

# 분산 AIoT 환경에서 합성곱신경망 기반 계층적 IoT Edge 자원 할당 및 관리 기법

정윤수

목원대학교 게임소프트웨어공학과 교수

## Hierarchical IoT Edge Resource Allocation and Management Techniques based on Synthetic Neural Networks in Distributed AIoT Environments

Yoon-Su Jeong

Professor, Department of Game Software Engineering, Mokwon University

**요약** 대다수의 IoT 기기들은 이미 AIoT를 사용하고 있지만, AI 애플리케이션을 구축하기 위해서는 아직 해결해야 할 문제가 많이 남아 있다. 본 연구에서는 IoT 에지 자원을 보다 효과적으로 분산하기 위해 머신러닝 기반의 IoT 에지 자원 관리 기법을 제안한다. 제안 기법은 머신러닝을 이용하여 IoT 에지 자원 동향을 파악함으로써 IoT 자원의 할당을 지속적으로 개선하며, 최적화된 IoT 자원은 머신러닝 컨볼루션을 활용하여 항상 변화하는 IoT 에지 자원을 안정적으로 유지한다. 제안 기법은 각각의 머신러닝 기반 IoT 에지 자원을 이전 패턴의 자원과 함께 해시 값으로 저장함으로써 분산된 AIoT 맥락에서 공격 패턴으로 자원을 효과적으로 검증한다. 실험 결과에서는 IoT Edge 리소스의 무결성을 검증하기 위해서 이질적인 계산 하드웨어가 있는 복잡한 환경에서 잘 동작하는지 세 가지 다른 테스트 시나리오에서 에너지 효율성을 평가하였다.

**주제어** : 가상화 기술, AIoT, 블록체인, 리소스 관리, 머신러닝

**Abstract** The majority of IoT devices already employ AIoT, however there are still numerous issues that need to be resolved before AI applications can be deployed. In order to more effectively distribute IoT edge resources, this paper propose a machine learning-based approach to managing IoT edge resources. The suggested method constantly improves the allocation of IoT resources by identifying IoT edge resource trends using machine learning. IoT resources that have been optimized make use of machine learning convolution to reliably sustain IoT edge resources that are always changing. By storing each machine learning-based IoT edge resource as a hash value alongside the resource of the previous pattern, the suggested approach effectively verifies the resource as an attack pattern in a distributed AIoT context. Experimental results evaluate energy efficiency in three different test scenarios to verify the integrity of IoT Edge resources to see if they work well in complex environments with heterogeneous computational hardware.

**Key Words** : Virtualization technology, AIoT, Blockchain, Resource management, Machine learning

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

Received August 20, 2023

Revised September 13, 2023

Accepted September 20, 2023

Published September 30, 2023

## 1. 서론

현재 IoT 시스템의 문제(보안, 상호운용성, 확장성, 신뢰성 등)를 해결하기 위해 AIoT는 AI를 사물인터넷(IoT)과 통합하여 인간의 개입 없이 작업을 자동화하고 있다[1]. 대부분의 IoT 장치가 메모리, 컴퓨팅 파워, 배터리 수명과 같은 제한된 리소스를 가지고 있지만 AI 애플리케이션은 상당한 리소스를 요구하고 제한된 리소스로 IoT 장치에서 실행하기가 어렵다. IoT 장치에서 수집된 데이터를 처리 및 분석하고 지식을 적용하여 현명한 판단을 내리고 즉각적인 조치를 취하기 위해서는 IoT Edge 리소스 또는 저지연 고성능 컴퓨팅 기능이 필요하다.

현재 데이터 센터와 비교하여 AIoT의 제한된 리소스를 극복하기 위해 연구자들은 리소스를 보다 효율적으로 분산할 수 있는 가상화 기술을 최근에 활용하고 있다[2,3].

본 연구에서는 IoT 리소스 성능 향상을 위해서 머신러닝 기반의 IoT 에지 자원 관리를 제안한다. 제안 기법은 IoT 리소스 자원의 특성을 추출하여 IoT 에지 리소스 패턴을 검출하기 위해서 머신러닝을 이용하여 IoT 리소스 할당을 지속적으로 최적화한다. 제안 기법은 머신러닝의 컨볼루션 과정을 통해 IoT 에지 리소스가 동적으로 변화하는 경우에 IoT 에지 리소스를 일관되게 유지할 수 있다. 제안 기법은 머신러닝 기반의 각 IoT 에지 리소스를 이전 패턴의 자원과 연계하여 해시값으로 저장함으로써 분산형 AIoT 환경에서 공격 패턴으로 자원을 성공적으로 검증한다.

