

A HIGHLY ACCURATE DEEP LEARNING BASED  
APPROACH FOR DEVELOPING WIRELESS SENSOR  
NETWORK MIDDLEWARE

Remah Alshinina

Under the Supervision of Dr. Khaled Elleithy

DISSERTATION  
SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE  
AND ENGINEERING  
THE SCHOOL OF ENGINEERING  
UNIVERSITY OF BRIDGEPORT  
CONNECTICUT  
September, 2018




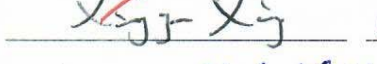

A HIGHLY ACCURATE DEEP LEARNING BASED  
APPROACH FOR DEVELOPING WIRELESS SENSOR  
NETWORK MIDDLEWARE

Remah Alshinina

Under the Supervision of Dr. Khaled Elleithy

**Approvals**

**Committee Members**

Name	Signature	Date
Dr. Khaled M. Elleithy		4/23/18
Dr. Miad Faezipour		4/23/18
Dr. Navarun Gupta		4/23/18
Dr. Xingguo Xiong		04/23/18
Dr. Eman Abdelfattah		4/23/18

**Ph.D. Program Coordinator**

Dr. Khaled M. Elleithy  8/28/18

**Chairman, Computer Science and Engineering Department**

Dr. Ausif Mahmood 

**Dean, School of Engineering**

Dr. Tarek M. Sobh  8/29/18

A HIGHLY ACCURATE DEEP LEARNING BASED  
APPROACH FOR DEVELOPING WIRELESS SENSOR  
NETWORK MIDDLEWARE

© Copyright by Remah Alshinina 2018

## **ABSTRACT**

Despite the popularity of wireless sensor networks (WSNs) in a wide range of applications, the security problems associated with WSNs have not been completely resolved. Since these applications deal with the transfer of sensitive data, protection from various attacks and intrusions is essential. From the current literature, we observed that existing security algorithms are not suitable for large-scale WSNs due to limitations in energy consumption, throughput, and overhead. Middleware is generally introduced as an intermediate layer between WSNs and the end user to address security challenges. However, literature suggests that most existing middleware only cater to intrusions and malicious attacks at the application level rather than during data transmission. This results in loss of nodes during data transmission, increased energy consumption, and increased overhead.

In this research, we introduce an intelligent middleware based on an unsupervised learning technique called the Generative Adversarial Networks (GANs) algorithm. GANs contain two networks: a generator (G) network and a discriminator (D) network. The G network generates fake data that is identical to the data from the sensor nodes; it combines fake and real data to confuse the adversary and stop them from differentiating between the two. This technique completely eliminates the need for fake sensor nodes, which consume more power and reduce both throughput and the lifetime of the network.

The D network contains multiple layers that have the ability to differentiate between real and fake data. The output intended for this algorithm shows an actual interpretation of the data that is securely communicated through the WSN.

The framework is implemented in Python with experiments performed using Keras. The results illustrate that the suggested algorithm not only improves the accuracy of the data but also enhances its security by protecting it from attacks. Data transmission from the WSN to the end user then becomes much more secure and accurate compared to conventional techniques. Simulation results show that the proposed technique provides higher throughput and increases successful data rates while keeping the energy consumption low.

## **ACKNOWLEDGEMENTS**

I would like to first thank the Almighty God who has helped me all the way to complete this work successfully. Without his help and willingness, this work would not have been possible.

I am honored that my work has been supervised by Professor Khaled Elleithy. I would like to thank him for his sincere help and guidance throughout the course of this work. His invaluable suggestions and support at every step made this work achievable for me. I would like to extend my gratitude to the research committee members Dr. Miad Dr. Miad Faezipour, Dr. Navarun Gupta, Dr. Xingguo Xiong, and Dr. Eman Abdlefattah for their willingness to be on the committee, their expert advice and evaluation of the work. I have learned a great deal from each one of the committee members.

I owe a debt of gratitude to my family for their love, understanding and encouragement. The prayers of my parents and love and support of my siblings allowed me to work tirelessly on this research. Last but certainly not the least, I thank my fiancé Dr. Omar Memon for his unconditional love and support throughout my tenure at the University of Bridgeport.

# TABLE OF CONTENTS

ABSTRACT.....	iv
ACKNOWLEDGEMENTS.....	vi
TABLE OF CONTENTS.....	vii
LIST OF TABLES.....	x
LIST OF FIGURES.....	xii
CHAPTER 1: INTRODUCTION.....	1
1.1 Research Problem and Scope.....	1
1.2 Motivation.....	4
1.3 Research Contributions.....	5
CHAPTER 2: BACKGROUND.....	7
2.1 Middleware Challenges for WSNs.....	7
2.1.1 Scalability and Network Topology.....	7
2.1.2 Security.....	7
2.1.3 Quality of Service (QoS).....	9
2.1.4 Fault Tolerance.....	10
2.1.5 Heterogeneity and Data Aggregation.....	11
2.1.6 The Taxonomy of Middleware Architectures for WSNs.....	11
2.1.7 Database Approach.....	12
2.1.8 Virtual Machine Approach.....	13
2.1.9 Message-Oriented Approach.....	13
2.1.10 Modular Approach.....	13
2.1.11 Application Driven Approach.....	13
2.1.12 Service-Oriented Architecture Approach.....	14
CHAPTER 3 LITERATURE SURVEY.....	15
3.1 State of the Art Middleware Approaches for WSNs.....	15
3.1.1 Eagilla.....	16
3.1.2 USEME Approach.....	17

3.1.3	SOMDM Approach.....	19
3.1.4	Mobile Web Service Approach.....	20
3.1.5	MiSense Approach.....	21
3.1.6	Sensors Middleware Approach.....	22
3.1.7	OASiS Approach.....	24
3.1.8	SOMM Approach.....	24
3.1.9	TinySOA Approach.....	26
3.1.10	ESOA Approach.....	27
3.1.11	HealthCare Approaches.....	28
3.1.12	Other Middleware Approaches for WSNs.....	29
3.2	Service-Oriented Architecture Approaches for WSNs.....	35
3.2.1	Healthcare Approaches.....	36
3.2.2	Service-Oriented Device for Smart Environments.....	37
3.2.3	Network Discovery and Selection Approach.....	38
3.2.4	Open Geospatial Consortium Approach.....	39
3.2.5	WSN Cloud User Interaction.....	41
3.2.6	Other Approaches.....	42
3.3	Service Composition for WSNs.....	47
3.3.1	Service Composition with Persistent Queries (SCPQ).....	48
3.3.2	Service Centric Wireless Sensors Networks (SWSNs).....	49
CHAPTER 4: LIMITATIONS OF TRADITIONAL MIDDLEWARE.....		51
4.1	Middleware Approaches for WSNs.....	51
4.2	Service-Oriented Architectures for WSNs.....	56
4.3	Service Composition Architectures for WSNs.....	60
4.4	Discussion.....	62



4.5	Summary .....	67
CHAPTER 5: SWSNM DESIGN AND IMPLEMENTATION OF THE SWSNM .....		69
5.1	Generative Adversarial Networks .....	73
5.2	Generator Network .....	75
5.3	Discriminator Network .....	77
5.4	Dataset .....	79
CHAPTER 6: EXPERIMENTAL RESULTS .....		81
6.1	Experimental Setting .....	81
6.2	Convolutional Neural Networks (CNNs) .....	82
6.3	Confusion Matrix .....	83
6.3.1	Full Feature of NSL-KDD Dataset .....	84
6.3.2	Features Selection .....	88
6.4	Data Visualization .....	89
6.5	Refeeding the Generated Data .....	92
6.6	Evaluation of the SWSNM .....	93
CHAPTER 7: EFFICIENT GAN BASED SWSNM .....		97
7.1	Network Model .....	97
7.2	Generating Fake Data .....	98
7.3	Simulation Test for Fake Data and SWSNM .....	101
7.3.1	Power Consumption .....	102
7.3.2	Throughput .....	104
7.3.3	End-to-End Delay .....	104
CONCLUSIONS .....		108
REFERENCES .....		110

## LIST OF TABLES

Table 2.1. Comparison of different middleware architectures approaches. ....	12
Table 4.1 Comparative Analysis of Middleware Architectures for WSNs. ....	52
Table 4.2. Advantages and Disadvantages of Middleware Architectures for WSNs. .....	54
Table 4.3. The Requirements and Benefits of Using Middleware Architectures for WSNs. ....	55
Table 4.4. Comparative Analysis of Service-Oriented Architectures for WSNs.	57
Table 4.5. Advantages and Disadvantages of SOA for WSNs. ....	58
Table 4.6. The Requirements and Benefits of Applied SOA for WSNs.....	59
Table 4.7. Analysis of Service Composition Architectures for WSNs.....	61
Table 5.1. Algorithm for proposed WSNM based on GANs.....	72
Table 5.2. Overview of NSL-KDD Dataset.....	80
Table 6.1. Accuracy Comparison for Different Layers of CNN Architectures ....	83
Table 6.2. Comparison of Proposed Approach with Different Approaches.....	86
Table 6.3. The Comparisons of Accuracy Rate for Proposed Approach with Existing Approaches on NSL-KDD Dataset.....	87
Table 6.4. Comparison Results between Proposed G Network with Original Dataset (NSL-KDD) with Only 20 Features.....	89
Table 6.5. Comparison of Accuracy Rate of SWSNM with other ML method with 20 Features.....	89

Table 7.1 Comparison Table of Proposed SWSNM Approach with and without Malicious Nodes. ....	107
Table 7.2. Comparison Proposed SWSNM and Eagilla Approaches .....	107

## LIST OF FIGURES

Figure 2.1. The classification of middleware architectures for WSNs. ....	12
Figure 3.1. Eagilla Middleware Approach.....	17
Figure 3.2. USEME Architecture.....	19
Figure 3.3. MiSense Architecture. ....	22
Figure 3.4. SOMM Architecture in the Server Node.....	26
Figure 3.5. TinySOA Approach.....	27
Figure 3.6. ESOA Approach.....	28
Figure 3.7. (a) EWSN Sensor-based Architecture and (b) SWSN Dynamic Service Platform.....	50
Figure 4.1. Generic Security SOM Architecture Framework.....	66
Figure 5.1. The proposed framework for GANs illustrates the sample flow from the generator network (G) to the discriminator network (D).....	74
Figure 5.2. Two models which are learned during the training process for a GAN are the discriminator (D) and the generator (G).....	74
Figure 5.3. The Generator Network Architecture. ....	76
Figure 5.4. The Discriminator Network’s Architecture.....	79
Figure 6.1. (a) Generated data in the proposed Generator Network. (b) Original Dataset (NSL-KDD). ....	85
Figure 6.2. (a) Generated data from G Network with 20 features. (b) Original Dataset (NSL-KDD) with 20 features.....	88

Figure 6.3. t-SNE Visualization with full features. (a) Original Dataset (NSL-KDD) and (b) Generated data in proposed SWSNM.....	91
Figure 6.4. t-SNE Visualization with 20 Features. (a) NSL-KDD Dataset and (b) Generated data in proposed SWSNM.....	92
Figure 6.5. Re-feeding the Generated Data into Generator Network .....	93
Figure 6.6. Generate Accurate Data Scenario and Detecting Errors for each Iteration.....	95
Figure 7.1. The Scenario of Proposed SWSNM Approach .....	99
Figure 7.2. Diagram of SWSNM based on GANs.....	100
Figure 7.3. Energy consumption for SWSNM with Eagilla Approach .....	103
Figure 7.4. Throughput for SWSNM and Eagilla.....	104
Figure 7.5. End-to-End Delay for SWSNM and Eagilla.....	106

# CHAPTER 1: INTRODUCTION

## 1.1 Research Problem and Scope

In the last decade, wireless sensor networks (WSNs) have been applied in monitoring systems that are capable of controlling and supervising various indoor premises, agricultural lands, and forest monitoring applications [1]. The foremost issues associated with WSNs are related to network security due to an increase in the usage of these devices. Traditional security algorithms in WSNs have achieved security goals such as base station protection [2], cryptography [3], attack detections [4], and security location and routing [5-7]. Many researchers have developed solutions to address WSNs' security challenges. The Intrusion Detection System (IDS) is a security management system that monitors all events within a network. IDS is capable of detecting attacks without compromising network security. The anomaly detection types of Intrusion Detection (ID) can detect any abnormal behavior in online data. Misuse detection is another type of ID, which works on offline data and is able to detect known attacks [8, 9]. These sensors introduce massive data for processing and transmission to the base station. Standard security algorithms are not suitable for WSNs due to limitations in power consumption, and communication, low memory (storage capacity), and resource constraints in sensors [10, 11].

The communication and exchange of information between sensors is a critical challenge due to energy consumption constraints in the network. This information must be

protected against various threats [12, 13]. The networks should be secured by support security properties such as confidentiality, authenticity, availability, and integrity. The authors in [12] applied cryptographic algorithms such as signature and encryption/decryption. However, these mechanisms used secret keys that are unsuitable to large-scale WSNs due to the large memory requirement to store these keys [12]. Most of these sensors lack physical protection, which leads to compromised nodes. Compromising one or more of the nodes in a network allows the adversary to launch different attacks to disrupt inter-network communication [14]. There are various attacks such as adversary, compromised node(s), eavesdropper, etc. [15]. These attacks are capable of dropping packets or modifying them, resulting in an impact in the performance of the WSN. Source location privacy (SLPs) is a mechanism that protects sensor data from attacks by generating fake nodes. The fake node and packets (dummy message) create a fake identity and packets without mentioning the source or destination identity. The drawback of this technique is that it requires more energy and overhead [14, 15].

Secure communication between WSNs has been a challenge in recent years [16]. WSNs produce massive data through their low-capacity sensors, which results in the loss of important information during transmission. In addition, sensor nodes have several limitations such as security, data aggregation, power consumption, and the heterogeneity of the sensors' networks. Previous research has shown that using middleware as an intermediate layer between WSNs and the end user provides a solution to the previously mentioned limitations. The middleware provides a bridging platform between the applications and the hardware components of WSNs. The middleware controls the sensor

data nodes while providing them temporary storage [17]. The ability to synchronize newer nodes with the existing nodes allows the middleware to be more efficient while providing support to various resources. This allows minimum or no disturbance in the network's performance if changes occur to the network [18]. Since the data sent over the wireless networks is sensitive, it is prone to unwanted intrusions. Security parameters, such as resource distribution and resource management, enable secure communication within WSNs. End-to-end security auditing can also be enabled to achieve secure communication between nodes [19].

Recently, middleware has been integrated into WSNs to address some of the aforementioned challenges. In [18], the authors reviewed and discussed various middleware approaches such as SOMM, USEME, ESOA, and MiSense. The loss of data during transmission to and from the middleware is still prone to attacks. Alshinina and Elleithy [18] showed a comprehensive, systematic study of the most recent research on WSNs' middleware; they compared existing efficient system designs, addressed the most significant challenges, and made several distinguished contributions within security, data aggregation, message exchange, and quality of service. The authors concluded that a middleware has to be both scalable to dynamic resources and secure at the same time. It was also hypothesized that synchronizing newer nodes with the existing nodes would allow the middleware to perform more efficiently while providing support to various resources. Most middleware approaches lack the security mechanism to secure the network and sensitive data from malicious attacks.



This work focuses on a new unsupervised learning algorithm and how it can be applied to provide a secure wireless sensor network middleware called SWSNM. This framework produces fake data to confuse the attacker and is capable of secure collecting and transmitting data securely between the sensor nodes and the base station compared to other approaches with or without middleware. This technique completely eliminates the need for fake sensor nodes, which consume more power and reduce both throughput and the lifetime of the network. The results show that the proposed approach provides higher throughput and increases successful data rates with low energy consumption.

## **1.2 Motivation**

Intelligent middleware provides many advantages in WSN applications. These advantages range from hiding the complexity of the network communication, dealing with the heterogeneity of applications or devices, and managing system resources. The components of the middleware's architecture are used to integrate WSNs with user applications while keeping the complexity and heterogeneities of the hardware and software hidden. The security of the system and massive data collected from sensors are both crucial issues. However, a number of research studies have attempted to design WSN middleware, but most middleware does not meet the specific needs of a larger-scale sensor network, such as security.

We propose a unique WSN middleware (SWSNM) which can control and monitor sensor data by using intelligent unsupervised machine learning to secure data. In some cases, the communication method between the sensor nodes needs to update and filter unnecessary information that is provided by the sensors, which can increase power

consumption and overhead. The proposed middleware provides an efficient process to transmit sensor data with minimum power usage and overhead.

### **1.3 Research Contributions**

To address the security challenges of WSNs, we developed an intelligent WSN middleware based on an unsupervised learning approach that provides a comprehensive security algorithm that can handle large-scale WSNs. The proposed middleware is able to secure information and resources from malicious attacks and also detect node misbehavior. The special characteristics of WSN such as power consumption, throughput, and network lifetime are taken into account in this contribution.

The proposed intelligent wireless sensor network middleware, which is based on generative adversarial networks, has improved the traditional middleware and other security mechanism but can handle the heterogeneous characteristics of sensor nodes and is capable of filtering and passing only real data. To the best of our knowledge, it is the first time that the GANs algorithm has been used for solving the security problem in WSNs' middleware. Additionally, in the proposed contribution, WSNs' middleware applies a GAN that is capable of filtering and detecting anomalies in the data. The proposed approach is motivated by the limitations of the existing middleware and will improve performance based on the following reasons:

- 1) The proposed techniques provide a unique WSN middleware which can control and monitor sensor data by using intelligent, unsupervised machine learning to secure the data. The power consumption and overhead can be increased by updating and

filtering unnecessary information from the sensors. This problem is addressed through the proposed unsupervised learning.

