벤트에 대한 평균 응답 시간을 최소화하기 위한 포그 지원 실시간 교통관리 시스템을 제안하였다 [15]. 이 시스템은 차량, 클라우드릿, 클라우드 서버로 구성되어 있으며, 실험을 통해 포그 노드 수와 서비스 요청의 영향을 조사하고 있다.

3. 딥러닝 기반의 IoT 데이터 분산 처리기법

3.1 네트워크 환경

네트워크 환경에서 딥러닝 기반으로 IoT 데이터를 수집 및 처리하기 위한 제안 기법의 네트워크 모델은 Fig. 2와 같다. 제안 기법은 IoT 장치들이 네트워크 환경에 분산 배치되었고 IoT 장치 간 서로 연결되어 있지 않다고 가정하고 있다. 제안 기법은 네트워크 모델에서 에지와 클라우드 컴퓨팅의 위치에 대한 투명성을 제공하기 위해서 응답 지연 시간, 대역폭 소비, 스토리지 등의 최적화를 통해 서로 다른 영역에 있는 IoT 장치의 데이터를 에지 노드, 중간 노드 또는 클라우드에서 처리할 수 있도록 하고 있다. 또한, IoT 노드 중 계산 노드의 선택은 시스템의 조건(공유 데이터, 응용 프로그램 요청, 데이터 볼륨 또는 기타 관련 데이터 등)에 따라 동적으로 변경되도록 하였다.

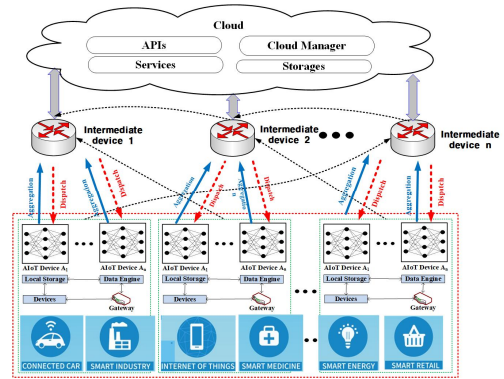


Fig. 2. Network Model of Proposed scheme

제안 기법에서 데이터 처리 과정은 네트워크에 분포된 IoT 장치로부터 수집된 데이터를 에지 노드에서 사전 처리한 후 로드밸런스 장치를 통해 클라우드 서버에 전달된다. Fig. 1처럼 제안 기법은 로드밸런스 과정을 통해 IoT 장치로부터 수집된 데이터를 비지도 학습으로 처리하기 때문에 클라우드 서버에서는 IoT 데이터의 학습이 완료될 수 있도록 필요한 추가 파라미터를 업데이트한다.

3.2 IoT 데이터의 특징 추출 및 선택 방법

제안 기법은 에지 네트워크 환경에서 수집되는 IoT 데이터를 에지 장치에서 Table 1과 같은 함수들을 이용하여 추출된 IoT 데이터의 평균, 분산, 최대, 최소, 루트 평균 제곱 및 에너지 등의 특징들을 추출한다.

Table 1. IoT Data Features Extract and Selection

Feature	Equation
Mean	$\bar{X} = \frac{1}{n} \sum_{i=1}^N X_i$
Variance	$\sigma^2 = \frac{1}{n} \sum_{i=1}^N (X_i - \bar{X})^2$
Maximum	$X_{\max} = \max_{i=1,2,\dots,N} \{X_i\}$
Minimum	$X_{\min} = \min_{i=1,2,\dots,N} \{X_i\}$
Root mean square	$S = \sqrt{\frac{1}{n} \sum_{i=1}^N X_i^2}$
Energy	$E = \frac{1}{n} \sum_{i=1}^N X_i ^2$

제안 기법에서는 각 IoT 데이터를 심층 신경망 모델을 이용하여 완전 연결 계층(Fully-Connected Layer) 형태로 구성한 후 입력 데이터와 출력 데이터를 설정하

여 학습하도록 하고 있다. Hidden Layer는 게이트웨이 역할을 하는 중간 장치 수에 따라 1에서 3을 설정한다. IoT 데이터가 중간 장치를 거치면서 가중치는 IoT 데이터의 연계 정도에 따라 값이 업데이트된다.

