

<https://doi.org/10.7236/IIBC.2017.17.4.137>

IIBC 2017-4-17

# 센서 네트워크에서 질의 처리 시스템

## Query Processing Systems in Sensor Networks

김정준\*, 정성택\*

Jeong-Joon Kim\*, Sung-Taek Chung\*

**요약** 최근 IoT 기술의 발전과 더불어 센서 노드, RFID, CCTV, 스마트폰 등에서 다양한 데이터를 Sensing하는 기술과 무선 통신 기술이 급격히 발전하면서 여러 응용 분야에서 센서 네트워크 관련 기술을 활용하기 위한 다양한 연구가 활발히 추진되고 있다. 따라서, GeoSensor 활용이 증가함에 따라 공간 센서 데이터와 같은 2차원 데이터를 효율적으로 처리하기 위한 질의 처리 시스템이 활발히 연구되고 있다. 하지만 기존 질의 처리 시스템들은 시공간 센서 데이터와 같은 다차원 데이터를 처리하기 위한 데이터 타입과 연산자를 지원하지 않기 때문에 이와 같은 다차원 데이터를 처리하기에 미흡하다. 따라서, 본 논문은 이러한 센서 네트워크에서 다차원 데이터를 효율적으로 처리하기 위하여 질의 처리 시스템을 연구 개발하였다.

**Abstract** Recently, along with the development of IoT technology, technologies for wirelessly sensing various data, such as sensor nodes, RFID, CCTV, smart phones, etc., have rapidly developed, and in the field of multiple applications, to utilize sensor network related technology Have been actively pursued in various fields.

Therefore, as GeoSensor utilization increases, query processing systems for efficiently processing 2D data such as spatial sensor data are actively researched. However, existing spatial query processing systems do not support a spatial-temporal data type and a spatial-temporal operator for processing spatial-temporal sensor data. Therefore, they are inadequate for processing spatial-temporal sensor data like GeoSensor. Accordingly, this paper developed a spatial-temporal query processing system, for efficient spatial-temporal query processing of spatial-temporal sensor data in a sensor network.

**Key Words** : Sensor Networks, Sensor Query Processing, Data Type, Operator

### 1. 서 론

최근 주변의 모든 사물에 센서를 부착하고 인터넷과 연결하여 다양한 정보를 관리 및 서비스할 수 있는 IoT(Internet of Things) 기술의 발전과 더불어 센서 노드, RFID, CCTV, 스마트폰 등에서 다양한 데이터를 Sensing하는 기술과 와이파이(Wi-Fi), 블루투스(Bluetooth), 지그비(ZigBee) 등과 같은 무선 통신 기술이 급격히 발전하면서 군사, 의료, 기상, 환경, 교통, 가정, 회사 등과 같은 여러 응용 분야에서 센서 네트워크(Sensor

Networks) 관련 기술을 활용하기 위한 다양한 연구가 활발히 추진되고 있다<sup>[1,2]</sup>.

센서 네트워크에서는 다수의 센서 노드들로 통신망이 구성된다. 센서 노드는 센싱(sensing), 프로세싱(processing), 저장(storage), 통신(communicating) 모듈로 구성되며, 각 센서 노드는 하드웨어와 소프트웨어가 아주 제한적이다. 무엇보다 에너지 용량이 작고 데이터 전송/대기로 인한 전력 소비가 심각하다. 그래서 센서 네트워크에서는 이러한 센서 노드의 특성을 고려하여 낮은 컴퓨팅 능력과 작은 전원 용량을 갖는 장치에 다양한 질

\*정회원, 한국산업기술대학교 컴퓨터공학과

\*정회원, 한국산업기술대학교 컴퓨터공학과

접수일자: 2017년 6월 2일, 수정완료: 2017년 7월 2일

게재확정일자: 2017년 8월 11일

Received: 2 June, 2017 / Revised: 2 July, 2017

Accepted: 11 August, 2017

\*Corresponding Author: unitaek@kpu.ac.kr

Dept. of Computer Eng., Korea Polytechnic University, Korea

의를 보다 효율적으로 처리하기 위한 연구가 진행되고 있다<sup>[5,6]</sup>.