- 2) From the given samples, the generator network creates fake data very similar to the real data. This fake data is combined with the real data from sensors so that the attackers cannot differentiate between them. In this case, there is no need to generate fake packets or data to confuse the attackers, which significantly decreases power consumption.
- 3) The generator is able to create new data that is very close to the original data. This helps balance the training set for all classes. As a result, the process of learning is more efficient.
- 4) Different analytical models are developed: Confusion Matrix, Visualization, and different CNNs layers confirm the validation of the proposed algorithm.
- 5) We provide a comprehensive comparison with other approaches such as Eagilla for verification of the proposed approach. The following metrics are used in comparison: average energy consumption, successful data delivery ratio, throughput, and end-to-end delay.

## **CHAPTER 2: BACKGROUND**

In this chapter, we propose to introduce a comprehensive challenges and taxonomy of the middleware for WSNs as discussed in details below [20, 21]. This dissertation will only focus through the experiments on security issues because of most middleware approaches lack the security mechanism to secure the network and sensitive data from malicious attacks.

### **2.1 Middleware Challenges for WSNs**

#### **2.1.1 Scalability and Network Topology**

Middleware architectures should be scalable to dynamic resources and interfaces to ensure superior performance as the size of the network grows. Scalability is challenged when any change occurs on large-scale networks. For example, when adding new nodes, the network should adopt and synchronize them with the existing nodes. An efficient middleware design is capable of maintaining a large network and adapting to any changes that occur without impacting network performance.

#### **2.1.2 Security**

With popularity and advancements in WSNs, large chunks of sensitive information are sent over wireless networks. This information can be easily hacked by malicious intrusions and internet attacks. The integration of security parameters in the system's design is necessary to achieve protection.

Most of the middleware focuses on resource distribution, management, and the communication efficiency of the sensor network. However, data aggregation mechanisms, security methods, and resource distribution still remain massive challenges. Security must be part of the middleware design for approaches that use multiple networks' distribution. The middleware reduces the probability of errors or failure by managing multithreads efficiently. Different security mechanisms should be increased by developers of networks during the design of middleware based on SOA. The abstraction layer, wrapping mechanism, and intelligent interfaces are used to address issues of heterogeneous data fusion. The security solutions are considered in several SOM architectures approaches. Al-Jaroodi *et al.*[19] Proposes a generic security service for SOM architecture frameworks that provides various independent security services such as authorization, authentication, and access control.

The SOA based on middleware is designed for Security and Surveillance WSNs with Commercial Off-The-Shelf (COTS) used to program and deploy the data processing applications after analyzing a web service [22]. This approach provides a unique, distributed data processing application in WSNs for Mobile Ad-hoc and Sensor Systems (MASSs). The architecture provides support to complex monitor applications aimed at global security, loose coupling, auto-organization mechanism, simplified connection heterogeneity, and interoperability [22].

In addition, the security mechanisms can be achieved by end-to-end security auditing for SOA as introduced in [23]. This solution provides two new components called Taint Analysis (TA) and Trust Broker (TB) with some advanced features that take from

WS-Security and WS-Trust Standards [23]. TA monitors the interactions of services at runtime and checks information flow between them, which can detect particular events. TB is considered a trusted third party responsible for maintaining end-to-end auditing in the information flow into client requests [23]. In this architecture, the service providers should register themselves closed to TB, which allows user verification by the security of the service providers via TB.

### **2.1.3 Quality of Service (QoS)**

It is important for the wireless networks to support QoS as it pertains to the accuracy of data, coverage and tolerance. The quality of service is important on the application level as well as on the network level. The QoS considers the resource constraints in new and adaptive WSN designs. Providing most efficient and suitable nodes to the client who is in need of the resources has been a major problem in cloud computing. The ability of the system to efficiently locate and provide the needed resources to the clients is vital. Recently, some researchers [24, 25] have tried to increase and optimize the QoS by using computing environments such as Cloud/Grid systems that comprise of several trusted nodes to manage local resources individually. A trust model is associated with each node that accurately evaluates the trustworthiness of its communicating clients [24]. The time-consuming and inefficient process of exploring the whole node space is avoided by allowing each node to efficient allocating resources by finding suitable collaborations. The authors showed the employment of a decentralized approach using Hypertrust where the nodes are organized in an overlay network given the criteria by the client. The Hypertrust gives the client an efficient way of searching for available resources while empowering the

nodes to use their respective trust models to limit the search. The unique node called Task Allocator (TA) allows clients to delegate the selection processes of the task as well as improving the overall QoS.

Another approach, called the partnership based approach [25], is introduced to maximize the QoS by improving and optimizing the global QoS for the large-scale federated resources [25]. This approach combines the trust models for software agents to support the federated computing nodes. The intelligent agents support the model computational nodes which can manage the Friendship and a Group of Membership (FGM). The Friendship and Group Formation (FGF) algorithms used in this approach enable the federated nodes to select their FGM that can increase and improve the global QoS. The authors in [25] showed metrics that allow most suitable resources in such Grid/Cloud systems. Potential collaborations and competition between resources providers for clients' needs are explored by the federation of computing.

#### **2.1.4 Fault Tolerance**

Many studies are focused on how to recover the system from failure. SOAs have an important feature that can maximize information reuse by separating the implementation of services from the interfaces and enabling failure-resistant networks. The Service-Oriented self-healing approach referred to as “clinic” is proposed in [26]. The self-healing service can, with help of SOA, detect faults and heal them, isolating them by only using information that is available from other services in different networks. The evaluation of the self-healing approach is applied on communication faults through a routing protocol called Multi-path, Multi-hop Hierarchical Routing (MuMHR) [27].

### **2.1.5 Heterogeneity and Data Aggregation**

The heterogeneity among the hardware, communication devices and configurational operations have to be granted for the middleware. The heterogeneity of the components may be an issue in large-scale applications of wireless sensor networks. In order to minimize the volume of data for transmission, a sensor network uses data aggregation quality. This ensures that redundant data is not generated in the memory, saving costs through memory usage and energy through processing time. This is a more data-centric approach in comparison to the conventional, address-centric approaches. Therefore, with smaller, more compact sensors, the available battery power is always limited. The middleware is designed to manage limited power by designing efficient processes and capabilities of the sensors. Mechanisms to ensure efficient power consumption are necessary for advanced wireless sensor networks.

### **2.1.6 The Taxonomy of Middleware Architectures for WSNs**

The middleware architectures for WSNs have been used widely to reduce the complexity of WSN applications. The classification of middleware architectures approaches are proposed in the literature [28, 29]. The middleware architectures based on SOA for WSNs can be classified based on the applications targeted as shown in Figure 2.1. Additionally, Table 2.1 presents the comparison between different middleware architectures designed for WSNs.



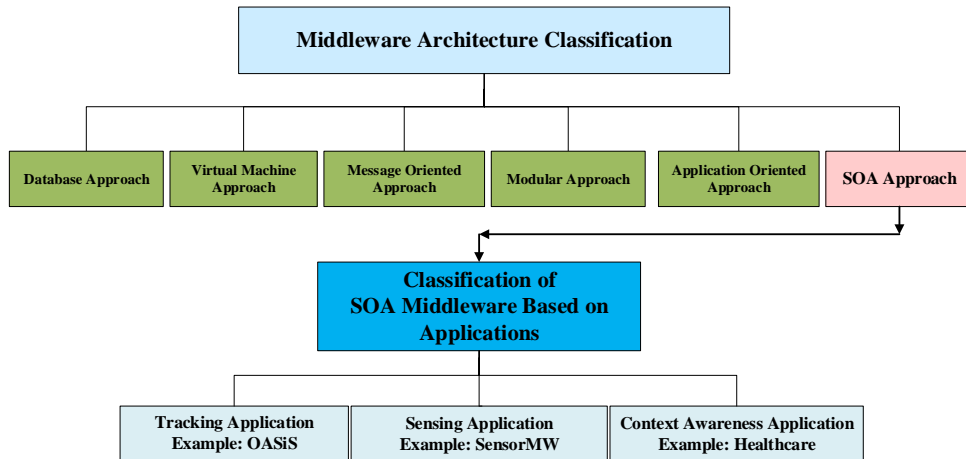


Figure 2.1. The classification of middleware architectures for WSNs.

Table 2.1. Comparison of different middleware architectures approaches.

Middleware Approaches	Scalability	Heterogeneity	Ease of use	Power Awareness	Application Type	Security	QoS
Database Approach	Not Supported	none	Yes	None	Event driven applications	None	None
Virtual Machine Approach	Supported	Not fully Supported	Little	Supported	Dynamic Applications	Yes	None
Message Oriented Approach	Supported	Not fully Supported	Yes	Supported	Event driven applications	Little	None
Modular Approach	Supported	None	Yes	Supported	Dynamic Applications	Yes	None
Application Driven Approach	Supported	None	Yes	None/Partial	Real-time applications	None	Yes

### 2.1.7 Database Approach

This approach considers the entire sensor network as a distributed database. The limitations of this approach is that it does not support real-time applications and only provides approximate results. The example for this middleware architecture is Sensor

Information Networking Architecture (SINA) [30]. The SINA is capable of monitoring changes within the network.

### **2.1.8 Virtual Machine Approach**

The Virtual Machine (VM) middleware architecture is a flexible approach that allows the developers to write the applications in separates modules. The modules are distributed in a network by using specific algorithms. Even though the issues related to the utilization of the resources and power consumption are addressed in this approach, the limitation of the VM approach is the overhead.

### **2.1.9 Message-Oriented Approach**

This middleware approach is used the publish/subscribe mechanisms which can facilitate the message exchange between the base station and the sensors nodes. The advantages of this middleware is that it supports loose coupling and asynchronous communications between the sender and the receiver.

### **2.1.10 Modular Approach**

This approach divides the applications as modular programs that help the integration and the distribution through network by using mobile codes. The limitations of this approach is that it does not support the heterogeneity sensors hardware.

### **2.1.11 Application Driven Approach**

This middleware allows the application to identify their QoS requirements then can modify the network according to application needs. The Middleware Linking Application

and Network (MiLAN) is one of the examples of the application driven [31]. The limitation of this middleware is not supported the heterogeneity sensors hardware.

### **2.1.12 Service-Oriented Architecture Approach**

The middleware based on SOA is proposed in detailed in Section 5. The Service-Oriented Middleware (SOM) architectures are presented below and is classified based on the applications targeted.

#### **2.1.12.1 Sensing applications**

SensorsMW is a SOM architecture that allows applications to configure and adapt to the low-level hardware based on their particular requirements. SensorsMW has been developed for vent monitoring and periodic measurements. This middleware is used to test temperature measurement applications.

#### **2.1.12.2 Tracking applications**

The OASiS is a tracking application for example fire detection and vehicle tracking [32, 33]. The WSN-SOA has been tested for surveillance applications with the ability to detect seismic vibrations [34, 35].

#### **2.1.12.3 Context awareness applications**

The middleware has been designed for context awareness applications and testing for healthcare and smart environments [36-39].

## CHAPTER 3 LITERATURE SURVEY

### 3.1 State of the Art Middleware Approaches for WSNs

The middleware architecture is the best platform to develop WSN applications to address hardware challenges such as QoS, security, and heterogeneity. The following is a brief description and summary of the selected approaches that are considered middleware architecture for WSNs. An open sensor middleware model based on the SOA for WSNs should have the ability to integrate, in real time, context data with flexibility, reusability, programming abstraction, and simplicity. In addition, many studies consider the network-embedded devices in different applications, such as managing enterprise architecture [40], smart home and industrial applications. These applications can be classified into two categories: SOA-ready devices and SOA not-ready devices [41]. The issue of integrating WSNs into IP-based networks and Internet is addressed in [41]. It provides solutions for implementing SOA based on SOA not-ready devices. A micro SOA model is implemented based on  $\mu$ IP protocols that only use Hyper Text Transfer Protocol (HTTP) philosophy instead of HTTP protocols [41]. The exchanged data can be between network devices on the same layer or between the embedded and middleware layers through efficient lightweight protocol called JavaScript Object Notation (JSON) (instead of XML format) [41]. JSON can reduce overhead and power consumption, request size, and complete request time. The  $\mu$ SOA uses the middleware layer. The middleware layer manages access to WSNs by filtering and protecting the system. The filter mechanism removes unnecessary information from any HTTP request. Other mechanisms the middleware provides are security, domain name services, and authorization. However, this middleware does not

address the issue of a heterogeneous network [41]. Similarly, the middleware can be designed based on a function block programming abstraction for a WSN that enables the operations to be done in a dynamic environment to reduce overhead and complexity. These features are completed by applying SOA with a Mobile Agent (MA) [42].

### **3.1.1 Eagilla**

While middleware systems are primarily developed for WSNs, different agents use it for various applications to detect any intrusion using the agent model. In [43], the authors introduced mobile agent middleware called Eagilla that is integrated with WSN for sensing data. This framework provides scalability and flexibility to the network. The agent is responsible for communication in this approach and acts as a mobile to move around in the network and update required tasks as shown in Figure 3.1. The sensor nodes in the network acts as a cluster head (CH) and runs their agents based on the functionalities of CH. In this approach, CH is applied to increase the network scalability and application controlled by CH instead of the base station. There are three types of sensor nodes; free, client, and server. Free nodes act as independent nodes and can leave or join the cluster/network at any time. The sever nodes are the CH that pass the communications to and from the base station. Finally, client nodes that have communication authority with CH. This framework increases the network scalability and supports heterogeneity sensor hardware. The Eagilla framework lacks security system since it is dealing with large-scale network.

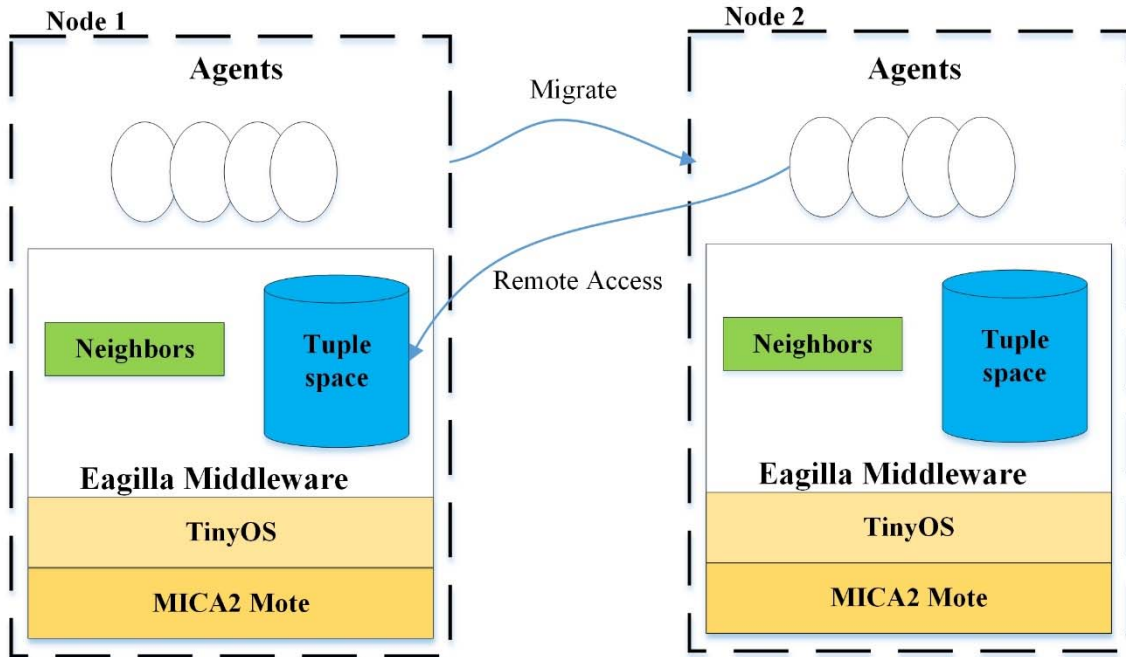


Figure 3.1. Eagilla Middleware Approach.

### 3.1.2 USEME Approach

In [44], the authors propose Ubiquitous Services on Mote (sensor) Environments (USEME), a new framework that uses Service-Oriented high-level programming models [44]. It also supports middleware development of Wireless Sensor and Actor Network (WSAN) applications [44]. Efficiency and scalability are realized through the middleware, which has various sensor nodes that can share a mutual behavior and control the use of services. The drawbacks of priority and deadline are considered in this approach, which can deal with the real-time actions of the services requirements. This approach combines macro-programming with node-centric programming. Different prototypes are developed by using three motes: Crossbow family MicaZ motes, Imote2 (Crossbow Technology, Inc., Milpitas, California, USA), and SunSPOT, as shown in Figure 3.2.

The authors of [44] did not provide data on whether the architecture is a distributed or centralized model, or on the methods of used services. The proposed framework did not consider the accuracy and QoS constraints. The solution for this limitation is to provide an application designed to define a set of services, nodes, and events. This approach should be supported in real-time, which can allow the programmers to recognize (define) QoS among the services by using communication. The study in [45] uses the same techniques as above but focuses on middleware to support USEME. This Service-Oriented Framework is used to deploy lightweight services on the sensors and actors. Two different prototypes are used to implement this approach, which are SunSPOT devices and Imote2.Net from Crossbow. The middleware provides an easy way to address any differences in the nodes as they pertain to the dynamic and logical relationship between the services in the application. The features of this middleware make the network more secure, facilitate updates, and ensure controlled deployment.