제안 기법은 특징 관련 문제를 결정하기 위해서 순차적 특징 선택(Sequential Feature Selection, SFS) 알고리즘을 사용하여 d 차원을 k 차원의 특징 부분공간 ($k < d$)으로 줄이고 있다. k 개의 특징 서브 셋에 도달할 때까지 분류기의 성능에 기초하여 한 번에 하나의 특징을 제거하거나 추가함으로써 계산 효율을 향상시켰다.

3.3 IoT 데이터 연계 처리

제안 기법은 IoT 장치로부터 수집되는 데이터를 예지 서버에서 IoT 데이터를 사전에 서로 연계 처리하기 위해서 각 IoT 데이터의 서명 값을 랜덤하게 생성하여 비트 형태로 보안 정보(Security Information, SI) 값을 각 IoT 데이터에 할당받는다. 그 이유는 네트워크 구성 요소 간 부하를 낮아지도록 할 뿐만 아니라 구성 요소들의 역할을 강화할 수 있기 때문이다.

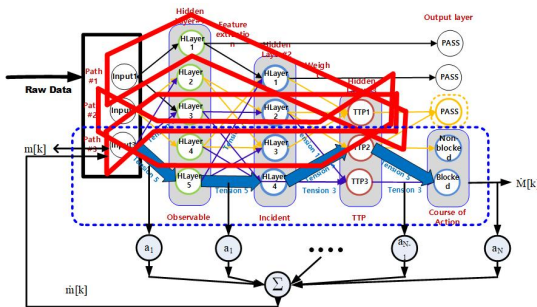


Fig. 3. IoT Data Link Information for Asymmetric Processing

Fig. 3는 IoT 데이터의 서명 값과 IoT 데이터를 비트 형태로 서로 연계하는 과정을 나타내고 있다. Fig. 3를 통해 IoT 데이터의 서명 정보 값은 랜덤하게 할당받지만, IoT 데이터는 비트 형태로 보안 정보 값을 할당받는다. 이때, IoT 데이터는 비대칭 처리를 수행하기 때문에 IoT 데이터의 비트 정보들은 쌍대 비교 행렬 d_{ij} 의 i 번째 벡터와 j 번째 벡터 사이에서 연계 정보 R_{ij} 을 식 (1) 처럼 만들어진다.

$$R_{ij} = \sum_{i=1, k=1}^n D_{ik}^x \times D_{jk}^y \quad (1)$$

제안 기법은 벡터화된 IoT 데이터가 서로 비대칭적으로 직교하여 그룹화한 후 가중치를 식 (2) 처럼 계산하여 IoT 데이터를 확인한다.

$$y = wx + \beta \quad (2)$$

여기서, w 는 가중치를 의미하고 β 는 편향 값을 의미한다.

제안 기법은 비대칭적으로 서로 연계 처리한 IoT 데이터를 클라우드 서버에서 다중 해쉬 체인을 통해 블록 체인으로 묶는다. 이때, 제안 기법은 비대칭적으로 그룹화된 IoT 데이터에 식 (3)과 같은 확률 함수를 적용하여 IoT 데이터를 저장하고 관리한다.

$$E(B_i^x) = - \sum_{i=1}^n \frac{1}{n} \log \frac{1}{n} = \log n \quad (3)$$

여기서, n 은 IoT 데이터 수를 의미하고, i 는 연계 정보의 i 번째 위치를 의미하고, x 는 비대칭 연계 정보의 그룹 인덱스 값을 의미한다.

3.4 IoT 데이터 처리 알고리즘

제안 기법에서 사용되는 IoT 데이터 처리 알고리즘은 IoT 데이터의 레이블 유·무에 따라 학습 종류가 달라진다. 제안 기법은 IoT 데이터 처리에 따라 odd/even으로 해쉬 값을 IoT 데이터의 첫 번째와 마지막에 추가하여 IoT 데이터를 블록 형태로 연계 처리하기 때문에 IoT 데이터의 처리 효율성이 높다. Table 2는 IoT 데이터 처리를 위해서 제안 기법에서 사용되는 알고리즘이다.