따라서, 본 논문에서는 이러한 센서 네트워크에서 다차원 데이터에 대해 질의 처리를 효율적으로 제공하기 위한 STTinyDB(Spatio-Temporal TinyDB)를 개발하였다. 본 논문에서 개발하는 STTinyDB는 다차원 센서 데이터를 처리하기 위해 대표적인 센서 질의 처리 시스템인 TinyDB를 확장하여 개발하며, 상호운용성을 위해서 국제공간정보표준화기구 OGC(Open Geospatial Consortium)에서 제시한 “Simple Features Specification for SQL” 표준을 확장하여 구현한다<sup>[3,4]</sup>. 본 논문의 STTinyDB에 대한 전체 시스템 구성은 인터페이스 관리자, 데이터 통신 관리자, 다차원 데이터 관리자, 다차원 연산 관리자, 다차원 데이터 스트림 관리자로 이루어져 있다.

## II. 관련 연구

### 1. 센서 운영체제

일반적인 센서 노드는 마이크로 컨트롤러(Micro Controller)를 내장한 소형 컴퓨터 시스템으로 센싱 응용 처리와 노드 간 통신 등을 위한 운영체제(Operating Systems)를 필수적으로 요구한다. 센서 운영체제는 자원이 제한적인 센서 노드에서 수행되어야 하기 때문에 크기가 작고 전력 소비가 적어야 하며, 센서 노드 간에 저전력 통신을 제공하면서도 프로세서와 메모리를 효율적으로 관리하여야 한다<sup>[7,8]</sup>.

센서 운영체제로는 Berkeley 대학의 TinyOS, SICS의 Contiki OS, Colorado 대학의 MANTIS OS, 한국전자통신연구원의 NanoQPlus 등이 있다.

현재 가장 널리 사용되고 있는 대표적인 센서 운영체제인 TinyOS는 미국 버클리(Berkeley)대학에서 nesC(Network Embedded System C) 언어로 개발한 오픈 소스 시스템이며, 2012년 8월에 나온 2.1.2 버전까지 공개되어 있다.

TinyOS는 컴포넌트 단위로 모듈화를 하여 시스템을 구성한다. 컴포넌트는 몇몇 상태와 태스크로 구성된다. 각 컴포넌트는 다른 컴포넌트들과 인터페이스라는 함수 호출을 통해 동작을 연결시키며, 인터페이스를 통해 컴포넌트는 외부와 연결된다. 인터페이스는 커맨드와 이벤

트로 분류되며, 커맨드는 어떤 동작에 대한 호출이고, 이벤트는 호출된 어떤 동작이나 환경변화에 의해 실행되는 것을 의미한다. 이벤트의 반환 값으로는 처음 호출한 동작의 성공 여부를 알 수 있으며, 특정 이벤트들은 하드웨어 인터럽트에 직접 연결되어 동작한다. 컴포넌트들은 어플리케이션 개발자에 의해 라이브러리로 사용되며, 인터페이스들을 사용하여 컴포넌트들을 서로 연결해서 어플리케이션을 만든다. 그림 1은 TinyOS의 구조도를 보여준다.

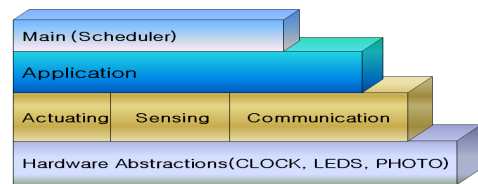


그림 1. TinyOS의 구조도

Fig. 1. Architecture of TinyOS

그림 1에서 보는 바와 같이 운영체제 부분과 응용 어플리케이션 부분이 함께 컴파일 되어 하드웨어 플랫폼에 다운로드 된다. 어플리케이션 부분은 사용자가 직접 만들게 되며, 나머지 부분은 기존 라이브러리를 재사용하게 된다.

### 2. 센서 데이터베이스 시스템

센서로 환경과 사물의 변화를 그대로 수집한다면 문제가 발생할 수 있다. 끊임없이 수집되는 방대한 양의 데이터로 인해 무선으로 전송하는 패킷의 크기가 커지고, 수집된 데이터에서 원하는 정보를 추출하는 데에도 비용이 발생하기 때문이다. 이러한 문제를 해결하기 위해서 센서 데이터베이스 시스템이 연구되었다.

센서 데이터베이스 시스템은 센서 네트워크에 흩어져 있는 센싱 정보들에 대하여 센서 노드들의 제한된 리소스들(저성능의 CPU, 작은 용량의 메모리, 저용량의 통신 대역폭)을 고려하면서 사용자 질의를 효율적으로 처리해주는 소프트웨어이다.