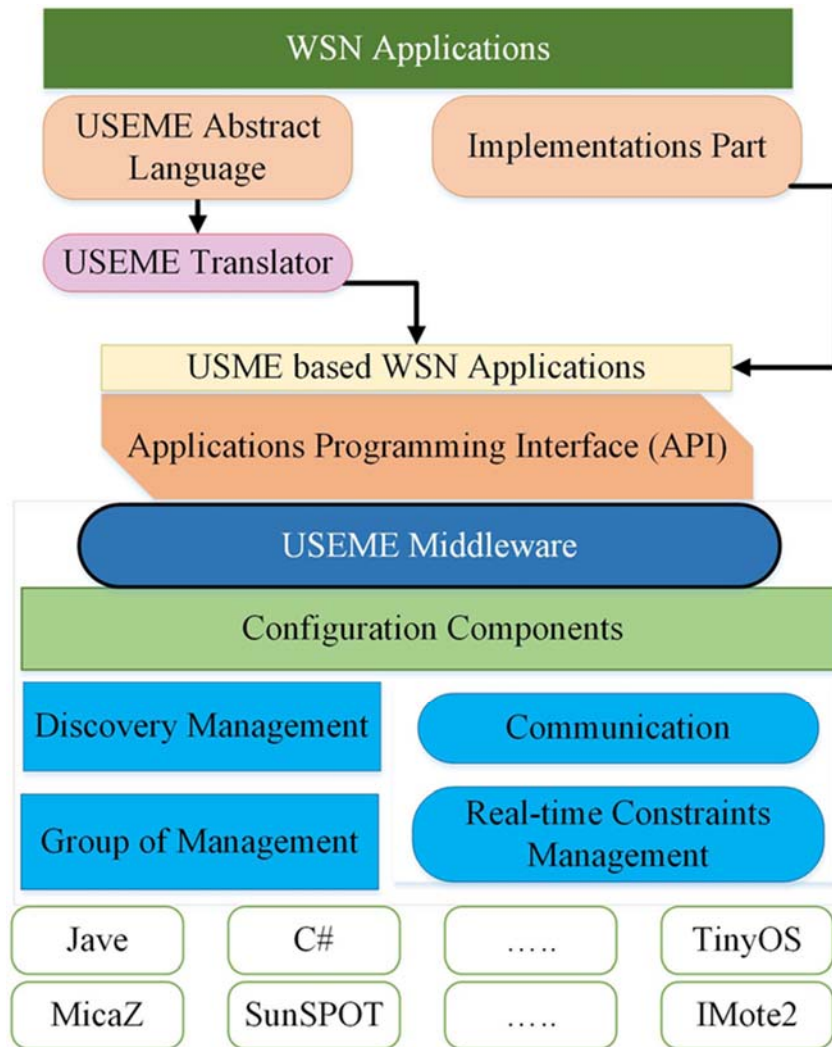


Figure 3.2. USEME Architecture.

### 3.1.3 SOMDM Approach

In [46], the authors proposed a unique, SOM architecture with a Message-Driven architecture for an ambient aware sensor networks (SOMDM) technique [46]. The limitations of web service as well as time, power, and memory consumption issues in the physical layer are addressed in this middleware. This approach has enabled the SOA to



reduce process load in real-time during query processes, warning the system, and performing processes for ambient aware sensor networks. The system approach uses the data filtering mechanism which has been used to filter the event of interest. The object codes are the nodes in a sensor network that will follow the ambient program model, which permits nodes to communicate in two asynchronous ways. The object codes should go to a data filter box with intelligent mechanisms to filter normal and abnormal data. Moreover, normal data goes to the Management System Database (MSDB), which stores the data that comes from the Data Filter Box and can be used to query other parameters. This approach is tied to abnormal data, which should go to the message queue through a Java Message Server (JMS). Then, it Normalizes the Message Router (NMR) using a fast response time in warning messages. The NMR can reduce the load of discovering and subscribing the route. It provides the best solution for communication time between services. This approach does not consider security mechanisms for internal and external communication between the nodes and client. The quality of service should be considered in this approach in order to obtain better accuracy and faster operations.

### **3.1.4 Mobile Web Service Approach**

In [47], a Mobile Web Service (Mob-WS) middleware that provides the best management and representation of wireless networks was designed. The Mob-WS is used as a back-end resource for in-network computations. The Mob-WS middleware addresses the issue of inflexible collector nodes. The middleware deployed with hosting a long-lived asynchronous services. The Mob-WS middleware is deployed on the collector node, which can make it independent of any transmit protocols. The collector node concept is used to

perform Mob-WS base in-network that can cooperate, control, and monitor. It is the best representation of the network. The service processing model is based on in-network services, and these services are implemented on the sensor by using the computation in wireless networks [47]. This method increases the scalability of the network and makes decisions locally based on the sensing data [47]. The limitations of Mob-WS designs do not provide mechanisms to secure accessing to the services or managing the QoS on the Mob-WS. It cannot handle multi-interfaces.

### **3.1.5 MiSense Approach**

In [39], the authors proposed MiSense, Service-Oriented, components-based middleware layers that support the distributed sensor applications with a different performance of requirements [39]. The MiSense middleware provides an abstraction layer in between an underlying network infrastructure and the application. In addition, it provides an abstract programming model to the WSN application that can maintain the balance between network lifetime and QoS requirements for the application. The content-based, publish/subscribe service, provided by MiSense, enables the designer of any application to adapt to the services. MiSense also helps break down the middleware into different layers. The layers can be self-contained, and interact with the components that address the issues of tension between the requirements' optimization, flexibility, and the ability to develop reusable WSN applications with efficient energy.

The middleware has three layers: the communication layer, common service layer, and domain layer, as shown in Figure 3.3. They handle data aggregation, event detection, routing, and topology management. This approach uses adapted rules for the middleware,

which can increase the data accuracy and bandwidth. The energy consumption decreases by an increased data rate and changes some sensors into the sleep state mode [39]. The MiSense does not support heterogeneous data that comes from different networks. It is also dependent on TinyOS (TinyOS Alliance). This approach does not determine the standard of SOA used between the gateway and the applications [39]. This SOA has flexibility and interoperability limitation between the various platforms provided in this approach. Since binary forms are used for remote procedures, the execution of SOA applications can be slow. The results can increase the processing time and energy consumption.

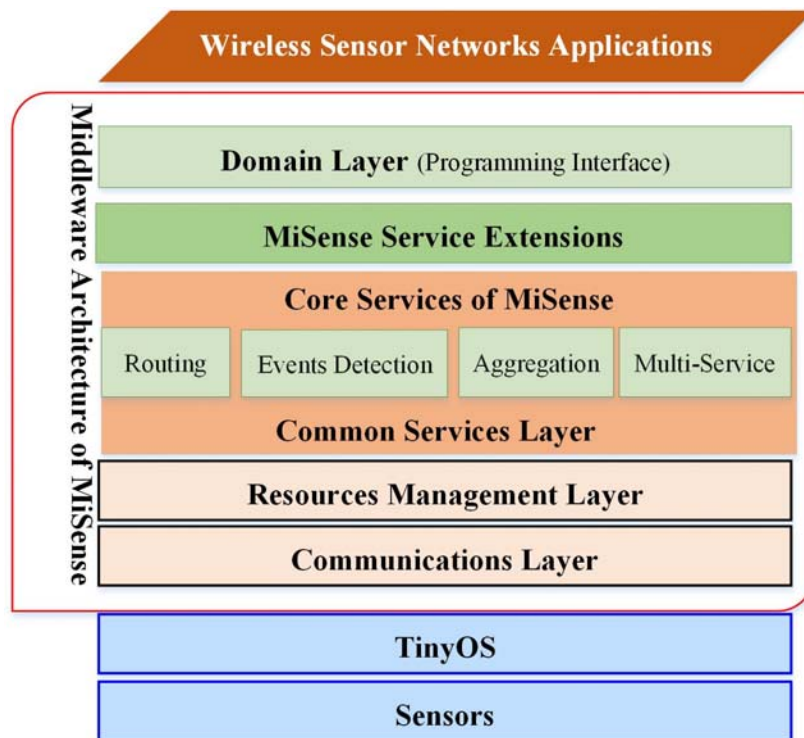


Figure 3.3. MiSense Architecture.

### 3.1.6 Sensors Middleware Approach

In [48], the middleware architecture is used for QoS configuration and the management of the WSNs. The authors presented Service-Oriented, adaptable, and flexible

middleware (SensorsMW). This middleware supports the dynamic management of heterogeneous data. The middleware has the capability to hide the complexity of low-level sensor devices [48]. Once the SensorsMW abstracts the WSNs, it acts as a gathering service and easily integrates into the enterprise information system. The applications collect the sensed information by using a web service. Consequently, the SensorsMW allows high-level applications to configure a data collection level for the WSNs in a simple manner. This approach enables the application to collect data by using a web service, which can guarantee flexibility in the delivery of the data. Furthermore, this architecture enables applications to independently negotiate from run time by using a technique called the contract negotiation approach, based on a Service Level Agreement (SLA) [48]. SLA stops the application from requiring knowledge of the other QoS applications. The SLA enables the application to reconfigure and maintain the network within its lifetime. Every end-device node contains Crossbow MicaZ (Crossbow Technology, Inc., Milpitas, California, USA) [48]. Every node has TinyOS 2.0 (TinyOS Alliance) [48]. The implementation only focuses on service level management and does not provide any mechanism to handle a secure execution or communication. Typically, in WSNs, a faulty node is factored into the performance of the system in order to generate the correct execution. Unfortunately, this approach does not take this fact into consideration. In addition, the resource management of the system does not support any node with low capacity. The details of QoS parameters, resource surveillance, scalability, and data evaluation are not provided.

### **3.1.7 OASiS Approach**

The OASiS is an Object-centric, Ambient aware Service-Oriented Sensor network applications, and Service-Oriented Framework introduced in [32]. The OASiS middleware includes various services, such as a dynamic service configurator, node manager, and object manager [32]. It can easily provide dynamic service discovery and configuration, data aggregation, and support heterogeneity (the application developers aren't required to have any experience in sensor programming). The middleware architecture is supporting OASiS and is capable of tracking the application. The ambient aware sensor network consists of efficient mechanisms that can detect failure if any node drops out during the application execution or communication. The network application is retrieved by applying an isolation and recovery technique [32], providing a stable configuration achieved by taking some advantages of OASiS-SOA [32].

The authors introduced the sensor network application in [33] that is obtained as graphs of modular and autonomous services with determined interfaces which allow them to be published, discovered, and provide a mechanism to integrate the services from a heterogeneous sensor system [33]. The SOA model allows the composition of a dataflow application [33].

### **3.1.8 SOMM Approach**

The Service-Oriented Middleware (SOM) architecture called (SOMM) is described in [49]. It can support the application development for Wireless Multimedia Sensor Networks (WMSNs) [49]. Several middleware designs are proposed for WSNs but this

middleware is not suitable due to its constrained resources. SOMM consists of two components that are service registry servers [49]. SOA is used in SOMM, which leads to scalable and dynamic server node networks which can provide several services to different clients [49]. In this case, the network has the ability to handle many clients simultaneously and add new functions to the network [49]. The application code size is decreased by using a Virtual Machine (VM) as middleware, which supports the reprogramming of the nodes. The VM is located between the application layer and the operating system. The VM provides code mobility that is helpful for Generic WMSN (GWMSN). The overview of the middleware solution [49] is shown in Figure 3.4. The codes of each service are stored in specific nodes that have enough memory space (repository) to act as the mobile agents [49].

SOMM only supports Transmission Control Protocol (TCP) binding, which is in binary format, not SOAP. HTTP binding provides an overhead and increases the power consumption of nodes. The transmission of multimedia in WMSNs is supported by using some of the middleware advantages, heterogeneous nodes, and QoS. The cost of the application development is decreased while improving the scalability and modifiability of the network, which can increase power efficiency [49]. Additionally, the authors in [50] introduced a Service-Oriented Agent-based Middleware called SAWM based on a network architecture that is proper for WMSNs [50]. The middleware of WMSNs handles QoS, managing bandwidth network heterogeneity.

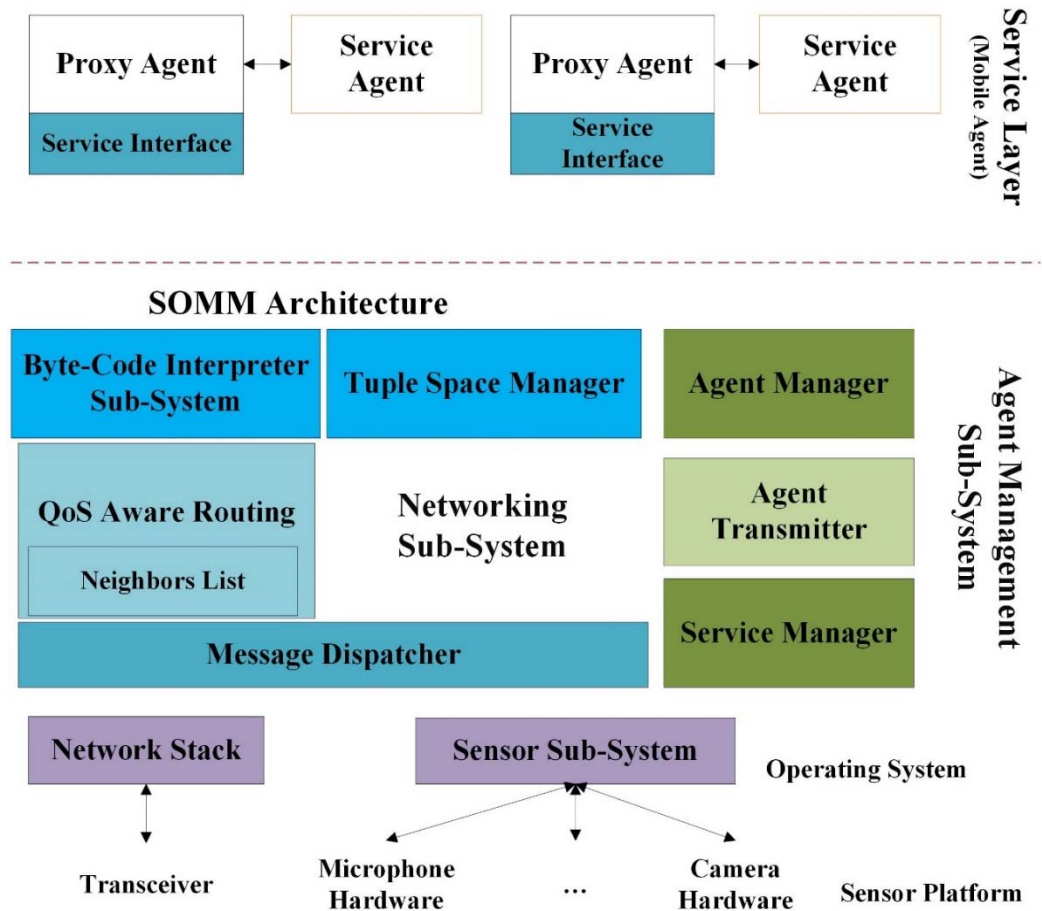


Figure 3.4. SOMM Architecture in the Server Node.

### 3.1.9 TinySOA Approach

TinySOA enables programmer access to WSNs from an application by using Service-Oriented API [51]. This approach helps integrate a WSN with the internet application, providing an abstraction for the developers' applications. The TinySOA acts as a basis for the middleware system and has the ability to allow application developers (that do not deal with low-level of WSNs) to obtain data from the sensors. The middleware helps integrate all the elements into the architecture.

TinySOA consists of two types of services: internal and external, as shown in Figure 3.5. They are provided by the node, gateway, server, and register components. The mechanism of TinySOA provides node discovery and gateway for the WSN infrastructure. The gateway component is a bridge between external applications and the WSN. The hardware platform of TinySOA includes MicaZ motes (Crossbow Technology, Inc., Milpitas, California, USA) [51].

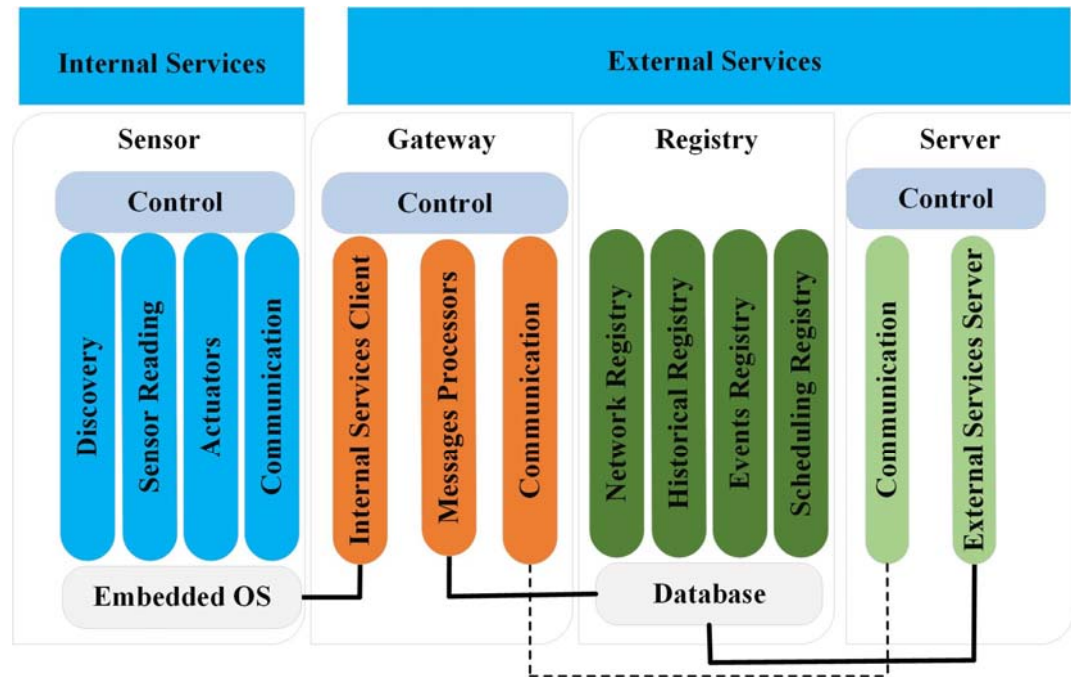


Figure 3.5. TinySOA Approach.