이 논문의 구성은 다음과 같다. 2장에서는 IoT 에지 리소스 관련 기존 연구에 대해서 설명한다. 3장에서는 머신러닝 기반 IoT 에지 리소스 관리 기법을 제안하고, 4장에서는 제안 기법모델의 실험환경을 구축하여 성능 평가를 수행한다. 마지막으로 5장에서 결론을 맺는다.

## 2. 관련연구

Mostafa et al. 은 가상화된 안개(fog) 인프라에 IoT 애플리케이션을 배치하기 위한 WOA(Whale Optimization Algorithm) 메타 휴리스틱 기법을 활용한 효율적인 서비스 배치 알고리즘 제안하였다[4]. 이 알고리즘은 자동으로 포그 노드를 배치하기 위해서 MAPE-k(Monitor-Analyze-Plan-Execute over a

shared Knowledge) 루프라는 공유 지식 기반을 사용하기 때문에 기존 알고리즘(예, GA, SA, PSO 등)보다 검색 프로세스 속도가 높고 로컬 최적화 회피 기능을 제공한다. 또한, 이 알고리즘은 탐색 및 개발 단계에서 최적화 문제를 해결하기 때문에 검색 비용을 낮추기 위한 곳에 적합하다.

Li et al. 은 AIoT 클라우드 에지 컴퓨팅 환경에서 데이터 검증률을 향상시키기 위해서 블록체인 기반의 데이터 무결성 검증 체계를 제안하였다 [5]. 이 검증 체계는 2단계(데이터 업로드 단계와 데이터 검증 단계)로 구분되며, 계산 및 통신 비용을 줄이기 위해서 해시 연산만을 사용한다. 또한 이 검증 체계는 하이퍼레저 쏘투스(Hyperledger Sawtooth) 기반으로 시뮬레이션된 클라우드 에지 환경에서 두 개의 최첨단 시스템에 대해서 높은 검증률을 가지며 데이터 무결성을 효율적으로 검증하였다.

Masoumeh et al. 은 IoT 애플리케이션을 배포하기 위한 포그 인프라에서 효율적인 딥 러닝 기반의 리소스 자동 스케일링 접근 방식을 제안하였다 [6]. 이 방식은 IoT 요청을 처리하기 위한 포그 리소스를 프로비저닝하기 위한 자동 스케일링 작업을 식별하기 위해서 딥 러닝 기술을 의사 결정자로 활용하였다. 또한, 이 방식은 안개(fog) 환경에서 IoT 요청을 처리하기 위한 평가 항목으로 딥 러닝 기술을 사용하는 효율적인 리소스 자동 스케일링을 비용, 안개 장치 활용도, 네트워크 사용률, 지연 위반 등이 있다.

Fariba et al. 은 MEC 환경에 적합한 3계층 아키텍처(IoT 장치, 에지 서버, 클라우드)를 제시하였다[7]. 이 기법은 자율 관리(MAPE 제어 루프)를 사용하여 원하는 콘텐츠를 수집하고, 에지 서버 및 네트워크의 컨텍스트 정보를 사용하여 에지 장치 또는 클라우드에 대한 최적의 대상을 찾는다. 특히, 이 기법은 기계 학습 방법으로 BLA(Bayesian learning automata) 접근법을 사용하여 알고리즘을 최적화하였다. 이 기법은 에너지 소비, 실행 비용, 네트워크 사용, 지연, 모바일 유형, 모듈 크기 및 오프로드 작업의 시간 간격과 같은 일부 메트릭을 기반으로 평가한 결과, 기존 연구보다 성능이 뛰어났고 컨텍스트 방법에 따라 오프로드하는 것을 보여주고 있다.