TinyDB는 TinyOS가 무선 센서 네트워크로부터 정보를 추출하기 위한 질의 처리 시스템이며, 최신 버전은 2004년 7월에 나온 1.1.3이다. TinyDB의 특징은 첫 번째, 센서 네트워크에서 수집된 센서의 종류를 기술하기 위해 메타 데이터 카탈로그를 제공한다. 두 번째, 사용자가 다

루기 쉽게 원하는 데이터를 기술하는 질의 언어를 사용한다. 세 번째, 네트워크 망을 가지고 있다. 네 번째, 같이 묶여 있는 센서 노드들을 동시에 실행할 때 다중 질의를 할당한다. 다섯 번째, 센서 네트워크를 확장하기 위해서 단순히 새로운 센서 노드에 표준 TinyDB 코드를 다운로드하고 리셋만 하면 된다. 그림 2 은 TinyDB의 질의 처리 과정을 보여준다.

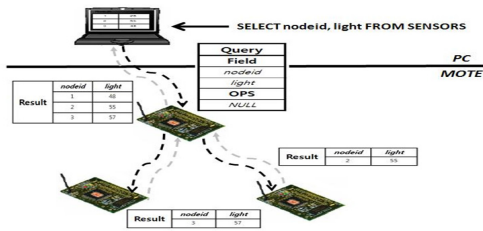


그림 2. TinyDB의 질의 처리  
 Fig. 2. Query Processing of TinyDB

그림 2 에서 보는 바와 같이 TinyDB는 센서 네트워크에 있는 센서 노드들을 하나의 테이블로 간주한다. 질의 처리 과정은 PC에서 입력 받은 질의를 최적화된 네트워크 경로에 따라 MOTE에 전송되며, 전송된 질의는 라우팅 트리를 기반으로 이웃 센서 노드의 데이터를 습득하고 집계 연산을 하여 PC에 결과를 전송한다.

### III. 시스템 설계

본 논문에서 개발한 STTinyDB는 센서 네트워크에서 군사, 의료, 기상, 환경, 교통, 가정, 회사 등과 관련된 다양한 서비스를 제공하기 위해서 TinyDB를 확장하여 다차원 센서 데이터를 효율적으로 처리하는 질의 처리 시스템이다. 시공간 질의 입력 모듈은 사용자가 Java GUI로 만들어진 인터페이스를 통해서 센서/시공간 속성과 연산자를 선택하여 시공간 질의를 생성하는 기능을 제공한다. 또한 질의 수행 주기, 필터링 조건, 질의 ID 등을 설정하는 기능과 입력 받은 시공간 질의를 시공간 질의 분석 모듈로 전달하는 기능을 제공한다. 표 1은 STTinyDB에서 지원하는 시공간 질의 구문 형식과 질의 예를 보여준다.

표 1. 질의 구문 형식과 질의 예  
 Table 1. Query Type and Example

Query Grammar	
SELECT select-list [FROM sensors] WHERE where-clause [EPOCH DURATION]	
Query	Example
Time Relation Operator	SELECT nodeid, time, temp FROM sensors WHERE tDisjoint(tPeriod(2015/05/01 12:00:00, 2015/05/01 13:00:00), time)
Time Analysis Operator	SELECT nodeid, time, temp FROM sensors WHERE tContains(tUnion(tPeriod(2015/05/01 12:00:00, 2015/05/01 14:00:00), tPeriod(2015/05/01 13:00:00, 2015/05/01 15:00:00)), time)
Spatio-temporal Relation Operator	SELECT nodeid, temp, stLoc FROM sensors WHERE stDisjoint(stPolygon(tPeriod(2015/05/01 14:00:00, 2015/05/01 15:00:00), 0 0, 0 700, 700 700, 700 0, 0 0), stLoc)
Spatio-temporal Analysis Operator	SELECT nodeid, temp, stLoc FROM sensors WHERE stContains(stIntersection(stPolygon(tPeriod(2015/05/01 12:00:00, 2015/05/01 18:00:00), 0 0, 0 400, 400 400, 400 0, 0 0), stPolygon(tPeriod(2015/05/01 16:00:00, 2015/05/01 19:00:00), 100 100, 100 500, 500 500, 100 100)), stLoc)
Spatio-temporal Trajectory Operator	SELECT nodeid, temp, stLoc FROM sensors WHERE stEnters(stPolygon(tPeriod(2015/05/01 12:00:00, 2015/05/01 23:00:00), 0 0, 0 100, 100 100, 100 0, 0 0), stLoc)

표 1 에서 보는 바와 같이 시간 관계/분석 연산자와 시공간 관계/분석/궤적 연산자를 이용하여 다양한 시간 및 시공간 질의가 가능하다.