### 3.1.10 ESOA Approach

Another solution to the problems generated by an SOM architecture approach is the Extended Service-Oriented Middleware Architecture (ESOA). The ESOA, as discussed in [52], provides integrated services, customizes sensor networks, and manages applications. The ESOA is inserted above the actual SOA model and below the LiteOS operating system,



as shown in Figure 3.6. This architecture allows users to develop new applications through mix-and-match services without any programming efforts by the developers. Since this system supports the heterogeneous WSNs, it executes various applications on multi-platforms. The ESOA approach is limited because it does not provide any methods of user accessibility data collection to the services. Also, ESOA is not applied in real time.

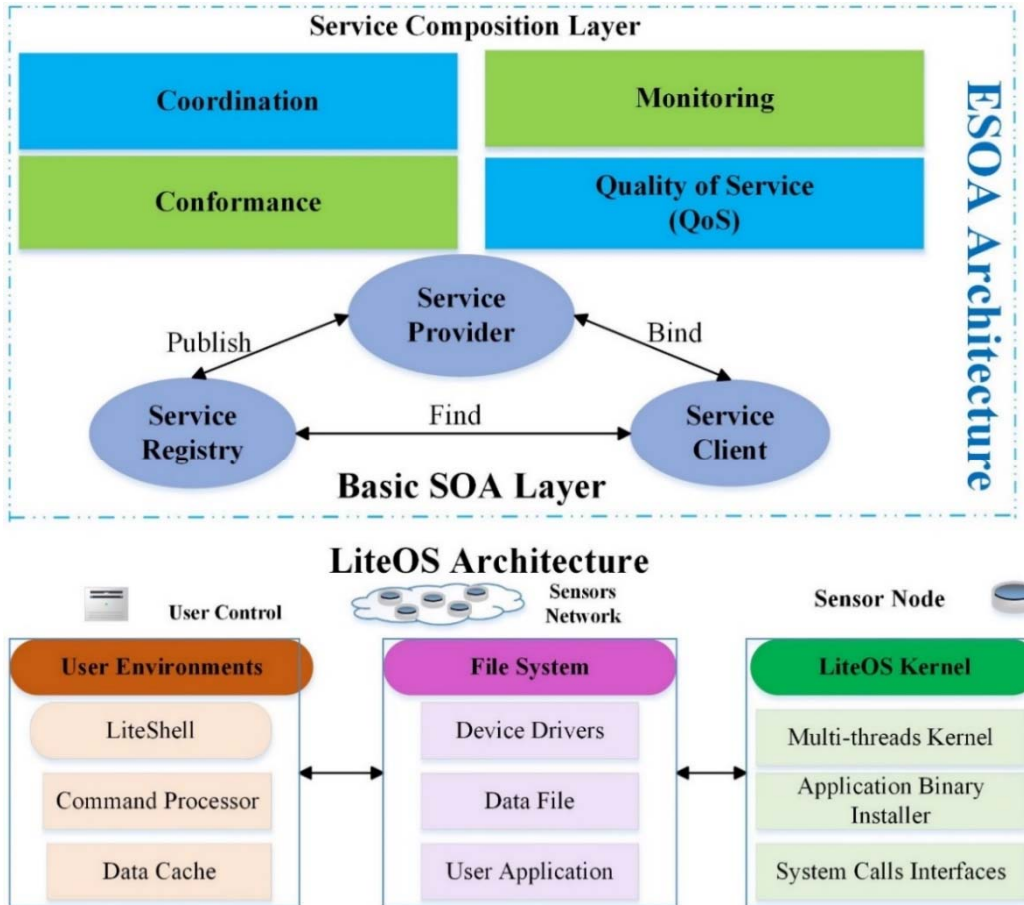


Figure 3.6. ESOA Approach.

### 3.1.11 HealthCare Approaches

Within the healthcare industry, SOA is widely used to improve the transmission of important patient information. By linking the data to the healthcare community, doctors and caregivers have remote access to all of their patients' daily activities. The monitoring

system for a patient using SOA [38], An SOA approach is applied into WSNs to design different applications to monitor the patients for long periods of time [53]. Through SOA, the sharing of patient data has become cost-effective and secure. In [38], WSNs are introduced as an integrated with a web service, using context-aware SOM architecture that increases system flexibility. A web service combined with Radio Frequency Identification (RFID) is necessary to manage patient information. It is responsible for collecting, storing, and making clinical data available [38]. The context-aware service searches the patient information and obtains the most accurate output without errors. In its own capacity, RFID can access secured patient information. RFID is designed as a smart card accompanied with a verifiable, individual patient photo ID to obtain patient history that helps doctors give accurate diagnoses with less fault detection. This process produces an improved QoS and reduces costs.

### **3.1.12 Other Middleware Approaches for WSNs**

The implementation of SOM architecture is based on Devices Profile for Web Services (DPWS) architecture that contains new layers [54]. The SOM architecture provides a mechanism that mediates data exchange between a web service and the heterogeneous sensors [54]. The limitation of resource constraints in WSNs are addressed by using optimization mechanisms that can reduce the overhead required through using traditional WS. The energy-aware mechanism is important for extending the network lifetime. This architecture focuses on sensor nodes that impose restrictions on the resources and data aggregation. Also, SOA controls the energy consumption of each sensor by reducing transmission messages to the base station using multi-hop communication.

DPWS used inside the middleware has various new components that include binary encoding, WS-eventing, and a roaming manager. The binary encoding mechanism is used instead of an XML message to reduce the overhead generated by XML. Before messages are transmitted between the layers, they should be encoded in a binary format. WS-eventing removes the requirement for necessary periodic call services and the user can subscribe to the interface of service eventing [54]. Also, WS-eventing has the ability to report to clients that a change in the data occurred. This method helps save the limited network bandwidth [54]. This approach lacks the mechanisms that can handle interaction with different components.

Another SOM architecture approach to consider is called the Service-Oriented Wireless Platform for Acquisition and Control (SOWPAC) [55]. SOWPAC is introduced in [55] as a design with an open interface to have an efficient and cost-effective deployment. Most of the platform studies focus on the industrial acquisition and control of using WSNs, which are considered only at the network, node, or data abstraction level. This consideration lacks a holistic point of view, which can limit the use of these approaches [55]. The middleware API is used to manage data, facilitate communications, and define the processes of data exchanged between functional blocks. The SOWPAC consists of a basic element called Remote Terminal Unit (RTU), which is responsible for remote sensing and actuation. The WSN-gateway is used as an intermediate element to send data from the RTU to the Central Control Point (CCP) through the WSN. The internal database in an RTU [55] can store sensing data and has the capability to recover from any failure of communication and reset the entire network. The Central Control Point (CCP)

provides a user interface and application programming to manage platforms, data, and services. It also offers a Service-Oriented Protocol based on SensorML that provides an easy way to integrate a web service with high-level applications. The WSN-gateway is responsible for translating data and meta-data [55].

In addition to an Open Framework Middleware (OFM), [56] introduced a comprehensive framework designed a middleware architecture for WSNs. OFM architecture consists of a protocol stack which has some limitations, such as overhead and load on execution. The Hybrid Native Architecture (HNA) [56] addresses the drawbacks of the OFM by removing the stack-based protocol layers. It runs the Service-Oriented OFM Micro-Middleware through the device abstraction level [56]. The solution of HNA lies within system distribution services and the management of node operations which can interact with low level resources. In order to solve the above-mentioned issues, HNA should collaborate with OFM functionality to improve WSNs. Therefore, OFM-HNA enables access to available resources in the nodes through implementing a standard abstraction system that does not require access to the device. The OFM-HNA approach provides flexibility, adaptability, and reliability with control of the WSN by using models. These models deploy, manage, and update the network in the device, gateway, and enterprise levels. However, the proposed architecture does not provide any collaborative results of OFM functions with WSNs.

The Rescue and Crime Information in the Cloud (RCIC) [57] is based on SOM architecture. RCIC consists of a set of heterogeneous sensor nodes that form a cloud-based system in MANET [57]. The sensors send data to the cloud to process and analyze it. Then,

the data is normalized through the middleware and transmitted to the Rescue and Crime Information System (RCIS) [57]. RCIS is a method that individually assesses secure data versus at-risk data. RCIS detects natural disasters or criminal activities. It can easily monitor any event by providing a fast response time. The simulation result of 500 sensor nodes shows that the power consumption and range size of each node is reduced by using clusters. Each cluster consists of 100 nodes executed in parallel. RCIS's limitation is in its accuracy. It is not accurate enough to handle complex services or networks. The network uses a lot of data that causes processing delays. Even though the RCIS acts as a filter, it should enhance the database to filter unnecessary data. If this filtering takes place, overhead and processing delay of data will decrease and the network accuracy will increase.

Another SOM architecture called Service Mid-Tier Component (SMC) based on SOA is introduced in [57]. In this technique, each component is represented as a service within the middleware framework. This approach has a repository that includes various types of interfaces and a middleware. It handles any type of delivered request and then identifies a suitable interface from the repository and links it to the service. It can decrease overhead, storage space, and power consumption on each node in the network. Each layer should be independent of others because individual layers provide a self-contained module increase flexibility and scalability within the system, and protect individual data. In this case, the repository should use secure algorithms to establish interactions with the nodes. In [57], the proposed method is used to handle the traffic route between the sources and destinations; however, it should be optimized to increase quality of service in the system.

In this approach, the authors need to evaluate additional applications in order to compare their results with other techniques.

Another middleware proposed is based on SOA through a web service [58]. It addresses different issues such as the serviceability of WSNs and the power efficiency for sensor application services [58]. The solution for serviceability occurs in the application of a Web Service Resource Framework (WSRF) within an Open Grid Service Architecture (OGSA) [58]. The power efficiency is solved by WSR. A web service based on the Markov Decision Process (MDP) produces query optimization techniques [58]. However, WSRF does not provide any quality of service for Service-Oriented for WSN applications [58], which is a critical issue especially in the case of massive data. The parameters of the quality of service such as data and process accuracy as well as the speed and failure rate of the operation should be considered. Data and system security are not addressed in this approach, and therefore can impact the system's applications. Under OGSA, the WSRF transfers massive data between WSN applications; it should provide a method to control any loss or delay of data.

Similar to the preceding studies, the authors attempted to apply the quality of service (QoS). The active QoS Infrastructure of WSNs within SOM architecture is labeled as (QISM). The QISM was introduced in [59]. QISM is a software layer located between the protocol stack and applications [59]. It communicates with the layers by using API standards. The design of QISM has mechanisms and metrics that guarantee QoS for the entire network. The lifetime of the network and its application is increased through applied switching between the nodes [59]. By using two different regions of two different nodes,

the network adjusts itself to the node with the highest power. The limitation of this approach is that there is no strategy for low-cost QoS monitoring processes, detection of QoS degradation, and data or service aggregation exists. The QoS degradation can be addressed by using the monitoring frequency approach [59]. This approach is more cost-effective than static or dynamic approaches. The management of the system and service should focus on the node and service level. The data aggregation in a sensor network can deal with simple data; however, it cannot deal with complex data.

Furthermore, many approaches of SOM architectures attempt to implement a flexible and scalable architecture with less cost. In this study, authors present an elastic sensor actor network (ESANET) environment [60], which proved to be more cost-effective. These applications run on top of SANET shared resources. ESANET is a software system that can bridge the gap between existing software and the next generation of SANET. The Role Oriented Adaptive Architecture (ROAA) is used to build a collaborative and adaptive ESANET software. The middleware architecture is used to achieve the goal of ESANET. The security mechanism is applied to the Nano kernel Middleware, an outside and inside security mechanism within the system. The limitation of this approach is that it does not provide details about the system's performance, accuracy, and overhead.

The issues of integrating SOM architecture with sensor networks in the internet of things (IoT) technology were addressed in [61]. The authors proposed this type of SOA based on the middleware architecture. The features of SOA include a publish/subscribe mechanism that mediates communication between the IoT technology and the applications

of existing automation systems. The publish/subscribe mechanism monitors traffic and manages asynchronous events. The IoT appears as either wireless sensors or identification tags. The middleware allows a smooth integration between heterogeneous technologies within applications [61].

According to [62], the existing Laboratory Information Management System (LIMS) at the Center for Life Science Automation (CELISCA) laboratories combined SOA with WSNs (SOA-WSNs) [62]. This approach relied on Sensor Web Enablement (SWE) and Sensor Observation Services (SOS) that provided the sensor measurement of data in different WSNs [62]. The architecture used a DPWS-based web service to assist in the cooperation, abstraction, and device orchestration of the LIMS services. In Life Science Automation (LSA), Carbon Monoxide (CO) and Hydrogen (H<sub>2</sub>) must be regulated by sensors [62]. Unfortunately, WSNs do not support these dangerous gases. However, SOA-WSNs in LIMS were designed to detect any of these risks and block any disasters within LSA to guarantee a valid analysis procedure. The LSA observation service analyzes the actual sensor readings and will release the necessary responses in the case of any abnormalities. The flexibility, usability, and extensibility of this architecture is increased through a developed WSN-based service infrastructure. In [62], the researchers claim that this approach decreases cost and setup times. However, since no results were provided, this approach cannot be fairly evaluated.

### **3.2 Service-Oriented Architecture Approaches for WSNs**

This section discusses the latest approaches based on SOA. SOAs do not apply middleware architecture on their schema.



### 3.2.1 Healthcare Approaches

The Service Layers Over Light Physical Device (SYLPH) architecture [63] consists of layers added over the application layer in each WSN stack [63]. SYLPH is a unique architecture that helps in integrating SOA with WSNs that can be used to build a system based on Ambient Intelligence (AI) for maintaining patient information, which was presented in [63]. The AI provides an intelligent distributed system, allowing effective communication irrespective of location and time [63]. The SYLPH gateway is connected to different sensor networks by using various hardware interfaces. It enables two device types (either the same or different) to work together, such as ZigBee and Bluetooth devices. The system improves the healthcare monitoring of home-bound patients through a prototype system. The drawback of SYPLH is that it has not been tested in real-time.

Similarly, in [64], a unique framework based on SOA with Wireless Body Sensor Networks (WBSNs) and Web Services (WB) was proposed. The framework provides healthcare services to monitor elderly people and allow doctors and nurses to access patient information. This framework provides a mechanism to keep the healthcare data secure and private, based on the authentication mechanism which decides to allow or reject the user access request. This service helps elderly individuals by carrying a very lightweight and efficient biosensor. The feature of this framework includes reduced memory space, interoperability of service, maintenance cost through storing strange data in a central server, a fast response time, increased privacy, and throughput. The limitations of this framework include overhead, due to its use of XML and SOAP.

The concept of SOA is used in tele-monitoring. SunShine is integrated with distributed WSNs and the internet to perform complex tasks [65]. SunShine is a web-based system that manages data after collecting it, by analyzing the sensing data to see if it's normal or not. However, applying SOA enables the creation of a Web Management System (WMS) for SunShine, providing flexible and reusable architecture. It can easily extend the sensing region coverage in web-based software design and monitor patients at all the times. The authors do not provide any security method to keep the patients' data secure at all times, especially communication between clients and their doctors. Patients' information is not sent or updated securely.

Correspondingly, the architecture of a tele-monitoring system can remotely monitor patient data. It has the ability to support efficient retrieval of information and addresses the QoS for visualizing data. SOA-based data architecture for healthcare monitoring with assistance from an algorithm that uses Extract Transform and Load (ETL) and Oracle Business Intelligence Enterprise Edition (OBIEE) is introduced in [66]. The drawback of this architecture is that it does not support heterogeneous sensors.

### **3.2.2 Service-Oriented Device for Smart Environments**

The Simple Object Access Protocol (SOAP) is deployed based web service on the node without a need to build it on the gateway. This approach supports and integrates into a legacy IT system by using SOA in a simple manner; this can support the heterogeneities at low level, without requiring additional middleware. The experiments of this architecture are done using Mulle, which is a resource-constraint sensor platform. Every device consists of SOA interfaces, which can enable interaction with high-level business applications

without using intermediate gateway protocols. An efficient lightweight TCP/IP stack combines with IwIP and gSOAP web service toolkit, increasing the processing time for SOAP messages. This design supports different network layers. The security is considered by using the DPWS, as the sensor nodes in this approach are behind a firewall enterprise. The approach is only suitable for noncritical applications. In this method [67], sensor data aggregation reduces transmission time and increases battery life is shown. The processing of SOAP messages generates overhead, but not as much as the message transmission. The limitation of this approach is the performance of overhead communication [67].

### **3.2.3 Network Discovery and Selection Approach**

Wireless mobile networks have a limitation due to the heterogeneous network environments [68]. The mechanism to discover and select the best network can be reduced during the transmission of network services that takes place when heterogeneous networks exist [68]. The Access Network Discovery and Selection Function (ANDSF) was proposed but still has challenges such as collecting and enabling network data from access networks, making available this information to be available for network discovery and selection, and updating this information in real time. The SOA provides a flexible mechanism to discover and select a network in wireless mobile networks [68]. The SOA is applied to ANDSF to process heterogeneous wireless mobile networking. Costs are reduced because the notification message consists of only an updated network state and does not contain the entire service description. Network service descriptions keep the most recently updated information at the network service registry. This mechanism helps discover and select the most optimal access network in real-time instead of republishing all network service

descriptions. The system increases the capability of the network service description by using the capability matrix [68].