Battula et al. 은 Smart Contract 및 블록체인 기술을 활용하여 SLA(service-level agreements) 관리 프레임워크를 제안하였다[8]. 이 프레임워크는 블록체인

사실 네트워크에서 수행된 실험에서 얻은 결과로 안개 속에서 정확하고 효율적인 SLA 실행을 보장할 수 있다. 또한 이 프레임워크는 거래 비용 및 시간 측면에서 기존 연구보다 성능이 우수한 것이 특징이다.

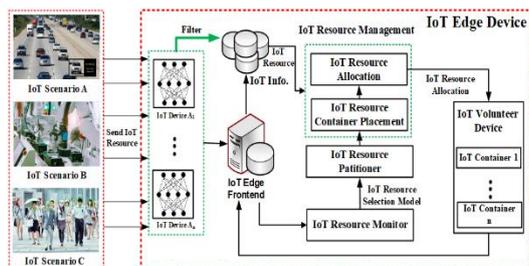
Mehran et al. 은 가상 시스템(VM)의 물리적 호스트 간에 적합한 대상을 선택하기 위한 효과적인 마이크로 유전 알고리즘을 제안하였다 [9]. 이 알고리즘은 시뮬레이션 환경에서 마이크로 유전 접근 방식이 다른 방법에 비해 전력 소비 측면에서 개선되었음을 보여주고 있다.

Mahboubeh et al. 은 포그 노드에 IoT 애플리케이션을 배포하는 효율적인 자동 서비스 배치 접근 방식을 제안하였다 [10]. 이 방식은 IoT 요청 서비스에 응답하기 위해 다양한 QoS 및 QoE 수준을 가진 이기종 포그 플랫폼의 다중 사용자/다중 서버 환경을 고려하고 있다. 또한, 이 기법은 MAPE-K 루프 개념을 활용하여 서로 다른 안개 실행 노드 간의 서비스 배치 문제를 자동으로 모델링하였다. 그리고, 이 기법은 포그 노드에서 가장 최적의 애플리케이션 배포를 달성하기 위해 GWO(gray wolf optimization) 알고리즘 적용한 것이 특징이다.

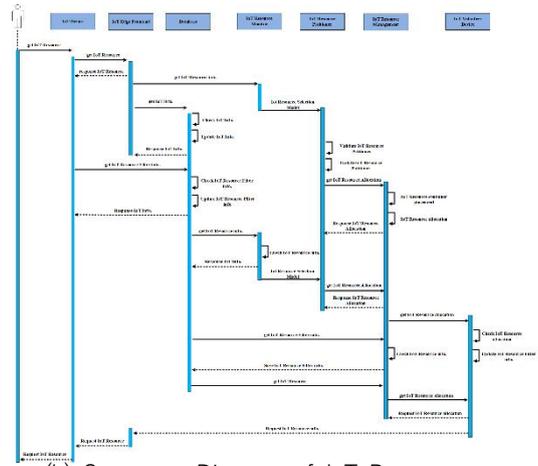
### 3. 머신러닝 기반 IoT 리소스 관리 기법

#### 3.1 개요

분산 AIoT 환경에서는 IoT 장치에서 송·수신되는 IoT 정보의 정확도 이외에 공격자에 의해 IoT 리소스가 탈취변경사제 되는 부분을 예방하기 위한 연구가 다양하게 진행되고 있다. IoT 정보들은 유선 환경보다는 무선 환경에서 대부분 IoT 정보들이 송 수신되기 때문에 안전성과 관련된 문제들이 중요시 여겨지고 있다. 최근 IoT 장치들은 분산 환경에서 클라우드 서비스와 함께 사용되고 있어 클라우드 정책 및 접근 권한에 따라 IoT 장치들이 다양하게 지원되고 있다.



(a). IoT Resource Operation



(b) Sequence Diagram of IoT Resource

Fig. 1. Proposal scheme based on Machine Learning

제안 기법은 AIoT 환경에서 단계적으로 IoT 정보들을 안정적으로 수집하기 위해서 Fig. 1 처럼 IoT 리소스 블록을 다중 블록 체인으로 묶어 처리하고 있으며, 서로 다른 IoT 리소스 중에서 특정 IoT 리소스만을 안정적으로 처리하기 위해서 IoT Edge 장치에서 IoT 리소스만을 인식할 수 있는 추가 리소스 정보를 추출하여 IoT Volunteer에서 처리할 수 있도록 IoT 리소스 손실을 최소화하고 있다.