Table 2는 IoT 장치에서 생성되는 데이터를 딥러닝으로 처리할 때 IoT 데이터 장치 종류에 따라 최적화된 모델을 생성하는 데 사용된다. Table 2는 IoT 모델을 생성하기 위해서 IoT 장치 종류(모델), IoT 장치 타입, IoT 장치 목록 및 훈련 데이터를 입력한다. IoT 훈련(train) 데이터는 IoT 장치 종류에 따라 별도를 처리되며, 처리된 정보는 1로 표시하고 처리되지 않은 정보는 0으로 표시한다. IoT 데이터는 특징(features)과 항목(레이블)별로 분리하여 모델을 훈련하는 데 사용된다. Table 2를 통해 모델이 생성되면 모델 이름은 IoT 장치 이름과 날짜, 인덱스 정보 등을 이용한다. 모델 예측은 IoT 데이터

가 저장된 순서대로 가져와 생성된 모델을 통해 처리된다. IoT 데이터의 레이블에 따라 모델이 분류되면 학습은 IoT 데이터의 거리 및 확률 분포에 따라 학습이 이루어진다.

Table 2. IoT Data Process Distribution Algorithm

<p>Input: IoT Data, IoT Device Type Information Output : Optimized IoT Data Processing Classification Model</p>
<p>1: IoT data is input like $I=\{i_1, i_2, \dots, i_n\}$. 2: IoT data types are input, such as $T=\{t_1, t_2, \dots, t_m\}$. 3: Features and labels are initialized using IoT data(I) and type (T).</p> $F = \{f_1, f_2, \dots, f_n\}$ $L = \{l_1, l_2, \dots, l_m\}$ <p>4: IoT data is sequentially mixed using the shuffle() function. $I = shuffle(I)$</p> <p>5: IoT data (I) is optimized using a <i>MinMaxScaler</i>() function according to the state of IoT data. if(function ==1) $I = MinMaxScaler(I)$ else $I = StandardScaler()$ end if</p> <p>6: Repeat while finding a type suitable for IoT data among IoT data types. for t $\in T$ of IoT data do 7: IoT data is generated by combining IoT features and labels. $I = F + L$ 8: If the IoT item (Label) is the same as the IoT data, the IoT item (Label) is designated as 1, otherwise it is designated as 0. for l $\in L$ do if l = t $l = 1$; else $l = 0$; end if end for</p> <p>9: The model (<i>RandomForest</i>(), <i>KNN_Regression</i>()) is determined according to the processed IoT data state. if status = 1 Model = <i>RandomForest</i>(); else Model = <i>KNN_Regression</i>(); end if</p> <p>10. The model is optimized using features (F) and items (L) of IoT data. $Model.fit(F, L)$</p> <p>11. The IoT data processing model is determined. Select Model end for</p>

4. 평가

4.1 실험 환경

실험 환경은 Windows 운영체제에서 OPNet과 Anaconda의 Python3를 사용하여 실험하였다. 원시 데이터 세트의 각 기능은 값 범위가 서로 달라서 Sklearn 라

이브러리에 내장된 *MinMaxScaler*()와 *StandardScaler*()을 사용하여 데이터에 대한 최소-최대 스케일링을 수행하여 모든 데이터의 범위를 [0, 1] 가 되게 하였다. 분류자(classifier)는 Sklearn에 내장된 랜덤 포레스트(Random Forest)를 사용하여 파라미터들(n_composators(=200), criterion(=entropy), max_depth(=20), min_samples_leaf(=5), L2 regularizer, constant laraning($\lambda=10^{-5}$) rate Epoch(=100) 등)을 설정하였다.