### **3.2.4 Open Geospatial Consortium Approach**

Recently, internet services have applied Geographic Information Systems (GIS) that support environmental observations such as weather, a fire alarm, and indoor surveillance systems. As introduced in [69], a WSN Application Service Platform (WASP) is a novel sensor control service with web/GIS based architecture [69]. The WASP (acting as a cloud service) manages data through many data recovery points by sensors that are sent to the server for query by the user. The users are not able to identify between raw and processed data, which results in the loss of necessary information. The WASP is used to manage data and provides various mechanisms, such as data presentation, remote control functions, and security. The limitation of this approach is addressed in [70]; the sensor web enablement was developed to provide a solution for raw data identification and issues related to the mashup between WSN applications. The Sensor Web Enablement (SWE) is based on the Data Observation and Event Notification framework (SWEDOEN) [70] and has been used for smart home applications. This framework has a flexibility of application with WASP and can assign the action and message flows between SWE components. These approaches are not providing mechanisms for a WASP with GIS web service to handle large heterogeneous data in real-time. The middleware can handle a massive amount of this data by using different interfaces, languages, and content messages to convert data to fit the users' needs. The accuracy and performance of their approach is not considered.

Moreover, Open Geospatial Consortium with Sensor Web Enablement (OGC SWE) is capable of real-time monitoring. The integration of WSNs into SOA by using a web service proxy linked to high-level SWE to low-level sensor platforms is presented in [71]. OGC SWE is applied for the sensor description, and observation with open Message Queue Telemetry Transport (MQTT) provides a suitable solution for low-level uplink from the WSN to the sensor web. The communication at the proxy layer is done through MQTT. The MQTT is used to solve the issue of one-way communication by using bidirectional communication for OGC SWE. This system is required for WSNs to have web-enabled remote management platforms, which allow data management API to manage and configure WSNs. The Sensor Planning Service (SPS) only describes the ideas but no real world tests were shown. The OGC SWE standard has challenges such as performance, robustness, and reliability. In [72], SOA provides Sensor Node Management Cloud (SeNoMa-cloud) software, which is extended on a proposed framework in [71]. SeNoMa is designed to manage the WSN configuration. The system deploys nodes in different locations of interest, for example, crop fields, and then assigns a sensor to the nodes, locates login, and transfers periods. The GeoSense system is used as a tool for clients to collect, analyze, and visualize the data. The system has many sensor nodes and base stations and can easily manage a WSN using SeNoMa-cloud by a virtual private network. The development of SeNoMa-cloud has to be suitable with OGC SWE. The OGC SWE has one-way communication in which it can only receive data/services from SeNoMa and send it to the cloud. This approach provides advantages for WSN management on multiple stations and deals with raw data. The sensor node management mechanism was designed to manage WSN configuration. This approach is limited because it increases overhead by

using XML-based web service. An increase in the overhead could cause data transmission with low bandwidth. OGC SWE provides mechanisms to detect and determine failure, in order to reconfigure the system so that it can continue execution.

WSNs are widely used in many studies, such as agriculture control applications and natural resources. Different architectures are used in agriculture to provide an efficient platform for making decisions on how to manage crop planning. An Open Geospatial Consortium (OGC) with SWE that provides a direction for semantic standardization of sensor networks is presented in [73]. The components of SWE are SensorML (Sensor Model Language) and an SOS (Sensor Observation Service) [73]; it can be interoperable for processing data online [73]. The SensorML is XML and used to represent different features of a sensors' system. It provides performance characteristics such as accuracy and the capability to describe the sensor system, process models, and connect sensor networks over the internet. The OGC SWE through SOA was implemented by using two distributed sensing systems.

### **3.2.5 WSN Cloud User Interaction**

The new concept for WSN cloud is designed specifically to apply to a network as a service (NaaS), which provides solutions in large-scale WSNs for Service Orchestrating Architecture provisioning called (WSNs-SOrA). WSNs-SOrA enables WSNs to act as a cloud and is required to support SOA at all WSN tier infrastructure. The SOA enables another system to provide WSN infrastructure based on their needs, while allowing multi-systems to use the WSN. The service provisioning is done using XML [74]. This approach is one of the first state-of-the-art protocols proposing to combine WSNs with cloud

computing [75]. In [76], methods that use sensor data by cloud users are presented. It designs service stacks, interfaces, and repositories based on SOA. The services allow communication between the cloud, WSNs, and the consumer. This architecture supports setup for WSNs which can collaborate, share data efficiently and easily determine the sensed data behavior. The issues of this WSNs setup is addressed through isolated sensor networks and non-collaborative approaches. The isolated sensor network drawbacks are solved by using one registry for sensor networks, and the challenges of non-collaborative approaches are addressed by designing a service stack. The heterogeneity issue is addressed by using SOA.

### **3.2.6 Other Approaches**

Recently, SOA has gained a lot of attention for providing flexibility in the designing of WSN applications. In [74], a method of service selection with flexible Service-Oriented Network Architecture (FSONA) addresses the issues of WSNs. These issues are increasing because of the lack of interoperability and the addition of new services or adaptation new protocols between the sensors and communication architecture. Addressing these issues provides a general communication between users, developers, and applications. In this architecture, a common platform connects the heterogeneous and homogeneous services [77].

Most of the existing routing protocol studies exploit SOA in WSNs. In [78], the path vacant ratio is used to find a group of disjointed paths from available ones and link them. The load balance and congestion control algorithms are used to check and control the load on multipath. The Threshold Sharing Algorithm (TSA) has the ability to divide

each packet into many segments before transmitting it to the destination over the multipath based on path vacant ratio [78].

A secure and adaptive load-balancing multipath routing protocol based on AODV called Service-Oriented Multipath AODV [78]. The benefit of applying AODV protocol is to extend the load balance algorithm due to its routing protocol efficiency, without generating any congestion. SM-AODV provides secure data transmission and improves data confidentiality in Service-Oriented WSNs [78]. The features of multipath routing protocol include a secure transmission of data, independent applications, adaptive congestion control, and extensibility [78].

Another Service-Oriented approach supports QoS and real-time in Industrial Systems [79]. The SOA philosophies can be applied in the enterprise IT and the sensor network itself [79]. The enterprise IT system integrates into the sensor nodes by linking the Service Descriptions (SD). The linked data of the SD and RDF (Resource Description Format) addresses the problem generated through integrated enterprise IT system with sensor nodes [79]. The sensor nodes interact with different service descriptions connected to other service descriptions by the Unified Service Description Language (USDL) method. The corresponding interfaces and the service description are located on/off the sensor or on both, which can lower cost reducing data on the sensor [80].

The flexible architecture is introduced in [81] for sensor networks based on web services and web mashup [81]. Web services build based on SOA. The data is provided through sensor nodes, and service is provided through WSNs for client applications and provided services, such as sensor nodes, to generate raw data. The raw data is processed



and generated by different analyses, filters, complex processes, and web mashup, which provides value-added services. This architecture is adaptive SOA for designing WSNs. The services consist of the abstraction that can be used for developing WSNs applications. XML is used for representation and exchanging data between applications and the network. The WSN is integrated with the mashup, which is used to build different applications on top of the virtual ecosystem of services [81]. SOAP and HTTP modules manage communications. The SOAP should be presented in web mashup and sink nodes, with HTTP module in sensor nodes [81].

Additionally, SOA is applied in business applications. The SOA and mashup have allowed the enterprise to transfer complex applications through integrating the information over internal and external sources. It enables the user to take heterogeneous data from different sources. Therefore, it provides graphical tools called “enterprise mashup” for business users to select, integrate, and analyze data as needed. The approach addresses the collection of accurate and real-time information to satisfy business requirements based on enterprise location and the structure of the data [82].

Moreover, there are various concrete implementations of SOA approaches. A multi-SOA approach is designed to increase the efficiency and QoS of the system [34]. The WSN-SOA, a multi-level based on the existing SOA on the higher tiers with a protocol stack is presented in [34]. The SOA has the capability to handle the nodes with low capacity without generating an overhead of XML-based technology. WSN-SOA allows the SOA-based communication of low capacity sensors in the networks as MICAz motes. The multi-level via auto-configuration can enable all sensors to turn into reusable resources and allow

the distributed collaboration between them. The “software stacks” help link between low capacity and full capacity nodes [34]. The extension of WSN-SOA stacks is introduced in [34]. It supports dynamic deployment of Service-Oriented cooperative tasks in the networks efficiently. The WSN-SOA is implemented on open source operating system TinyOS 2.1 (TinyOS Alliance) and develops WSN-SOA for Crossbow MICAz (Crossbow Technology, Inc., Milpitas, California, USA) [35].

Similarly, the x-SOA approach [83] is related to previous approaches. There is X-SOA framework for sensor web service discovery mechanism, which is based on the Natural Language Query Processing (NLQP) by using semantic grammar [83]. The framework acts as the intermediate layer, called RPQ (Request Parser & Query generator), which supports interoperability between the service requester and the service registry [83]. A novel algorithm called Sensor Web Registry Services Discovery (SWRSD) is used in all steps of the processes of sensor service discovery [83]. The different layers can interact with each other by Unified Modeling Language (UML) sequence diagrams. The limitation of this architecture considers only the QoS function but does not deal with QoS non-functional. The non-functional is known to provide efficiency to the sensor web registry. In [84], the authors used the same mechanism and added QoS non-functional to the sensor web registry. Multi-layers of SOA framework are proposed for Sensor Web Service Discovery (SWSD) mechanisms that are based on the Natural Language Query Processing (NLQP) [84]. The architecture reduces the burden of novice requesters. The overhead decreases by converting user requests in XML or SOAP to other formats. The architecture has fewer capabilities for dealing with other QoS or for supporting different types of sensor

web services. The limitation of this approach is that it tests only five sensor nodes and should be evaluated with additional sensors to obtain more QoS parameters. The power consumption, data aggregation, and delay should be considered with this approach.

The studies [85, 86] proposed a generic framework approach based on web service which can be built as a standardized interface between external networks, applications, and WSNs. The implementation is based on Direct Service-Oriented Diffusion (DSOD) and the Service-Oriented Routing Protocol for WSN [85, 86]. The SOA is implemented on the sensors. The security services are addressed in this architecture and provide Authentication, Authorization, and Accounting (AAA) mechanisms. The drawback of this approach is that accuracy is not considered. The name-centric service architecture framework based on the data/Content-Centric Network (CCN) for cyber physical system (CPS) can address the limitation provided by using transparent methods for accessing the services in the CPS. It implements a lightweight approach for WSNs which is called Content-Centric Networking Protocol for WSN (CCN-WSN) and can easily implement a gateway between CCN-WSNs and CCNx to build the SOA [87]. This approach still has limitations due to the named services required when coordinating naming in CPSs. This drawback should be addressed by using standard naming system for the CPSs.

The NanoSD is a service discovery protocol which designed for mobile, dynamic, and heterogeneous of WSNs [88]. The implementation of NanoSD provides a lightweight service discovery protocol for WSNs [88]. This implementation meets the requirements of service discovery, such as supporting mobility and dynamics in the network, running on heterogeneity nodes platforms, adapting to software modified/changed, and being flexible

and easy to maintain. The heterogeneities of WSNs are supported in this architecture by providing a gateway library. The NanoSD protocol reduces packet size and communication overhead which can provide fast processing. The developer has the ability to select proper routing for WSNs and applications of the routing protocol [88].

The WSNs and SOA approaches are integrated for Intelligent Transportation Systems (ITS), which can obtain the best results for safety and security in its applications. This integration is useful to develop several ITS applications [89].

In addition, a WSN based on SOA with web service is used to detect collision, such as vehicles with motorway guardrails. The simulation applied to determine the propagation wave on guardrails uses the Finite Element Method (FEM) in real-time. This system improved the reliability of collision detections, reduced cost, and is easy to maintain [90]. This approach has packet collide limitation. Due to the receiver node being received, information from multi-sensors are transmitted at the same time.

### **3.3 Service Composition for WSNs**

In this section we introduce an overview of Service-Oriented computing in sensor networks and ad hoc. Most approaches focus on SOM architectures and service composition still under research. In the next section, we discuss some approaches based on service composition for WSNs. The service composition is a design principle applied within the SOA, which is composing a massive service by combining many small services. The service composition is a method that combines and coordinates the aggregate of service and processes service entities into high-levels of application. For example, a

controller service application requires the design service to control the other service. The service composition is responsible for allocating all required service to the service provider. The performance load balance, resource and end to end delay are studied well in service composition.

### **3.3.1 Service Composition with Persistent Queries (SCPQ)**

The service composition can reduce the total number of solutions over the lifetime of persistent queries. Reduction in this number can decrease the total cost of service composition [91]. Routing in WSNs is used only to find a path from the source sensors to the receiver node. Thus, Service-Oriented query routing protocols are applied in order to guarantee a path from the source sensors to the sink and should also include service providers [91]. Two algorithms are applied to minimize energy consumption, which can provide service composition solutions for a persistent query. These algorithms are called Greedy and Dynamic Programming. The Greedy algorithm is applied to minimize the total number of service composition solutions during the lifetime of a persistent query. The Dynamic Programming algorithm uses the results of the Greedy algorithm to find a shorter path and reduce the total cost of service composition solutions. The time complexity of the Dynamic Programming algorithm is defined as  $O((D/T)^3)$  [91]. Similarly, another study uses the Greedy algorithm to select the best nodes. The middleware system service-based approach for WSNs provides QoS and context-awareness [92].

### **3.3.2 Service Centric Wireless Sensors Networks (SWSNs)**

Flexible solutions are necessary to properly handle complex issues that arise within heterogeneity data and devices. SOA has the ability to control these types of data. The work presented in [93], the integration of the Extended WSNs and RFID tags within a web service, is called EWSN nodes. The framework is used to collect and share data from RFID and WSNs as shown in Figure 3.7a. The studies propose the integration of EWSN schemes into the IoT as shown in Figure 3.7b. The EWSN has challenges during the application phases in real-time. It cannot handle different operations and heterogeneities in the system or sensors and has difficulty executing the data. These challenges are addressed by applying SOA and EWSN to the service centric WSNs. This is referred to as intelligent SWSN nodes. Once a web service is applied to EWSN, any interoperability that existed between different applications, heterogeneities or dynamic systems is remedied. The Electronic Product Code (EPC) acts in the network as a mechanism that can process the data of the WSN and RFID. The EPC with SOA provides an easy way to integrate WSNs with RFID tags for IoT applications without the above-mentioned issues.

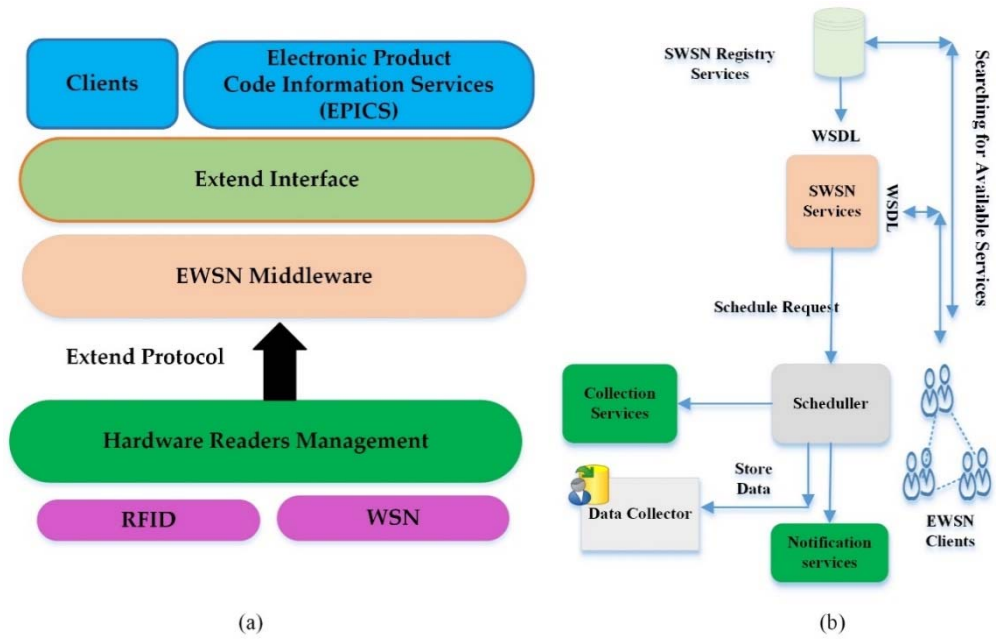


Figure 3.7. (a) EWSN Sensor-based Architecture and (b) SWSN Dynamic Service Platform.

## **CHAPTER 4: LIMITATIONS OF TRADITIONAL MIDDLEWARE**

Most of the existing approaches on middleware architectures and SOA for various WSN applications are highlighted. The proposed approaches attempted to address most of the WSNs challenges and are classified in three types. First, the approaches that applied different middleware architecture to achieve well-designed architecture for WSNs. Second, approaches that attempted to implement SOA for WSN without applying the middleware into the design. Third, an overview of the management and the service composition of some approaches that have remained relatively unexplored are shown.