시공간 질의 분석 모듈은 시공간 질의 입력 모듈로부터 입력 받은 시간 및 시공간 질의를 분석하는 기능을 제공하고, 분석된 정보를 바탕으로 질의의 오류 및 유효성을 검사하는 기능을 제공한다. 그림 3은 입력된 질의를 분석하는 과정을 보여준다.

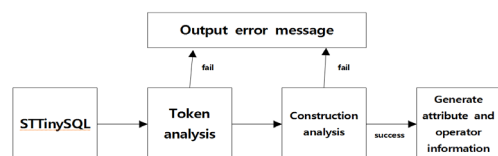


그림 3. 질의 분석 과정  
 Fig. 3. Analysis Processing of Query

그림 3과 같이 STTinySQL(Spatio-Temporal TinyDB SQL) 질의를 입력 받으면 토큰 분석이 이루어지는데, 이는 파서에 정의된 예약어와 비교하여 확인하고, 오류가 발견되지 않았다면 각각 개별적인 토큰(tokens)으로 분리해 저장한다. 그다음 구문 분석에서는 질의가 올바르게 구성되어 있는지를 파서에 정의된 규칙들을 통해 확인한다. 만약 분석 과정 중 오류가 발생되어 실패되었다면 사용자에게 오류 메시지를 출력한다. 이와 같은 분석 과정을 통하여 확인된 질의의 속성(select-list)과 연산자(where-clause) 등의 정보를 생성한다.

세션 관리 모듈은 사용자가 입력한 시공간 질의 영역에 질의하기 위해서 해당 질의에 만족하는 센서 노드들의 세션을 생성, 유지, 삭제하는 기능을 제공한다. 질의 영역에 센서 노드가 해당되면 세션을 생성하고, 질의가 끝날 때까지 세션을 유지하며, 질의가 종료되면 세션을 삭제한다. 동시에 여러 개의 질의가 발생한 경우에는 시스템에 의해 부여된 고유한 질의 ID를 통해서 개별적으로 질의를 구별하여 세션을 관리한다.

세션을 생성하기 위해서 센서 네트워크 망의 센서 노드에 접속 요청을 하는데 지연 시간이 지나도 접속을 하지 않는다면 접속 요청을 취소하고, 접속이 잘 되었는데 설정된 유효 시간 동안 일련의 작업이 수행되지 않는다면 세션의 연결을 종료시킨다.

## IV. 시스템 구현

본 논문에서 STTinyDB를 구현하기 위해서 센서 운영체제는 TinyOS 1.1.15, 센서 데이터베이스 시스템은 TinyDB 1.1.3 기반으로 실행하였고, 프로그래밍 언어는 TinyOS에서 제공하는 nesC 1.2.8과 g++ 3.4.3을 사용하였다. 관리 툴은 Windows XP 기반으로 Java 1.4를 사용하였다.

사용자로부터 시공간 질의를 입력 받는 기능, 질의를 분석하는 기능, 질의에 대한 결과 화면을 보여주는 기능을 구현하기 위하여 TinyDB에서 제공하는 질의 입력 GUI, 시공간 질의를 분석하기 위한 토큰/구문 분석기, 질의 결과 GUI를 확장하였다. 그림 4는 질의 입력 GUI에 대한 구현 화면을 보여준다.