Fig. 1은 다양한 AIoT 환경(스마트 시티, 스마트 홈, 스마트 인더스트리, 스마트 헬스케어, 웨어러블 등)에서 AIoT 장치의 기능 요구사항을 충족시키기 위해서 클라우드 컴퓨팅 플랫폼과 결합하여 AIoT 애플리케이션에서 생성된 IoT 리소스들을 클라우드 서버(또는 데이터 센터)로 전달함으로써 네트워크 지연을 해결하고 있다. Fig. 1에서 블록체인은 AIoT 장치들의 IoT 리소스들을 분산 처리할 때 시간 흐름에 따라 순차적으로 연결된 체인의 구조를 가지고 있기 때문에 모든 IoT 리소스들의 정보를 대조하고 확인한다. Figure 1에서 블록체인은 여러 IoT 리소스들이 분할 및 저장하기 때문에 IoT 리소스들을 위변조하기 위해서는 블록체인에 참여한 모든 IoT 장치들을 모두 공격해야 하기 때문에 사실상 해킹이 불가능하여 보안성이 보장되는 동시에 IoT 리소스들의 신뢰성이 보장된다.

### 3.2 IoT 에지 리소스 관리

제안 기법의 IoT 에지 리소스의 계층적으로 서로 연결하기 위해서 IoT 에지 리소스 별로 일정 확률의 가중치가 Fig. 2처럼 부여하여 다단계 그룹이 구성된다.

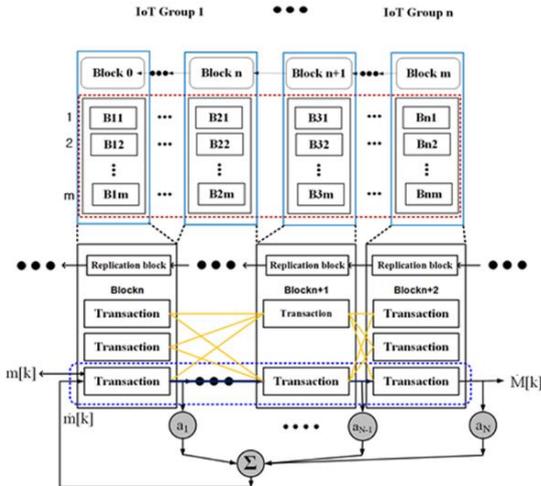


Fig. 2. Blockchain-based IoT Edge Resource Connection Structure

Fig. 2에서 번째 IoT Edge 리소스 블록이 일 때, 제안 기법은 IoT Edge 리소스 블록의 오류를 (= -)처럼 예측할 수 있기 때문에 IoT Edge 리소스 블록 중 임의의 블록을 추정하여 공격 패턴과 관련된 IoT Edge 리소스의 오류 블록을 추가하여 IoT Edge 리소스 블록을 생성함으로써 IoT Edge 리소스의 무결성을 검증할 수 있다. IoT Edge 리소스 블록을 연결하기 위한 알고리즘은 Table 1과 같다.

Table 1. IoT Edge Resource Block Connection Algorithm

|               |                                                              |
|---------------|--------------------------------------------------------------|
| <b>Input</b>  | IoT Edge Resource Block                                      |
| <b>Output</b> | IoT Edge Resource Multi-Block Connection                     |
| 1:            | for all IoT Edge resource block do                           |
| 2:            | Generate random IoT Edge resource blocks in n-bit form using |
| 3:            | IoT Edge checks its blocks                                   |
| 4:            | if Replication block data presents in blocks then            |
| 5:            | Makes tree in replication block                              |

|     |                                                  |
|-----|--------------------------------------------------|
| 6:  | hash resource block from tree                    |
| 7:  | Inform other IoT devices                         |
| 8:  | else                                             |
| 9:  | Request replication resource block               |
| 10: | end if                                           |
| 11: | End for                                          |
| 12: | <u>IoT Edge resource generates the new block</u> |

Table 2은 IoT Edge 리소스 블록을 이용하여 서로 연결하는 동안 오류가 발생했을 때 IoT Edge 서버는 IoT Edge 리소스 블록들에 대한 오류 판단을 수행하고 IoT Edge 리소스의 연결 블록을 구한다. 이렇게 함으로써 IoT Edge 리소스 블록에 대한 연결 정확도를 향상시킬 수 있다.