### **4.1 Middleware Approaches for WSNs**

In our best knowledge, numerous middleware architectures for WSNs have been specifically designed to address the complexity issues that are related to resources and optimization of the pervasive technology. These approaches were aimed towards tackling the open issues that were previously identified in WSNs. None of the reviewed state-of-the-art approaches fulfil every requirement of the WSNs, as shown in Tables 4.1- 4.3. The heterogeneities between sensor hardware and communication devices in large-scale WSN applications have difficulty executing data from different networks. The data/service aggregation aims to minimize energy consumption and network load on the sensor networks by optimizing the transmission data based on time and battery life. Some approaches do not provide any mechanisms that are independent of the middleware; instead, they depend on particular operating systems. The ESOA framework is built on LiteOS while MiSense is built over TinyOS. The support for heterogeneous multi-service



composition highlights the enhancement of service interworking and provisioning to end-users, enabling service orchestration, and discovery at the middleware level. However, these mechanisms are only provided in USEME, OASIS, and ESOA approaches. On the other hand, the security mechanisms have been taken into account through different SOM architectures approaches like SOMM, ESOA, and SAWM. Data or service aggregation is supported in approaches like OASiS, MiSense, SensorsMW, and ESOA. However, most of these approaches do not provide specific implementation and mechanism details. In Table 4.1, a summary of Service-Oriented Middleware architectures are presented. These approaches are regarding the open issues in wireless sensor networks that identified previously. Table 4.2 highlights the representative SOM architectures for WSNs with the evaluation of its advantages and disadvantages. The implementation of these approaches offers relative limitations and strengths. Finally, the requirements and benefits of using SOM for WSNs are shown in Table 4.3.

Table 4.1 Comparative Analysis of Middleware Architectures for WSNs.

<b>Middleware Architecture</b>	<b>Platform Type</b>	<b>Operating System/Platform Independence</b>	<b>Software Applications and Communication Model</b>	<b>Data/Service Aggregation</b>	<b>Heterogeneity</b>
<b>USEME [44, 45]</b>	WSANs	Independent with in-network middleware	Abstract programming language (APL)	Not Supported	Not Supported
<b>OASIS [32, 33]</b>	WSNs	Independent with in-network middleware (middleware is implemented on Mica2 mote hardware Platform)	Application development based on the separation of concerns (SoC)	Supported	Supported
<b>MiSense [39]</b>	WSNs	Built on top of TinyOS operating system	Programming Interface and Services Extensions	Supported	Not Supported
<b>SOMDM [46]</b>	WSNs	Independent with in-network middleware	Implemented based on Ambient Programming Model with the ported code	Not Supported	Not Supported

			in GALs by using Tiny GALs given by TinyOS		
<b>TinySOA [51]</b>	WSNs	Independent with in-network middleware	Not Supported	Not Supported	Not Supported
<b>SensorsMW [48]</b>	WSNs	Independent with in-network middleware	Not Supported	Supported	Not Available
<b>SAWM [50]</b>	WSNs	Middleware for WMSNs	Infra-red cameras are applied to decrease the power consumption	Not Supported	Supported
<b>Mob-WS [47]</b>	WSN	Independent with in-network middleware	XML for the messages instead of using any transport protocols	Not Supported	Not Available
<b>SOMM [49]</b>	Distributed Enterprise systems	Independent with in-network middleware	Programming tasks based on byte-code	Not Supported	Supported
<b>ESOA [52]</b>	WSN	Built on top of LiteOS operating system	Not Supported	Supported	Supported
<b>Middleware Architecture</b>		<b>Multi-Service Composition</b>	<b>Services</b>		
<b>USEME [44, 45]</b>		Supported	<ol style="list-style-type: none"> <li>1. Configuration and Routing Protocol</li> <li>2. Publication and Discovery [44, 45]</li> <li>3. Command and Event Invocation and Communication [44, 45]</li> <li>4. Real-Time Constraints [44, 45]</li> <li>5. Group and Event Management</li> </ol>		
<b>OASIS [32, 33]</b>		Supported	<ol style="list-style-type: none"> <li>1. Node Manager [32, 33]</li> <li>2. Service Discovery Protocol and Composer [32, 33]</li> <li>3. Object Manager [32, 33]</li> <li>4. GALSC queue ports [32, 33]</li> </ol>		
<b>MiSense [39]</b>		Not Supported	<ol style="list-style-type: none"> <li>1. Event detection</li> <li>2. Data aggregation</li> <li>3. Topology management</li> <li>4. Routing</li> </ol>		
<b>SOMDM [46]</b>		Not Supported	Not Available		
<b>TinySOA [51]</b>		Not Supported	<ol style="list-style-type: none"> <li>1. Discovery</li> <li>2. Sensor Reading</li> <li>3. Internal and External Services</li> <li>4. Network and Events Registries</li> </ol>		
<b>SensorsMW [48]</b>		Not Supported	<ol style="list-style-type: none"> <li>1. Data measurement</li> <li>2. Network maintenance</li> <li>3. Event notification</li> </ol>		
<b>SAWM [50]</b>		Not Supported	Not Available		
<b>Mob-WS [47]</b>		Not Supported	Not Available		
<b>SOMM [49]</b>		Not Supported	<ol style="list-style-type: none"> <li>1. service registry</li> <li>2. several servers</li> </ol>		
<b>ESOA [52]</b>		Supported	<ol style="list-style-type: none"> <li>1. Coordination and Service Discovery</li> <li>2. Performance, Monitoring and QoS</li> </ol>		

Table 4.2. Advantages and Disadvantages of Middleware Architectures for WSNs.

<b>Middleware Architecture</b>	<b>The Features and Advantages</b>	<b>Disadvantages</b>
<b>USEME [44, 45]</b>	<ol style="list-style-type: none"> <li>1. Deals with the changes in the web service (WS)</li> <li>2. Supports a set of real-time management constraints</li> <li>3. Allows the programmers to use the programming task of the wireless sensor and actors network (WSAN) easily</li> </ol>	<ol style="list-style-type: none"> <li>1. Not considered security and hardware resources management</li> <li>2. Not support any mechanism to handle a large of data and high communication loads efficiently</li> <li>3. Not supports interoperability with various systems and devices</li> <li>4. Not provides a secure communication/execution</li> <li>5. Cannot integrates with other systems</li> <li>6. Not supports interoperability with various systems and devices</li> </ol>
<b>OASIS [32, 33]</b>	<ol style="list-style-type: none"> <li>1. Development of environment based on separation of concerns</li> <li>2. Supports the node management</li> <li>3. QoS</li> <li>4. Dynamic service discovery</li> <li>5. Failure detection</li> </ol>	<ol style="list-style-type: none"> <li>1. Not provides a secure communication/execution</li> <li>2. Cannot integrates with other systems</li> <li>3. Not supports self-organization mechanisms</li> <li>4. Not supports interoperability with various systems and devices</li> </ol>
<b>MiSense [39]</b>	<ol style="list-style-type: none"> <li>1. Content based publish/subscribe service</li> <li>2. Provide programming API</li> <li>3. Supports data management</li> </ol>	<ol style="list-style-type: none"> <li>1. Not support configurable services</li> <li>2. Not supports self-organization</li> <li>3. Not provides a secure communication/execution</li> <li>4. Not support QoS</li> <li>5. Increase power consumption and processing time</li> </ol>
<b>SOMDM [46]</b>	<ol style="list-style-type: none"> <li>1. Decreased the data processing load by using multi-component architecture</li> <li>2. Supports DBMS</li> <li>3. Notification and data filtering techniques</li> <li>4. Handle a large of data and high communication loads efficiently</li> </ol>	<ol style="list-style-type: none"> <li>1. Not support configurable services</li> <li>2. Not supports self-organization</li> <li>3. Not provides a secure communication/execution</li> <li>4. Not support QoS</li> </ol>
<b>TinySOA[51]</b>	<ol style="list-style-type: none"> <li>1. It provides web service for internet Apps to access WSN</li> <li>2. Supports multiple programming language</li> </ol>	<ol style="list-style-type: none"> <li>1. Not support configurable services</li> <li>2. Not supports self-organization</li> <li>3. Not provides a secure communication/execution</li> <li>4. Not support QoS</li> </ol>
<b>SensorsMW[48]</b>	<ol style="list-style-type: none"> <li>1. The QoS configuration is provided by service level</li> <li>2. Providing mechanism for the application to manage WSNs</li> </ol>	<ol style="list-style-type: none"> <li>1. Not supports self-organization</li> <li>2. Not provides a secure communication/execution</li> </ol>
<b>Mob-WS [47]</b>	Increases the scalability	<ol style="list-style-type: none"> <li>1. Not provides a secure communication/execution</li> <li>2. Not support QoS</li> </ol>

<b>SOMM [49]</b>	<ol style="list-style-type: none"> <li>1. Supports multimedia transmission</li> <li>2. Ability to reduce the cost of development applications</li> <li>3. Supports scalability and</li> <li>4. Supports network level heterogeneity</li> </ol>	<ol style="list-style-type: none"> <li>1. Overhead</li> <li>2. Not support any mechanism to handle a large of data and high communication loads efficiently</li> <li>3. Not very easy to use due to its implementation that used a comprises byte code</li> </ol>
<b>SAWM [50]</b>	Provides secure architecture and modifiable	Not provides a secure communication
<b>ESOA [52]</b>	<ol style="list-style-type: none"> <li>1. Allows users to develop new applications through mix-and-match services without any programming efforts by developers</li> <li>2. Supports the heterogeneous of WSNs and execute various applications on multi-platforms</li> <li>3. It can integrate with other systems</li> <li>4. Provides a secure communication and execution through QoS composition</li> </ol>	<ol style="list-style-type: none"> <li>1. Not provides mechanism to handle a data collection of user to the services</li> <li>2. Not applied in real time</li> </ol>

Table 4.3. The Requirements and Benefits of Using Middleware Architectures for WSNs.

<b>Middleware Architecture</b>	<b>The Requirements</b>	<b>The Goals</b>
<b>USEME [44, 45]</b>	<ol style="list-style-type: none"> <li>1. The configurable service</li> <li>2. Auto discovery techniques of the service providers</li> <li>3. Middleware allows the application executing and running in the network in secure way and easier to update anytime</li> <li>4. Dealing with a large amount of data and increase communication load efficiently</li> <li>5. The consumer service supported to detect and use register service</li> </ol>	Middleware provide general-services such as configuration, invocation, and communication managements
<b>OASIS [32, 33]</b>	<ol style="list-style-type: none"> <li>1. The heterogeneity of underlying environments is hidden by applying abstraction such as protocols and languages</li> <li>2. The consumer service supported to detect and use register service</li> <li>3. Runtime is supported for the service provider to deploy services</li> <li>4. Support QoS</li> <li>5. Dealing with large amount of data and increase the communication load efficiently</li> </ol>	Minimize the resource requirements
<b>MiSense [39]</b>	<ol style="list-style-type: none"> <li>1. The heterogeneity of underlying environments is hidden by applying abstraction such as protocols and languages</li> <li>2. The consumer service supported to detect and use register service</li> <li>3. Runtime is supported for the service provider to deploy services</li> <li>4. Dealing with a large amount of data and increase communication load efficiently</li> <li>5. Interoperability with different device or system</li> </ol>	<ol style="list-style-type: none"> <li>1. Data Aggregation</li> <li>2. Events detection</li> <li>3. Resource and Topology management</li> </ol>

	6. has flexibility to access its services by the high level interface	
<b>SOMDM [46]</b>	<ol style="list-style-type: none"> <li>1. The heterogeneity of underlying environments is hidden by applying abstraction such as protocols and languages</li> <li>2. Interoperability with different device or system</li> <li>3. Dealing with a large amount of data and increase communication load efficiently</li> <li>4. low overhead</li> <li>5. data filter mechanism</li> </ol>	<ol style="list-style-type: none"> <li>1. Allow sensor to handle data from ambient aware sensor networks</li> <li>2. Reduce data processing loads by using multi-component architecture</li> </ol>
<b>TinySOA [51]</b>	<ol style="list-style-type: none"> <li>1. The heterogeneity of underlying environments is hidden by applying abstraction such as protocols and languages</li> <li>2. The consumer service supported to detect and use register service</li> <li>3. Can integrates with other system</li> </ol>	<ol style="list-style-type: none"> <li>1. Discovery data readings</li> <li>2. Actuators management, and network communications</li> </ol>
<b>SensorsMW [48]</b>	<ol style="list-style-type: none"> <li>1. The heterogeneity of underlying environments is hidden by applying abstraction such as protocols and languages</li> <li>2. Configurable services</li> <li>3. Can integrates with other system</li> <li>4. Dealing with a large amount of data and increase communication load efficiently</li> <li>5. Interoperability with different device or system</li> <li>6. Support requirement for QoS</li> </ol>	<ol style="list-style-type: none"> <li>1. Supports dynamic management of heterogeneous data</li> <li>2. Provides QoS configuration by service level</li> </ol>
<b>Mob-WS [47]</b>	<ol style="list-style-type: none"> <li>1. Used as back end resources to reduce the complex</li> <li>2. Asynchronous services</li> </ol>	Provides management and representation of wireless networks
<b>SOMM [49]</b>	<ol style="list-style-type: none"> <li>1. Support Multimedia</li> <li>2. Support QoS, Virtual machine (VM), Mobile Agents, and Tuple space</li> <li>3. provides highly scalable platform by using SOA</li> <li>4. Energy efficiency is increased for the application modification</li> <li>5. The Mobile Agents and Code Repositories are used to enable the Node to be reprogrammed</li> <li>6. Handle heterogeneous nodes with different capabilities</li> </ol>	<ol style="list-style-type: none"> <li>1. Provides Security</li> <li>2. Hardware resource management</li> <li>3. Supports QoS</li> </ol>
<b>SAWM [50]</b>	<ol style="list-style-type: none"> <li>1. The architecture is easy to update</li> <li>2. used less memory for processing the programming codes</li> <li>3. processed in real-time</li> <li>4. Provide low cost during the transmission</li> <li>5. decrease power consumption</li> </ol>	Provides secure architecture
<b>ESOA [52]</b>	<ol style="list-style-type: none"> <li>1. Support requirement for QoS</li> <li>2. Interoperability with different device or system</li> </ol>	Coordination, Monitoring, Conformance, QoS and Service Discovery

## 4.2 Service-Oriented Architectures for WSNs

The SOA comprises of diverse notions, concepts, and technologies from a wide range of studies. Tables 4.4-4.6 show the comparative analysis of service-oriented

architectures for WSNs. In this paper, state-of-the-art approaches based on SOA design for WSN are presented. Even though most well-known examples of SOA are web services, it is important to know that it is not limited to it. The biggest issue of applied traditional SOA into WSNs is that those are built on different platforms/operating systems (OS) without the use of middleware. The approach is considered to support general core functionalities independent of the platform and OS. None of these approaches supported the multi-service composition except for the FSONA approach. Table 4.4 shows the approaches that applied traditional SOA into WSNs that do not support middleware architectures. Some of these approaches provide general architecture with some limitations as shown in Table 4.5. In Table 4.6, the requirements and benefits of traditional SOA for WSNs.

Table 4.4. Comparative Analysis of Service-Oriented Architectures for WSNs.

<b>SOA Approaches</b>	<b>Operating System/Platform Independence</b>	<b>Type of Software Applications</b>	<b>Multi-Service Composition</b>
SODA for Smart Environment [72]	Mulle Sensor Platform (resource constrained sensor platform)	Built upon the gSOAP toolkit with TCP/IP stack-lwIP	Not Supported
SOA Model for Sensor Networks [73]	Not Supported	Built on different applications such as Agent Register, Resource Manager, and Multi-gateway	Not Supported
WSNs Cloud User Interaction [74-76]	<ol style="list-style-type: none"> <li>1. SOrA uses different platforms as TelosB and SunSPOT and acts as Node Network Tier [74]</li> <li>2. Stack of Services, Interfaces and Repositories[75, 76]</li> </ol>	Done by XML	Not Supported
FSONA [77]	Not Supported	Developed with Java Platform	Supported
Healthcare Approaches	<ol style="list-style-type: none"> <li>1. SYLPH [63]</li> <li>2. Wireless Body Sensor Networks (WBSNs) [64]</li> <li>3. SunShine [65]</li> </ol>	Built on ambient intelligence (AI) [63] Java (JDK 1.6, Apache tomcat server 6.0.) [64] and Java EE5 platform of NetBeans [65]	Supported
OGC-SWE standards (Web Service)	<p><b>WASP</b> has two sides</p> <ol style="list-style-type: none"> <li>1. ZigBee enables nodes communicate hop by hop with each other</li> </ol>	Built smart home system uses the SWE standard	None

	2. Software service using HTTPS protocol [69, 70] <b>SeNoMa-Cloud [71, 72]</b> A MQTT broker, ActiveMQ Apollo SensorML [73]		
Configuration Service[94]	Middleware Framework	Evaluation in CORE and EMANE	Not Available

Table 4.5. Advantages and Disadvantages of SOA for WSNs.