4.2 성능평가

4.2.1 네트워크 계층별 요청 수 대비 처리 수

네트워크 계층(P2P, Cloudlet, 로드 밸런스장치, Core Cloud)별 IoT 데이터 처리 수를 시뮬레이션 한 결과 Fig. 4와 같다. Fig. 4의 결과처럼 클라우드 환경의 각 네트워크 계층(P2P, Cloudlet, 로드밸런스 장치, Core Cloud)별 IoT 데이터 처리 수를 분석한 결과, P2P 데이터 요청 수가 증가할수록 IoT 데이터를 사전 처리하는 로드밸런스 장치의 처리 수가 가장 높았으며, Cloudlet의 처리 수는 IoT 데이터 수가 증가할수록 크게 IoT 데이터를 처리하지 못하는 결과를 얻었다. 이 같은 결과는 Cloudlet의 성능이 높지 않기 때문에 나타난 것으로 분석되었으며, Cloudlet에서 IoT 데이터 간 동기화를 수행했을 때, IoT 데이터의 오류율 또한 로드밸런스 장치보다 18.8% 정도 낮았다.

4.2.2 네트워크 계층에 따른 처리 시간

네트워크 계층(P2P, Cloudlet, 로드 밸런스장치, Core Cloud)별 IoT 데이터 처리 수를 증가하면서 IoT 데이터를 처리하는 처리 시간을 성능 분석한 결과 Fig. 5과 같다. Fig. 5의 결과처럼, 물리적인 환경에 따라 IoT 데이터 처리 시간이 상이하게 나타났지만, 로드 밸런싱 장치에서 IoT 데이터를 트래픽 별로 블록체인으로 연계 처리한 결과 IoT 장치와 서버 간 Hidden 레이어(5~30)에 따라 최대 19.9%의 처리 시간 차이가 나타났다. 이 같은 결과는 IoT 데이터의 특징(features)과 항목(레이블)별로 분리하여 모델을 훈련하여 IoT 데이터의 거리 및 확률 분포에 따라 학습이 이루어졌기 때문이다.

4.2.3 네트워크 계층에 따른 정확도

네트워크 계층(P2P, Cloudlet, 로드 밸런스장치, Core Cloud)별 IoT 데이터 처리 수를 증가하면서 IoT

데이터 처리 정확도를 분석한 결과 Fig. 6과 같다. Fig. 6의 결과처럼, IoT 데이터가 전달되는 물리적인 환경에서 IoT 데이터의 무결성을 보장받는 네트워크 계층이 그렇지 못한 계층 보다 평균 16.5% 정확도가 높게 나타났다. 이 같은 결과는 IoT 데이터를 사전에 서로 연계 처리하기 위해서 각 IoT 데이터의 서명 값을 랜덤하게 생성하여 비트 형태로 보안 정보(Security Information, SI) 값을 각 IoT 데이터에 할당하여 네트워크 구성 요소 간 부하를 낮추었기 때문에 나타난 결과이다. 또한, IoT 데이터 처리에 따라 해쉬 값을 블록 형태로 연계 처리하기 때문에 IoT 데이터의 처리 효율성이 높아졌기 때문에 처리 시간이 줄어들었다.

5. 결론

클라우드 환경에서 IoT 장치에서 수집되는 정보들은 다양한 분야에서 활용되고 있다. 그러나, 물리적인 네트워크 환경(간섭, 전파방해, 자연재해 등)에 따라 IoT 데이터들이 누락 또는 오류가 자주 발생하는 문제점이 발생하고 있다. 본 논문에서는 IoT 데이터의 처리시간을 최적화할 수 있는 딥러닝 기반의 효율적인 IoT 데이터 처리 기법을 제안한다. 제안 기법은 IoT 데이터의 서명 값을 랜덤하게 생성하여 IoT 데이터들에 보안 정보(Security Information, SI) 값을 각각 할당함으로써 IoT 데이터의 신뢰성을 보장하였다. 또한, 제안 기법은 IoT 데이터에 상관관계 지수에 따른 가중치를 적용하여 IoT 데이터를 비대칭적으로 블록체인 하였다. 그리고, 제안 기법은 IoT 데이터를 블록체인 형태로 전달하기 때문에 네트워크 트래픽을 최소화하도록 유지하여 확장성이 가능하여지도록 하였다. 실험 평가 결과, 제안 기법은 Cloudlet에서 IoT 데이터 간 동기화를 수행했을 때, IoT 데이터의 오류율 또한 로드밸런스 장치보다 18.8% 정도 낮았으며, Cloudlet에서 IoT 데이터 간 동기화를 수행했을 때, IoT 데이터의 오류율 또한 로드밸런스 장치보다 18.8% 정도 낮았다. 또한, Cloudlet에서 IoT 데이터 간 동기화를 수행했을 때, IoT 데이터의 오류율 또한 로드밸런스 장치보다 18.8% 정도 낮았다. 향후 연구에서는 기존 연구 결과를 이용하여 물리적인 환경을 확대하여 IoT 데이터의 처리 수와 정확도를 다양한 알고리즘에 적용하여 비교분석을 수행할 계획이다.