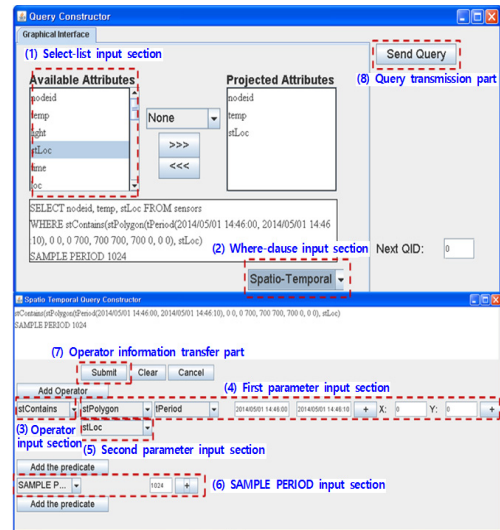


그림 4. 실행 화면  
Fig. 4. Execution Screen

그림 4는 시공간 질의 입력 모듈이 동작하는 과정을 확인하기 위해서 질의 입력 GUI를 통해 확인하였다. 질의 입력 GUI는 select-list 입력 부분, where-clause 입력 부분, 연산자 입력 부분, 연산자의 첫 번째 파라미터 입력 부분, 연산자의 두 번째 파라미터 입력 부분, SAMPLE PERIOD 입력 부분, 연산자 정보 전송 부분, 질의 전송 부분으로 구성되어 있다. (1) select-list 입력 부분은 센서/시공간 등의 속성을 입력하는 부분이다. (2) where-clause 입력 부분은 시공간 연산자를 구성하는 부분이다. (3) 연산자 입력 부분은 질의에서 사용할 연산자를 입력하는 부분이고, (4) 연산자의 첫 번째 파라미터 입력 부분은 시간 및 시공간 데이터 타입을 입력하는 부분이다. (5) 연산자의 두 번째 파라미터 입력 부분은 시공간 속성을 선택하는 부분이다. (6) SAMPLE PERIOD 입력 부분은 질의 수행 주기를 설정하는 부분이며, 시간 단위는 밀리세컨드(ms)이다. (7) 연산자 정보 전송 부분은 (3) ~ (6) 과정을 통하여 where-clause이 구성되었다면 이를 질의 입력 GUI에 전송하는 부분이다. (8) 질의 전송 부분은 (1) ~ (7)까지 수행하여 만들어진 시공간 질의를 센서 네트워크 망에 있는 센서 노드에 전송하는 부분이다.

사용자가 입력한 시공간 질의를 센서 노드에서 효율적으로 처리하기 위하여 시간 관계/분석 연산자, 시공간 관계/분석/계적 연산자를 구현하였다.

아래는 시간 관계 연산자 중 tRightDisjoint에 대한 연

산을 수행하기 위한 질의 예를 보여준다.

```
SELECT nodeid, time, temp
FROM sensors
WHERE tRightDisjoint(tPeriod(2014/05/01 21:18:00, 2014/05/01 21:18:15), time)
SAMPLE PERIOD 1024
```

위에서 보는 바와 같이 tRightDisjoint 질의는 특정 시간 구간(tPeriod(2014/05/01 21:18:00, 2014/05/01 21:18:15)) 이후에 포함되지 않는 센서 노드 아이디(nodeid), 시간(time), 온도(temp)를 반환한다. 그림 5는 tRightDisjoint의 결과 화면을 보여준다.

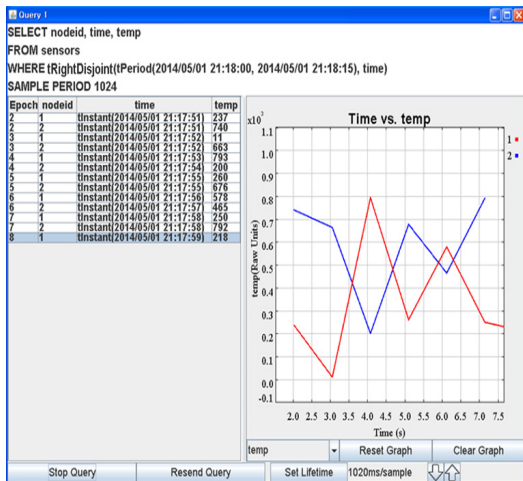


그림 5. 질의 실행 화면  
 Fig. 5. Query Execution Screen

그림 5에서 보는 바와 같이 센서 노드 1(nodeid=1, epoch=2), 센서 노드 2(nodeid=2, epoch=2)는 특정 시간 구간(tPeriod(2014/05/01 21:18:00, 2014/05/01 21:18:15)) 이후에 포함되지 않기 때문에 해당 센서 노드 아이디, 시간, 온도 정보를 텍스트와 그래프 형식으로 나타낸다.

## V. 결론

최근 IoT 기술의 발전과 더불어 다양한 센서 데이터를 수집하는 기술과 무선 통신 기술이 급격히 발전함에 따라 센서 네트워크 관련 기술에 대한 관심과 연구가 증대되고 있다.