### 3.3 IoT 에지 리소스 접근 방법

AIoT 에지 컴퓨팅 환경에서 IoT 에지 리소스 관리는 최적의 성능과 효율적인 리소스 활용을 보장하는 데 매우 중요하다. AIoT 환경에서 IoT 에지 리소스 관리의 예를 위해서 센서, 카메라 및 에지 서버와 같은 여러 에지 장치로 구성된 AIoT 에지 컴퓨팅 환경을 가정한다. 이러한 장치는 에지 게이트웨이를 통해 클라우드 인프라에 연결되어 장치와 클라우드 간의 데이터 교환 및 통신이 가능하다. 이 환경에서 IoT 에지 리소스를 관리하기 위해 자동화 및 수동 리소스 관리 기술을 조합하여 사용할 수 있다. 예를 들면 자동화된 리소스 할당, 수동 리소스 할당, 리소스 모니터링, 리소스 최적화, 리소스 확장 등이 있다.

자동화된 리소스 할당: 머신 러닝 알고리즘을 사용하여 현재 사용량과 예상 수요를 기반으로 리소스를 자동으로 할당할 수 있다. 예를 들어 예측 분석을 사용하여 특정 AI 워크로드에 필요한 리소스를 추정하고 필요한 리소스를 실시간으로 할당할 수 있다.

수동 리소스 할당: 수동 리소스 할당을 사용하여 특정 작업이나 애플리케이션에 리소스를 할당할 수 있다. 예를 들어, 많은 계산 능력이 필요한 우선 순위가 높은 작업에 더 많은 리소스를 할당할 수 있다.

리소스 모니터링: IoT 에지 리소스의 사용을 모니터링하여 이상을 감지하고 리소스 할당을 최적화할 수 있다. 예를 들어 실시간 모니터링을 사용하여 예상보다 많은 리소스를 소비하는 장치를 식별하고 리소스 사용량

을 줄이기 위한 조치를 취할 수 있다.

**리소스 최적화:** 데이터 전송 및 스토리지 요구 사항을 줄이기 위해 압축, 집계 및 필터링과 같은 기술을 사용하여 IoT 에지 리소스를 최적화할 수 있다. 예를 들어, 데이터 압축 기술을 사용하여 센서에서 전송되는 데이터의 크기를 줄일 수 있으며, 이를 통해 네트워크 대역폭 사용량을 줄이고 데이터 전송 속도를 높일 수 있다.

**리소스 확장:** 수요에 따라 IoT 에지 리소스를 확장하거나 축소할 수 있다. 예를 들어 사용량이 가장 많은 기간에는 리소스를 확장하여 높은 수요를 처리하고 사용량이 적은 기간에는 축소하여 리소스 사용량과 비용을 줄일 수 있다.

전반적으로 효과적인 IoT 에지 리소스 관리는 AIoT 에지 컴퓨팅 환경에서 최적의 성능을 보장하고 비용을 최소화하며 리소스 활용도를 개선하는 데 매우 중요하다. 자동화 및 수동 리소스 관리 기술을 조합하여 조직은 IoT 에지 리소스를 최적화하고 AIoT 애플리케이션에서 더 나은 결과를 얻을 수 있다.

#### 4. 평가

IoT Edge 리소스의 무결성을 검증하기 위한 이질적인 계산 하드웨어가 있는 복잡한 환경에서 잘 동작하는 평가하고자 세 가지 다른 테스트 시나리오에서 에너지 효율성을 평가한다.