REFERENCES

- [1] Q. Liu, L. Cheng, T. Ozcelebi, J. Murphy & J. Lukkien, (2019). Deep Reinforcement Learning for IoT Network Dynamic Clustering in Edge Computing. *Proceedings of the 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pp. 600–603.
- [2] H. Ye & G. Y. Li, (2018). Deep reinforcement learning for resource allocation in v2v communications. *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*. IEEE, pp. 1–6.
- [3] X. Sun & N. Ansari, (2016). EdgeIoT: Mobile edge computing for the internet of things. *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29.
- [4] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal & H. Flinck, (2017). Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine*, vol. 55, no. 3, pp. 38–43.
- [5] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty & C. T. Lin, (2018). Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment. *IEEE Access*, vol. 6, pp. 1706–1717.
- [6] B. Tang, Z. Chen, G. Hefferman, S. Pei, T. Wei, H. He & Q. Yang, (2017). Incorporating intelligence in fog computing for big data analysis in smart cities. *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2140–2150.
- [7] Z. Hao, E. Novak, S. Yi & Q. Li, (2017). Challenges and software architecture for fog computing. *IEEE Internet Computing*, vol. 21, no. 2, pp. 44–53.
- [8] Y. LeCun, Y. Bengio & G. Hinton, (2015). Deep learning. *Nature*, vol. 521, no. 7553, pp. 436–444.
- [9] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty & C. T. Lin, (2018). Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment. *IEEE Access*, vol. 6, pp. 1706–1717.
- [10] A. Sinaeepourfard, J. Garcia, X. Masip-Bruin & E. Marin-Tordera, (2018). Data preservation through fog-to-cloud (f2c) data management in smart cities. *Proceedings of the IEEE 2nd International Conference on Fog and Edge Computing*, pp. 1–9.
- [11] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub & E. Benkhelifa, (2016). The future of mobile cloud computing: integrating cloudlets and mobile edge computing. *Proceedings of the 23rd International Conference on Telecommunications*, pp. 1–5.
- [12] L. Li, K. Ota & M. Dong, (2018). Deep learning for smart industry: Efficient manufacture inspection system with fog computing. *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4665–4673.

- [13] G. G. Jia, G. G. Han, A. Li & J. Du, (2018). Ssl: Smart street lamp based on fog computing for smarter cities. *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4995–5004.
- [14] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou & Y. Zhang, (2017). Multi-tier fog computing with large-scale IoT data analytics for smart cities. *IEEE Internet Things Journal*, vol. 5, no. 5, pp. 677–686.
- [15] X. Wang, Z. Ning & L. Wang, (2018). Offloading in internet of vehicles: A fog-enabled real-time traffic management system. *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4568–4578.

정 윤 수(Yoon-Su Jeong)

[상위]



- 1998년 2월 : 대학교 전자계산학과 학사
- 2000년 2월 : 충북대학교 전자계산학과 석사
- 2008년 2월 : 충북대학교 전자계산학과 박사
- 2012년 3월 ~ 현재 : 목원대학교 정보

통신공학과 조교수

- 관심분야 : 유·무선 통신 보안, 정보보호, 바이오인포매틱, 헬스케어, 빅 데이터, 클라우드 컴퓨팅
- E-Mail : bukmunro@mokwon.ac.kr