<b>SOA Approaches</b>	<b>The Features and Advantages</b>	<b>Disadvantages</b>
SODA for smart environment [72]	<ol style="list-style-type: none"> <li>1. Support the Security, and heterogeneities at low level</li> <li>2. Not required additional middleware</li> <li>3. transmission time is reduced and battery life is increased by using Sensor data aggregation</li> </ol>	<ol style="list-style-type: none"> <li>1. Performance overhead communication while processing of SOAP messages but not as much as messages transmission</li> <li>2. Performance measurement effect on latency</li> <li>3. SOAP-based web services are required parse verbose XML documents</li> </ol>
SOA Model for Sensor Networks [73]	<ol style="list-style-type: none"> <li>1. Provide an efficient architecture</li> <li>2. Secure communication protocol</li> <li>3. Efficiently collecting data from WSNs</li> </ol>	Does not test in real time
WSNs Cloud User Interaction [74-76]	<ol style="list-style-type: none"> <li>1. WSN-SOrA and SOA have solutions and the ability to support infrastructure reuse [74]</li> <li>2. Enable data sharing in efficiently [75, 76]</li> </ol>	Overhead
FSONA [77]	Process heterogeneous wireless mobile networking. Costs are reduced	Overhead
SYLPH [63] WBSNs [64] SunShine [65]	<p>provides a flexible distribution of resources SYLPH and capable during performance time to add new component [63]</p> <p>Decreases memory space, interoperability of service, maintenance cost, fast response time, high privacy, and throughput. This technique was improved the QoS to make decision and time warning generation the authentication mechanism and lightweight and efficient biosensor [64]</p> <p>Collecting and managing then analyzing data [65]</p> <p>Cost reduces [65]</p> <p>It modify the requirement of monitoring [65]</p>	<p>SYPLH is that it has not been tested in real time [63]</p> <p>Framework has overhead due to the use of XML and SOAP in the system [64]</p> <p>Not support Security [65]</p>

OGC-SWE standards (Web Service)	<p>WASP</p> <p>It process the raw data from WSNs [69, 70]</p> <p><b>SeNoMa-cloud</b> [71, 72]</p> <ol style="list-style-type: none"> <li>1. WSN and SeNoMa-Cloud Services communicate with each other by using MQTT broker and ActiveMQ Apollo</li> <li>2. Small packet handles by using MQTT protocol</li> <li>3. Deals with raw data [63, 64]</li> </ol> <p>SensorML</p> <ol style="list-style-type: none"> <li>1. Provide Accuracy</li> <li>2. Ability to describe the sensor system</li> </ol>	<ol style="list-style-type: none"> <li>1. WASP</li> </ol> <p>Not provides mechanism of how WASP with GIS web service is handling large heterogeneous data in real time [69, 70].</p> <p>It provides mechanisms to detect and determine failure [71, 72].</p> <p>Overhead by using XML based web service [73].</p>
---------------------------------	--	---

Table 4.6. The Requirements and Benefits of Applied SOA for WSNs.

SOA Approach	The Requirements
<b>SODA for Smart Environment</b> [72]	<ol style="list-style-type: none"> <li>1. Support the heterogeneity</li> <li>2. Performance measurement effect on latency. The overhead that is related to SOAP message process was small when compared to messages transmission</li> </ol>
<b>SOA Model for Sensor Networks</b> [73]	<ol style="list-style-type: none"> <li>1. Multi-gateway uses to solve the issue of congestion that generate by using one gateway</li> <li>2. Authentication user</li> <li>3. Data should be located near the users and filter data near to distention</li> <li>4. Ability to linked various protocols that can be used for WSN</li> </ol>
<b>WSNs Cloud User Interaction</b>	<ol style="list-style-type: none"> <li>1. NaaS requires the WSN supporting Service-Oriented software architecture</li> <li>2. Non-collaborative[75, 76]</li> </ol>
<b>FSONA</b> [77]	<ol style="list-style-type: none"> <li>1. Interoperability between service</li> <li>2. Supports QoS and run time</li> <li>3. Integrated with other system</li> <li>4. Service abstraction and discovery</li> </ol>
<b>SYLPH</b> [63]	<ol style="list-style-type: none"> <li>1. The devices are not requiring any features as large memory to communicate with SYLPH</li> <li>2. Improves the system security and efficiency for care services</li> </ol>
<b>OGC-SWE standards (Web Service)</b> [69, 70]	<ol style="list-style-type: none"> <li>1. SWE standard helps to discovery sensors data and the interoperability</li> <li>2. Supporting the data detection</li> <li>3. Data retrieval increase for WSN through remote control</li> <li>4. Provide user authorized</li> <li>5. SWE standard helps to discovery sensors data and the interoperability</li> <li>6. Supporting the data detection</li> </ol>
<b>ANDSF</b>	<ol style="list-style-type: none"> <li>1. Solved problem the overhead between access networks and the service registry</li> <li>2. Provide mechanism for updating network states information in real time and service description</li> </ol>
<b>Healthcare Approaches</b>	<ol style="list-style-type: none"> <li>1. Supports efficient information retrieval</li> <li>2. Achieve the desired QoS in WSNs</li> <li>3. Support the heterogeneous and asynchronous</li> </ol>
<b>Configuration Service</b> [94]	<ol style="list-style-type: none"> <li>1. Adaptation at Runtime</li> <li>2. Reduce cost</li> </ol>



Security challenges and performance of data aggregation are not supported in most of approaches while only SODA and SYLPH approaches support security at a low level. In conclusion of this analysis, it is fair to comment that none of the reviewed approaches accomplishes all the requirements globally. The scalability, security, QoS, data aggregation, integration, and overhead limitations should be taken into account during the implementation processes of future designs.

### **4.3 Service Composition Architectures for WSNs**

Open issues of service composition show that the adaptive service composition is required to have flexible composition methods that can enhance the scalability when the services are integrated into the network based on their availability. The SCPQ provides QoS requirements and decreases cost and power consumption. On the other hand, SWSN is capable of collecting information and reusing resources. The SCPQ approach does not address service composition languages on its design. In case of adaptive service composition, SWSN is based on web services. Meanwhile, SCPQ focuses on specific methodology such as service composition solution that is provided through the greedy optimal algorithm. However, SCPQ does not address service integration with the IoT, while the SWSN addresses this issue through web service. Table 4.7 shows the analysis of service composition architectures for WSNs.

Table 4.7. Analysis of Service Composition Architectures for WSNs.

<b>SOA Approaches</b>	<b>Service Composition Programming</b>	<b>Active Service Composition</b>	<b>Services Integrated with IoT</b>	<b>Advantages</b>	<b>Disadvantages</b>
<b>SCPQ [91, 92]</b>	Not Supported	Service based on Greedy algorithm	Not Supported	<ol style="list-style-type: none"> <li>1. QoS and context-awareness</li> <li>2. Minimizes Cost and energy consumption</li> </ol>	None
<b>Intelligent SWSN Middleware [93]</b>	Proprietary semantic annotations for WSDL and XML	Semantic Web Services	Interoperability using WS-specifications	Collects information through the nodes can be reusable resources in the real world	<ol style="list-style-type: none"> <li>1. Data redundancy</li> <li>2. Network dynamics</li> <li>3. Energy balancing and Traffic congestion problem</li> </ol>

In conclusion of the conducted analyses, Tables 4.1–4.7 represent middleware architectures, SOA, and services composition approaches with their requirements and evaluation of their advantages and disadvantages. The implementation of these approaches offers relative limitations and strengths. These approaches are reinforced through the abstraction level, sensors platform, extensibility, and reconfiguration. In this paper, the disadvantages of implementing a comprehensive framework and its limitations are considered. The main limitations that must be addressed are the heterogeneity of sensors networks, end-to-end security from the sensor to end users, QoS (solved through scalability and privacy), response time, and throughput. The service discovery mechanism should be available to assure the continuity of the service. It has the ability to discover any failures and replace them with the best available service during runtime. Since our framework deals with massive data, the communication efficiency should be increased with minimum cost, minimum overhead, and minimum energy consumption. The extensibility that can

facilitate the inclusion of new networks or delete them without re-implementing the entire architecture should be taken into account.

#### **4.4 Discussion**

A number of research studies attempted to achieve the role of Service-Oriented software designs for network embedded system, but they only considered the software engineering aspect of it. The underlying computational platforms, such as SANET, and their limitations have not been considered. For security, none of the proposed approaches provide a comprehensive framework for different services or data secure architecture. The main issues with those approaches relate to the lack of consideration for accuracy in the architecture and data/service aggregation.

The middleware addresses the methods of publish/subscribe, virtual machine, database, and modular/macro programming. However, these solutions provide limited flexibility and interoperability based on interaction between end-users and high-level applications (clients).

Most middleware architectures for WSNs are based on heterogeneous services. These services impact the response time and network efficiency. There are different mechanisms and protocols to improve the efficiency of the services as well as the response time. Middleware architecture deals with massive amounts of messages and events from various services that share those messages and events between the components of the system. In this case, the system must have the reliability to guarantee that the messages run

correctly. The event management technique is used to increase reliability and QoS in WSNs. The QoS has the capability to decrease faults in communication as well as congestion. The QoS mechanisms can be selected from the best available network according to the QoS requirements and contract negotiations based on SLA [48].

There are several SOA protocols used in various architecture such as SOAP, WSDL, and DPWS. These protocols have addressed many challenges such as performance, overhead, exchange data, and security. DPWS used XML for data representation which represents slight limitation on the performance and increase overhead [95]. The web service has two types of protocol [96]: Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). The REST is an architectural-style application that can access resources/data. The SOAP is an XML-based message protocol which can wrap the business logic. The REST is better throughout and its response time is faster than SOAP. SOAP is used for message communication over SOA [92]. The description and discovery services are a web service description language (WSDL) and universal description discovery and integration (UDDI) [92]. These protocols are based on XML to share data between various computing systems. In order to keep the overhead low, these services use HTTP instead of SOAP for its implementation. In addition, DPWS-based web service is used in the architecture for the cooperation, abstraction, and device orchestration of services. In [97], DPWS uses different web service protocols to enable data exchange between data centric WSNs and other IP networks [97]. This approach uses a Service-Oriented Framework based on the DPWS gateway, which easily provides interconnection

between IP networks and data centric WSNs and supports load balance and fault tolerance by using many gateway nodes for one WSN [97].

DPWS is based on middleware that can easily increase the overhead due to power consumption and latency [76]. Furthermore, it provides a secure service process through authorized parties, message integrity, and confidentiality. The DPWS is suitable for devices from certain regions. The DPWS cannot handle the overhead generated through web service, hence an efficient SOA implementation is used. Due to the overhead of SOAP and HTTP protocols, DPWS can be used. DPWS has the capability to secure services, since most of the applications do not require confidentiality for sensor data [76, 97].

Most of the studies have not considered security mechanisms for sending the services/data from providers to the client, which can provide limitations to their approaches. In [46], a unique middleware based on Service-Oriented and message driven architecture for ambient aware sensor networks is presented. This approach does not provide a secure mechanism. Each node in the network should be registered to the main station to ensure security between sensor nodes and their station. The sensor nodes should encrypt their data through secure algorithms before sending it to their neighbors or the main station. Algorithms are needed to avoid any overhead or delay during processing and transmission of data. The QoS should also be taken into account to obtain more accuracy and a faster speed of operations.

In [65], SunShine is integrated with distributed WSNs in the internet to perform a complex task. However, this approach has limitations in sending and updating patient information in a secure manner. The authors do not provide any security method to keep

patients' data secure, especially during the communication between clients and their doctors.

In [98], a novel security mechanism is proposed for WSNs through SOA. In this architecture, the security measurement is used to address the flow of WSNs in a secure manner. The security is applied in the message level of the node, which is located near the cluster head and has the capability to recognize the identity of the sensor through SOA. The main goal of this approach is to reduce power usage and maximize the network's lifetime by decreasing the size of processed information in the sensor nodes [98]. This method has the capabilities to interact, manage, and extend the system easily. The main problem with this approach is that the security is applied only at the message level, not the entire system. Each node should apply an encryption mechanism/algorithm to ensure that all data is generated in a secure manner. The applied algorithm should not impact or increase cost, overhead, or power consumption. The studies in [99] and consider SOM architecture security requirements through a proposed generic framework that handles different security services independently as shown in Figure 4.1. These services support various security functionalities such as secure communications, messages protection, management trust, and access control.

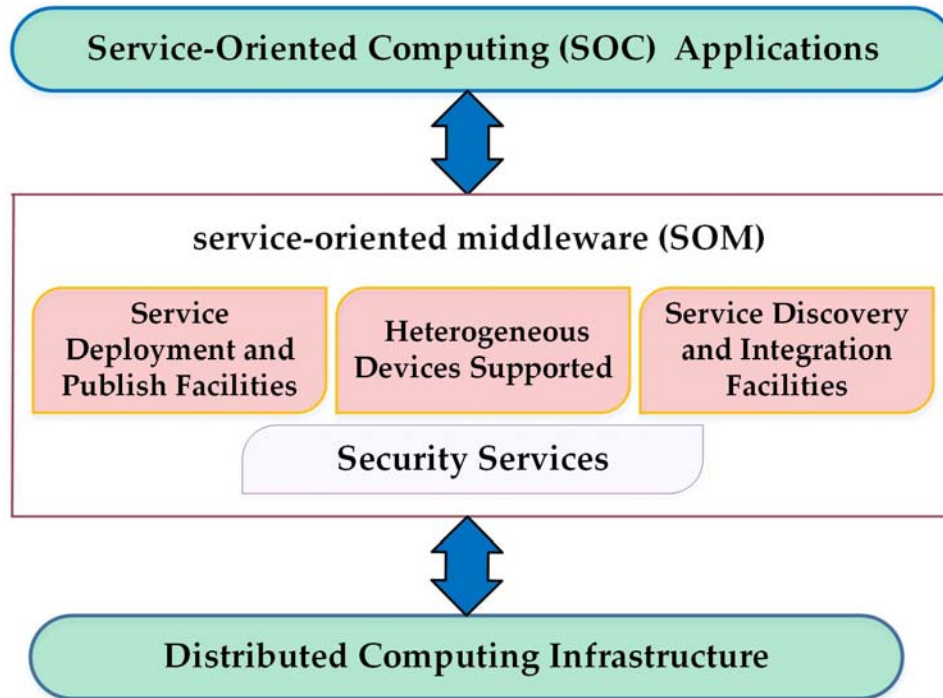


Figure 4.1. Generic Security SOM Architecture Framework

The middleware architectures for WSNs should provide different functionalities that support the system. However, most of the studies on middleware architectures approaches do not provide all functionalities, including:

- 1) Secure executions and communications.
- 2) Deployment of service providers and advertisement.
- 3) Service consumer support to help discover/determine and register these services.
- 4) Support for QoS requirements.
- 5) Support for massive data and high level of communication load efficiently.
- 6) The ability to view the heterogeneities of the underlying WSNs, which are hidden by abstractions.
- 7) The ability to interoperate with multi-devices and systems.

- 8) Client application service transparency.
- 9) The ability to auto-modify and auto-discover mechanisms.
- 10) Configurable services.

Therefore, middleware architectures approaches for WSNs are based on heterogeneous services or devices; the efficiency of these services is impacted due to the response time and network lifetime. The response time of these services should be improved to increase their efficiency through UDP-based SOAP without the need for HTTP [100].

Middleware architecture deals with massive data, messages and event notifications that are generated from different services and shared between different components [100]. In this case, the system reliability should ensure that these messages are delivered on time and are reliable. The reliability and QoS in WSNs are achieved by using event management mechanisms. However, some issues can be addressed by using QoS mechanisms such as congestion and faults communications, which are introduced in the OASIS and SensorsMW approaches. These approaches are developed by through selecting the most suitable available network based on QoS and service level agreements. The middleware has the ability to separate the application logic from the system logic.

## **4.5 Summary**

In this chapter, the representative middleware architectures, SOA, and the services composition approaches with their requirements and evaluation of their advantages and disadvantages are presented in detail. The implementation of these approaches offers relative limitations and strengths. These approaches are reinforced through the abstraction



level, sensors platform, extensibility, and reconfiguration. The main contribution of this paper is design, implementation, and validation of middleware architecture for various applications and environments based on WSN technologies. These requirements enable discovery, improved access, and sharing of the network service and data resources. Moreover, complex services can be achieved through an efficient execution of internetworking services and heterogeneous networks. These features allow the development of sensors based on the services of a third-party network. The analysis of the state-of-the-art middleware architectures foundations in sensor networks shows that most of the issues and challenges, not addressed in published approaches, have been discussed. Therefore, these architectures are designed to consider and address complexities related to the resources of the sensor networks. Most of existing WSN-based middleware architectures do not address scalability and heterogeneous data challenges. The main limitations that must be addressed are the heterogeneity of sensors networks, end-to-end security from the sensor to end users, QoS (solved through scalability and privacy), response time, and throughput. The service discovery mechanism should be available to assure the continuity of the service by discovering any failures and replacing them with the best available service during runtime.

## **CHAPTER 5: DESIGN AND IMPLEMENTATION OF THE SWSNM**

Due to the widespread growth of wireless sensor networks in industrial, healthcare, and military applications, the need for secure data transmission has increased tremendously. Recent literature reports the significance of middleware in WSNs. Unfortunately, many of these approaches do not address security problems, which leads to insecure communication and data transmission. Such data is generally sensitive and needs protection against attacks and possible risks of exposure.

Machine learning algorithms are categorized into supervised, unsupervised, and reinforcement learning [101]. Supervised learning takes place when the data sample (or the training set) is labeled. Machine learning algorithms such as support vector machine (SVM), decision tree (DT), and K-nearest neighbor (K-NN) have successfully addressed several challenges of WSNs such as data aggregation, localization, clustering, energy aware, detection and real-time routing.

The purpose of using an unsupervised learning is to classify data into different groups as clusters and enable them to investigate the similarity between the input samples. Reinforced learning takes place when the results from learning assist in some sort of environment change. Reinforcement learning algorithms control the behavior of the agent (as sensor nodes) within their environments. Based on the rules, the agents in the environment can select the action to transmit it from one state to another [102]. Neural Networks (NNs) are ML models that can solve several challenges and tasks in WSNs such as quality of service (QoS) and security. There is an immense need to boost security to

improve the QoS through NNs, which are comprised of distributed computation nodes as well as WSNs [102].

Machine learning algorithms are used to address non-functional requirements associated with WSNs. However, accuracy problems can be associated with each of the machine learning algorithms. One of them, a non-functional requirement in WSNs, is security. Machine learning algorithms provide solutions to resource constraints that pose a major security challenge in WSNs [103]. The observations in the network can sometimes be misleading due to a number of factors, such as unexpected attacks or intrusions, so it becomes important to detect a particular anomaly through machine learning algorithms and maintain a secure network [101]. When machine learning is applied to WSNs, it helps decrease their vulnerability to misleading information and unwanted attacks. The implementation of ML also drastically increases the reliability of the network by eliminating misleading information and unexpected intrusions. Additionally, ML techniques also increase the lifespan of the WSN by significantly reducing the energy required by the sensor nodes. Moreover, ML also reduces (and strives to eliminate) human intervention.