Scenario 1: 이 시나리오는 IoT 리소스 블록을 단일 블록 체인으로 속도를 높이기 위해서 모든 계산 하드웨어를 완전히 활용할 수 있는지를 평가한다. 단일 블록 체인 처리시 벡터/매트릭스 계산이 많이 필요하기 때문에 CPU는 대기열 지연시간이 높아 GPU에 비해 CPU가 실행하는 데 일반적으로 더 많은 시간이 걸린다. 특히 CPU방법과 비교하여 GPU 전용 방법은 평균 대기 시간이 38.47% 줄이고 처리량은 평균 1.93배 이상 향상되었다. GPU가 IoT Edge 리소스 처리를 가속화할 수 있지만 GPU에 너무 많은 IoT Edge 리소스 작업이 할당되면 대기열의 길이가 증가하고 추론 대기 시간이 급격히 증가하였다. Table 2에서 알 수 있듯이, CPU방식이 처리량이 가장 낮으며, 라운드 로빈 방식은 GPU 전용 방식보다 처리량이 낮다. 이 같은 결과는 런타임에 시스템 상태에 기반한 모든 계산 리소스를 완전히 활용할 수 있는 방법이 요구된다.

**Table 2. Throughput** Units : Tasks/Period

| Method | CPU | GPU | Round-Robin |
|--------|-----|-----|-------------|
| 1      | 984 | 406 | 327         |
| 2      | 876 | 378 | 298         |
| 3      | 928 | 394 | 310         |
| 4      | 903 | 387 | 304         |
| 5      | 951 | 401 | 309         |

Table 3에서 알 수 있듯이 IoT 리소스 블록을 단일 블록 체인으로 처리시 라운드 로빈은 평균 추론 대기 시간을 줄였다. 라운드 로빈 방식은 CPU 전용 방식과 GPU 전용 방식에 비해 평균 추론 대기 시간을 각각 평균 41.09%, 17.37% 줄였다.

**Table 3. Average inference latency** Units : s

| Method | CPU    | GPU   | Round-Robin |
|--------|--------|-------|-------------|
| 1      | 4.133  | 1.137 | 0.635       |
| 2      | 3.859  | 0.768 | 0.493       |
| 3      | 4.5723 | 1.583 | 0.852       |
| 4      | 3.905  | 0.935 | 0.647       |
| 5      | 4.033  | 1.054 | 0.764       |

Table 4에서 알 수 있듯이, 에너지 효율은 CPU 방식 및 라운드 로빈 방식이 작업당 에너지 소비량이 각각 평균 38.21%, 3.31% 감소하였다. 에너지가 보상 함수에서 명시적으로 다루어지기 때문에, 거의 동일한 처리량과 평균 추론 대기 시간이 약간 증가하여 에너지 효율을 달성하였다.

**Table 4. Energy efficiency** Units : Joules/Task

| Method | CPU   | GPU   | Round-Robin |
|--------|-------|-------|-------------|
| 1      | 0.311 | 0.878 | 0.119       |
| 2      | 0.284 | 0.814 | 0.093       |
| 3      | 0.296 | 0.835 | 0.102       |
| 4      | 0.247 | 0.768 | 0.086       |
| 5      | 0.279 | 0.793 | 0.097       |

Scenario 2: 이 시나리오는 CPU 방법, GPU방법, Round-Robin방법 등에서 이기종 계산 리소스를 효율적으로 활용하여 다중 블록 체인을 지원할 수 있는지 평

가한다. IoT 리소스 블록을 다중 블록 체인으로 실행되었을 때 하드웨어의 계산 리소스를 처리한 결과, Table 5~Table 7처럼 GPU방식과 Round-Robin 방식 모두 처리량과 대기 시간 측면에서 모든 기준선보다 훨씬 나은 성능을 제공하였다. 특히 CPU 방식에 비해 평균적으로 GPU 방식과 Round-Robin 방식은 처리량을 각각 1.3배, 1.5배 향상시키고 평균 추론 대기 시간은 24.05%와 32.5% 단축하였다. 에너지 소비는 GPU 방식에 비해 CPU 방식과 Round-Robin 방식은 작업당 평균 43.72, 2.37터 적은 에너지를 소비하였다.