센서 네트워크에서 2차원 공간 센서 데이터에 대한 효율적인 질의 처리를 위한 여러 가지의 공간 질의 처리 시스템이 연구되었다. 그러나, 기존 공간 질의 처리 시스템들은 다차원 센서 데이터 처리를 위한 데이터 타입과 연산자를 지원하지 않기 때문에 다차원 센서 데이터를 처리하는 데 미흡했다.

따라서, 이러한 문제를 해결하기 위해서 본 논문에서는 OGC의 표준 명세 중 “Simple Features Specification for SQL”을 확장하여 다차원 시공간 질의 처리 시스템을 개발하였다. 본 시스템은 다차원 데이터 타입과 시공간 관계/분석 연산자를 제공하고, 이동 센서의 궤적을 처리하기 위해서 궤적 연산자도 제공한다. 또한, 본 시스템에서 사용되는 다양한 다차원 시공간 질의들 간 필요한 데이터를 공유하여 데이터 스트림으로 인해 발생하는 시스템 부하를 줄이도록 메모리 공유 기능을 제공하였으며, 처리하지 못할 많은 입력 데이터 스트림이 들어왔을 경우 최소한의 정확도 손실로 입력 데이터 스트림 양을 최대한 줄여 과부하가 발생하지 않도록 필터링하는 기능을 제공하였다.

마지막으로, 본 논문에서 개발한 STTinyDB를 생체 모니터링 시나리오에 적용하여 검증함으로써 본 시스템이 센서 네트워크에서 다차원 시공간 질의 처리가 필요한 많은 응용 분야에 유용하게 사용될 수 있음을 보였다.

## References

- [1] Ale, R., “Working with Spatio-Temporal Data Type,” Proc. of the Int. Conf. on Society for Photogrammetry and Remote Sensing, 2012, pp.5-10.
- [2] Kim, D. O., Liu, L., Shin, I. S., Kim, J. J., and Han, K. J., “Spatial TinyDB: A Spatial Sensor Database System for the USN Environment,” Journal of Distributed Sensor Networks, 2013.
- [3] Open Geospatial Consortium, “OpenGIS Implementation Specification for Geographic Information-Simple Feature Access-Part 1:Common Architecture,” Version 1.2.1, 2011.
- [4] Open Geospatial Consortium, “OpenGIS Implementation Specification for Geographic

Information-Simple Feature Access-Part 2:SQL Option," Version 1.2.1, 2011.

- [5] Park, J., Kim, K., Ahn, S., and Hong, B., "Continuous Query Processing on Data Stream: Sensor, Location and Identification," Proc. of Seventh Int. Conf. on Information Technology, 2010, pp.518-522.
- [6] Hyeun Kim, Do-Hyeun Kim, Hee-Do Park, "A Tracking Service of Animal Situation using RFID, GPS, and Sensor," Journal of IIBC, 2009, Vol. 9, No. 5, pp.79-84.
- [7] Cho H., S., & Cho J., P., "The Development of New dynamic WRR Algorithm for Wireless Sensor Networks," Journal of IIBC, 2010, Vol. 10, No. 5, pp.293-298.
- [8] Im S., J & Hwang H., J., "Design and Development of Simulation Framework for Processing Window Query in Wireless Spatial Data Broadcasting Environment," Journal of IIBC, 2014 Vol. 14, No. 5, pp.173-178.

**정 성 택(정회원)**



- 1992년 : KAIST 전기 및 전자공학과 학사
- 1995년 : KAIST 정보 및 통신공학과 석사
- 2000년 : KAIST 전기 및 전자공학과 박사
- 2004년 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 교수

<주관심분야 : 헬스케어, 기능성 게임, HCI, etc.>

**저자 소개**

**김 정 준(정회원)**



- 2003년 2월 : 건국대학교 컴퓨터공학과 학사
  - 2005년 2월 : 건국대학교 컴퓨터공학과 석사
  - 2010년 8월 : 건국대학교 컴퓨터공학과 박사
  - 2010년 9월 ~ 2012년 8월 : 건국대학교 컴퓨터공학과 강의교수
  - 2012년 9월 ~ 2016년 2월 : 건국대학교 컴퓨터공학과 조교수
  - 2016년 3월 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 조교수
- <주관심분야 : Database Systems, BigData, Semantic Web, Geographic Information Systems (GIS) and Ubiquitous Sensor Network (USN), etc.>