**Table 5. Throughput** Units : Tasks/Period

| Method | CPU | GPU | Round-Robin |
|--------|-----|-----|-------------|
| 1      | 137 | 321 | 349         |
| 2      | 124 | 307 | 318         |
| 3      | 130 | 315 | 328         |
| 4      | 128 | 308 | 321         |
| 5      | 133 | 316 | 337         |

**Table 6. Average inference latency** Units : s

| Method | CPU   | GPU   | Round-Robin |
|--------|-------|-------|-------------|
| 1      | 3.746 | 1.632 | 0.921       |
| 2      | 3.159 | 1.243 | 0.656       |
| 3      | 3.324 | 1.327 | 0.733       |
| 4      | 3.582 | 1.413 | 0.822       |
| 5      | 3.247 | 1.255 | 0.656       |

**Table 7. Energy efficiency** Units : Joules/Task

| Method | CPU   | GPU   | Round-Robin |
|--------|-------|-------|-------------|
| 1      | 0.242 | 0.133 | 0.154       |
| 2      | 0.285 | 0.171 | 0.197       |
| 3      | 0.263 | 0.152 | 0.1734      |
| 4      | 0.277 | 0.166 | 0.182       |
| 5      | 0.259 | 0.144 | 0.169       |

Scenario 3: 이 시나리오는 IoT 리소스만을 인식할 수 있는 추가 리소스 정보를 IoT Voluater에서 추출하여 처리할 때의 성능을 평가한다. IoT Voluater에서 IoT 추가 리소스를 분류하는 데 처리 파이프라인을 구성하여 중복 추론을 방지함으로써 전반적인 효율성을 향상시켰습니다. Table 8~Table 10의 결과처럼 처리

량과 대기 시간 측면에서 평균적으로 CPU 방식보다 GPU방식과 Round-Robin 방식이 각각 11.04%, 9.87% 향상시키고 평균 추론 대기 시간은 각각 24.1%, 20.78% 감소시켰다. 이러 결과는 처리량 및 대기 시간 측면에서 Round-Robin 방식이 우수성하다. Round-Robin 방식의 에너지 사용은 CPU 전용 방법보다 태스크당 평균 24.12% 적은 에너지를 사용한다. 처리량과 대기 시간을 약간 줄임으로써 Round-Robin 방식이 CPU 방식과 GPU 방식보다 효율성이 5.21%와 3.24% 더 향상시켰다. 이러 결과는 여러 상관 관계가 있는 IoT 리소스 블록을 복잡한 다중 블록 체인에 적용할 경우에 Round-Robin 방식이 잘 작동한다는 것을 보여준다.

**Table 8. Throughput** Units : Tasks/Period

| Method | CPU | GPU | Round-Robin |
|--------|-----|-----|-------------|
| 1      | 234 | 389 | 423         |
| 2      | 254 | 401 | 465         |
| 3      | 212 | 357 | 398         |
| 4      | 242 | 395 | 411         |
| 5      | 257 | 408 | 476         |

**Table 9. Average inference latency** Units : s

| Method | CPU   | GPU   | Round-Robin |
|--------|-------|-------|-------------|
| 1      | 2.692 | 1.271 | 0.873       |
| 2      | 2.422 | 1.161 | 0.715       |
| 3      | 2.556 | 1.212 | 0.783       |
| 4      | 2.483 | 1.194 | 0.732       |
| 5      | 2.592 | 1.241 | 0.831       |

**Table 10. Energy efficiency** Units : Joules/Task

| Method | CPU   | GPU   | Round-Robin |
|--------|-------|-------|-------------|
| 1      | 0.115 | 0.095 | 0.078       |
| 2      | 0.151 | 0.119 | 0.096       |
| 3      | 0.147 | 0.106 | 0.089       |
| 4      | 0.092 | 0.072 | 0.066       |
| 5      | 0.132 | 0.103 | 0.082       |

### 5. 결론

최근 IoT 기술이 발전하면서 다양한 분야에서 활용되고 있다. 그러나 IoT 장치가 다양한 분야에서 송수신

되는 정보가 다양해지면서 애플리케이션에서 IoT 리소스 관련 보안 문제가 대두되고 있다. 본 논문에서는 분산 AIoT 환경에서 IoT Edge 리소스 할당을 계층적으로 합성곱신경망(Convolution Neural Network, CNN) 서비스를 지원받을 수 있도록 IoT Edge 리소스를 안전하게 관리할 수 있는 기법을 제안하였다. 제안 기법은 계층적 IoT Edge 리소스의 특징을 계층별로 추출하여 특징들의 공격 패턴을 Convolution과정과 Pooling 과정을 통해 파악한 후 동일 패턴의 리소스를 블록체인으로 묶어 처리하기 때문에 DNN 방식보다 IoT 장치의 리소스 할당을 안전하게 최적화할 수 있었다. 제안 기법은 각 IoT 장치의 리소스를 블록체인 기반으로 연결함으로써 공격패턴을 갖는 IoT 리소스는 이전 공격 패턴의 리소스와 해시 값으로 저장한 후 체인을 연결함으로써 분산 AIoT 환경에서 효과적으로 공격 패턴을 갖는 리소스를 검증하여 신뢰성을 보장받도록 하였다. 향후 연구에서는 기존 연구 결과를 기반으로 다양한 스마트 팜에 농작물 생육 정보 활용 방법을 불규칙적인 대규모 데이터를 처리할 때의 신뢰도 검증 정확도가 일정하게 유지할 수 있는 연구를 지속적으로 수행연구할 계획이다.

## REFERENCES

[1] M. N. Aman, B. Sikdar, K. C. Chua & A. Ali, (2018). Low power data integrity in IoT systems. *IEEE Internet of Things Journal*, 5(4), 3102-3113. DOI : 10.1109/JIOT.2018.2833206

[2] F. Contietal et al. (2017). An IoT endpoint system-on-chip for secure and energy-efficient near-sensor analytics. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(9), 2481-2494. DOI : 10.1109/TCSI.2017.2698019

[3] M. N. Aman, K. C. Chua & B. Sikdar, (2017). Mutual authentication in IoT systems using physical unclonable functions. *IEEE Internet of Things Journal*, 4(5), 1327-1340. DOI : 10.1109/HOST54066.2022.9840132

[4] M. Ghobaei-Arani & A. Shahidinejad. (2022). A cost-efficient IoT service placement approach using whale optimization algorithm in fog computing environment. *Expert Systems with Applications*, 200, 117012. DOI : 10.1016/j.eswa.2022.117012

[5] Y. Li, J. Shen, S. Ji & Y. H. Lai. (2023). *Blockchain-Based Data Integrity Verification Scheme in AIoT Cloud-Edge Computing Environment*. *IEEE Transactions on Engineering Management*. DOI : 10.1109/TEM.2023.3262678

[6] M. Etemadi, M. Ghobaei-Arani & Shahidinejad, A. (2021). A cost-efficient auto-scaling mechanism for IoT applications in fog computing environment: a deep learning-based approach. *Cluster Computing*, 24(4), 3277-3292. DOI : 10.1007/s10586-021-03307-2

[7] F. Farahbakhsh, A. Shahidinejad & M. Ghobaei-Arani, (2021). Multiuser context-aware computation offloading in mobile edge computing based on Bayesian learning automata. *Transactions on Emerging Telecommunications Technologies*, 32(1), e4127. DOI : 10.1002/ett.4127

[8] S. K. Battula, S. Garg, R. Naha, M. B. Amin, B. Kang & E. Aghasian, (2022). A blockchain-based framework for automatic SLA management in fog computing environments. *The Journal of Supercomputing*, 78(15), 16647-16677. DOI : 10.1007/s11227-022-04545-w

[9] M. Tarahomi, M. Izadi & M. Ghobaei-Arani. (2021). An efficient power-aware VM allocation mechanism in cloud data centers: a micro genetic-based approach. *Cluster Computing*, 24, 919-934. DOI : 10.1007/s10586-020-03152-9

[10] M. Salimian, M. Ghobaei-Arani & A. Shahidinejad. (2021). Toward an autonomic approach for Internet of Things service placement using gray wolf optimization in the fog computing environment. *Software: Practice and Experience*, 51(8), 1745-1772.

정윤수(Yoon-Su Jeong)

[정회원]



- 1998년 2월 : 충북대학교 전자계산학과 학사
- 2000년 2월 : 충북대학교 전자계산학과 석사
- 2008년 2월 : 충북대학교 전자계산학과 박사

- 2012년 3월 ~ 현재 : 목원대학교 정보통신공학과 조교수
- 관심분야 : 유·무선 통신 보안, 정보보호, 바이오인포매틱, 헬스케어, 빅 데이터, 클라우드 컴퓨팅
- E-mail : bukmunro@mokwon.ac.kr