Literature [104-107] presents a number of machine learning algorithms that address the security problems in WSNs. Janakiram et. al [104] showed the detection of outliers using Bayesian Belief Networks (BBNs). The authors correlated both temporal and spatial data points to identify similar readings in neighboring nodes. These readings are approximated and matched with one another to find possible outliers in the data obtained from sensor nodes. Conditional relationships are built to not only identify outlying data

points, but also to fill in the missing data [104]. Similar to the investigation of k-nearest neighbor presented in [108], Branch et. al [105] developed an outlier detection method within the network using k-nearest neighbor. A major disadvantage, however, of using the k-nearest method is that it requires significant memory space to store data.

Black hole attacks are common in the transmission of data in WSNs. In such attacks, misleading routing reply messages are sent by the nodes whenever route requests are received. These misleading messages result in the termination of the route discovery; real routing reply messages are ignored [101]. Kapalantzis et al. [106] presented a mechanism of detecting similar forwarding attacks using support vector machine (SVM). This mechanism detects black hole attacks by using routing information, bandwidth, and the hop count of the nodes [106]. Rajasegarar et al. [107] were able to combine SVM with the outlier detection scheme to establish a one-class, quarter-sphere SVM anomaly recognition technique [107]. The SVM methods are far superior due to their efficient learning and enhanced performance in non-linear and complex network problems.

We are conducting our research to develop an efficient middleware based on machine learning to address WSNs' security challenges. The proposed middleware is able to secure information and resources from malicious attacks and detect node misbehavior. The special characteristics of WSNs such as power consumption, throughput, and network lifetime are taken into account in this contribution.

We introduce an intelligent security system for WSN middleware based on GANs to improve traditional middleware in terms of security mechanism, handling of heterogeneous characteristics of sensor nodes, and to filter and pass only the real data. To

the best of our knowledge, it is the first time that the GANs algorithm has been used for solving the security problem in WSNs' middleware. Additionally, in the proposed contribution, WSNs' middleware applies a GAN that is capable of filtering and detecting anomalies in the data. The proposed procedure is described in Algorithm 5.1.

Table 5.1. Algorithm for proposed WSNM based on GANs.

---

**Algorithm 5.1 The Proposed WSNM Based on GANs**

---

- 1: **Inputs:** : training set :  $X = \{(x_i, y_i)\}_{i=1}^N$ ,  $N_g$ : sample size is randomly selected from  $X$  for Generator (G) to learn data distribution  
**Inputs:**  $M_F$ : number of fake data will be generated from the G once the training is completed,  $n$ : mini-batch size,  $T_{ts}$  is testing set
- 2: **Outputs:**  $M_F$  samples generated from the  $G$ , *Accuracy*
- 3: Select  $N_g$  samples ( $x$ ) randomly from original data
- 4: **For**  $i=1$  to training iterations **do**
- 5:     **For**  $k$  steps **do**
- 6:         Sample of  $n$  noise samples  $z = \{z_1, z_2, \dots, z_n\}$  from noise  $p_g(z)$
- 7:         select  $n$  samples from original data  $x = \{x_1, x_2, \dots, x_n\}$
- 8:         Concatenate  $x$  and  $z$ . Then, define  $y = [1] * n + [0] * n$
- 9:         Update the Discriminator by descending its stochastic gradient  

$$\nabla_{\theta_d} \frac{1}{2n} \sum_{i=1}^{2n} [\log D(x_i) + \log(1 - D(G(z_i)))]$$
- 10:        Update the Generator by descending its stochastic gradient  

$$\nabla_{\theta_g} \frac{1}{n} \sum_{i=1}^n \log(1 - D(G(z_i)))$$
- 11:     **End for**
- 12: **End for**
- 14: Generate new data ( $T_d$ ) from the Generator after the training is completed.
- 15:  $T_r = \text{Append } x \text{ to } T_d$
- 16:  $T_r = \text{Shuffle}(T_r)$
- 17: Update the Discriminator by descending its stochastic gradient  

$$\nabla_{\theta_d} \frac{1}{n} \sum_{i=1}^n [\log D(x_i)]$$
- 18: Compute the accuracy of the Discriminator based on testing set  $T_{ts}$
- 19: **Return** accuracy

---

## 5.1 Generative Adversarial Networks

GANs, inspired by Ian Goodfellow in 2014 [109], are a class of artificial intelligence and are used in unsupervised ML. GANs contain two networks: the generator (G) network and the discriminator (D) network, as a minimax two-player game [109-111]. The generator network creates fake data similar to the real samples, and the fake data passes through the discriminator network (D) with data from the real distribution as inputs. Figure 5.1 represents the general model of a GANs algorithm. The G network is designed to learn the distribution of the training data, while the D network is designed to calculate the probability of the data originating from the training data (real) rather than the generator data (fake). These networks improved WSNs' performance and optimization during iterative optimization and mutual confrontation. The discriminator improves by extending the target dataset. The generator and discriminator networks must be differentiable during implementation. The proposed GANs provide an efficient way to learn deep representations without extensively explained training data. These networks achieve this by deriving backpropagation results from the competitive process including a pair of networks as shown in Figure 5.2.

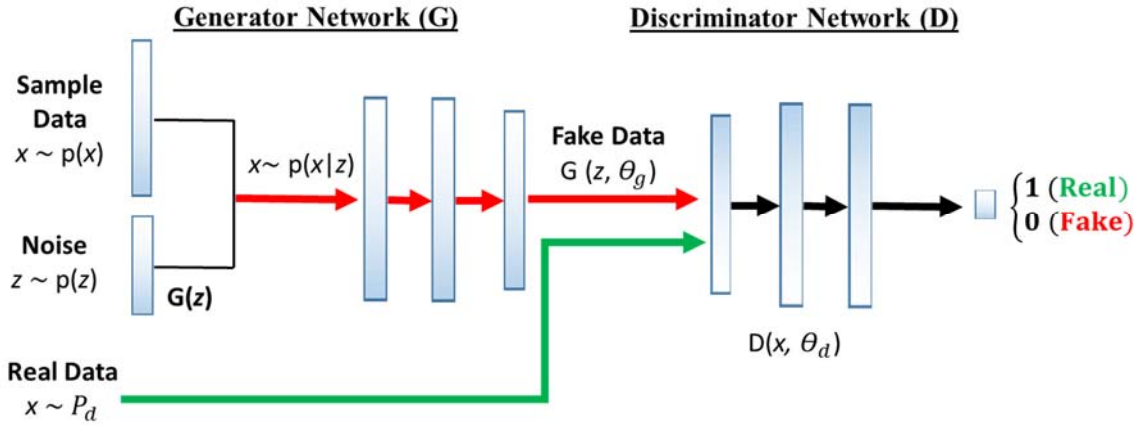


Figure 5.1. The proposed framework for GANs illustrates the sample flow from the generator network (G) to the discriminator network (D).

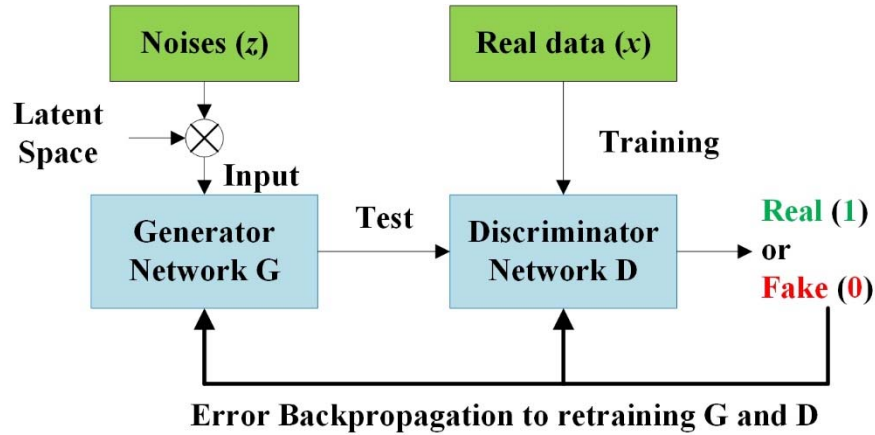


Figure 5.2. Two models which are learned during the training process for a GAN are the discriminator (D) and the generator (G).

The general formula for GANs is shown in equation 5.1. The D takes real data ( $x$ ) and fake data from the generator, represented as  $G(z)$ , and the output is the probability of that data being real ( $p(x)$ ). Thus, the D network is capable of increasing the likelihood of identifying real data and lowering the probability of accepting fake data (from the generator). The G network takes vector random number ( $z$ ) as the input. The first term corresponds to optimizing the probability of the real data ( $x$ ) (close to 1) and the second term corresponds to optimizing the probability of the fake data ( $G(z)$ ) (close to zero) [109-111].

The proposed GANs are based on a minimax game where one agent attempts to maximize the probability while the other attempts to minimize it. G's ability to generate new data that is similar to the real samples is thus improved. The idea is to confuse the attacker and prevent them from differentiating between the new data from the generator and the real data from the sensors and dataset. The D differentiates between real and attack data by maximizing the probability of the real data to 1 and minimizing the probability of fake data (from the G or an attacker) to 0.

$$\min_G \max_{D = \{x \in D\}} \log D(x) + \int_{x \in D} \log(1 - D(G(z))) \quad (5.1)$$

## 5.2 Generator Network

The proposed generator network (G) is used to create various attack data (fake) from one sample (which is acquired from the NSL-KDD dataset). Crucially, generator has no any direct access to the real data (dataset) G learns only through its interactions with D. The discriminator has access to both the real data and the synthetic data drawn from the dataset. From the error backpropagation results, as shown in Figure 5.2, the G uses it to retrain the generator again, leading it towards being able to produce fake data of better quality.

The output of this network range (0, 1) contains the numbers of neurons, where the activation function applied in the last layer of this network is sigmoid. The first layer of the G, the noise input, is fully connected, and this layer is reshaped into a size of (8×5) and



then fed into the convolutional layers. G's architecture is comprised of a fully connected layer and two convolutional layers. This network architecture is illustrated in Figure 5.3.

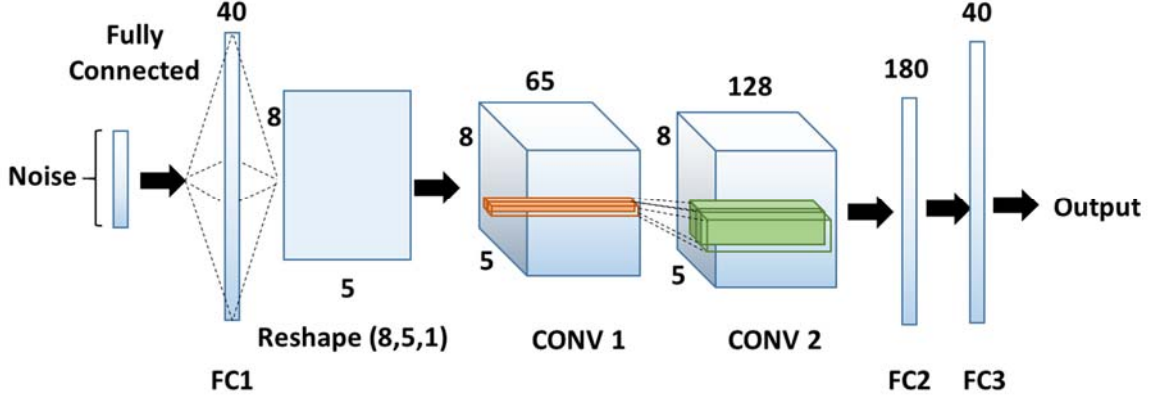


Figure 5.3. The Generator Network Architecture.

The generator network is mapped from a representation space called latent space to space of data. The general formula  $g : g(z) \rightarrow R^{|\mathbf{x}|}$  where  $z \in R^{|\mathbf{z}|}$  is a sample from latent space, where the data is  $x \in R^{|\mathbf{x}|}$  then turns these into multilayer feed forward neural networks with a weight of  $\theta g$ . The proposed G network calculates this with equation 5.2.

The output of the generator is  $g = \{x_i\}_{i=1}^{M_F}$  where  $M_F$  stands for the newly generated fake data from G with random sample data  $Z = \{z_i\}_{i=1}^{M_F}$  as inputs.

$$G = \sum_{o=1}^h \sum_{i=1}^N \beta_o f((\omega_i) + v_o) \quad (5.2)$$

where  $h$  is the number of hidden neural nodes,  $o$  and  $i$  represent the output and input of the hidden layers respectively,  $f$  stands for the activation function in the neural networks,  $\omega_i$

represents the input weights of the  $i$ -th hidden neural nodes,  $\beta_o$  is the output weights, and  $v_o$  represents the threshold values of the  $i$ -th hidden neural nodes.

### 5.3 Discriminator Network

The discriminator D takes both real (authentic) and fake data and aims to differentiate between them. Both the G and D networks are trained simultaneously and in competition with each other. Therefore, the discriminator has access to both the real data and synthetic data drawn from the dataset. The D uses error backpropagation results for 150 iterations as shown in Figure 5.2 to retraining and updated, leading it towards being able to distinguish between real and fake data.

The inputs of the discriminator are  $D = \{x_i\}_{i=1}^N$ , where  $N$  represents the sample number of the dataset. The discriminator is initialized in Keras (TensorFlow) as shows in following equation 5.3:

$$D(x_i) = \sum_{o=1}^h \sum_{i=1}^N \beta_o f(\omega_i^T x_i + v_o) \quad (5.3)$$

where  $h$  is the number of hidden neural nodes,  $o$  and  $i$  represent the output and input of the hidden layers respectively,  $f$  stands for the activation function in the neural networks,  $\omega_i$  represents the input weights of the  $i$ -th hidden neural nodes,  $\beta_o$  is the output weights, and  $v_o$  represents the threshold values of the  $i$ -th hidden neural nodes.

In the training set, the discriminator takes  $g = \{x_i\}_{i=1}^{M_F}$  and  $D = \{x_i\}_{i=1}^N$  as inputs, with the outputs one for real data and zero for fake/attacks data respectively. The discriminator is capable of determining the probability of new generated fake data falling within the interval time; if it does, then the D network accepts it as real data. The G network performs very well in convergence.

The generated fake data (new)  $\mathbf{g}$  and the real dataset  $\mathbf{D}$  will combine and then send the full data to the destination, the base station. The base station then takes the combined data, defined as  $D = \{x_i\}_{i=1}^{N+M_F}$  and feeds into another discriminator to distinguish between the real and fake data, filtering them before transmitting them to user.

The discriminator network D contains multiple-layers that feedforward the neural network with a weight of  $\theta_d$ . The input is a feature vector  $x$ . The D network has the ability to differentiate between real and attack data. The training data for the D network is comprised of real data and malicious (attack) data generated by the generator. The output shows a true interpretation of whether the data is normal or abnormal. Figure 5.4 shows the visualization of the discriminator network's architecture. The first layer of D is the input, the real and fake data from the G network. The last convolutional layer of D is flattened and then fed into a sigmoid function, giving an output in the range of 0 to 1. Batch normalization is used as the input for both the D and G networks, shifting inputs to zero-mean and unit variance. This method helps deal with training issues from poor initialization and supports the gradient flow in deeper models.

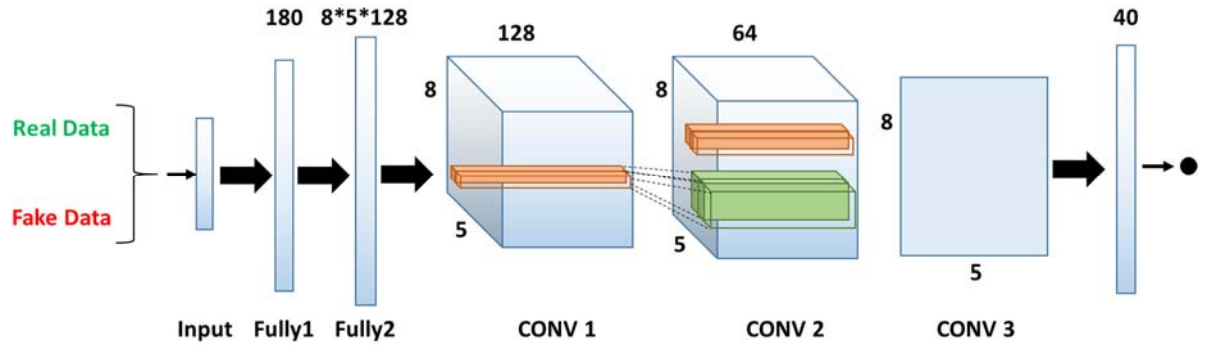


Figure 5.4. The Discriminator Network's Architecture.

## 5.4 Dataset

Many conventional classifiers fail to differentiate between normal and attack traffic. The benchmark NSL-KDD dataset [112] is used to detect any intrusion into the sensors' data in the system. The NSL-KDD dataset contains an imbalance of classes in normal and attack data traffic. The ratio of attack to normal traffic is comparatively low. The phenomenon of normal traffic outweighing the attack traffic is referred to as the Class Imbalance Problem (CIP). This occurs when the minority class, also known as the attack class, exhibits a much lower representation in comparison to that of the majority, or normal, attack classes. The CIP benefits the attack traffic, and the intrusion detection system is unable to withstand it. Therefore, there is a strong need to identify specialized techniques to counteract such an attack by placing an importance on the minority classes.

The proposed approach solves the imbalance problem through the proposed generator model. The main difference between this model and existing algorithms is that the generator creates a balanced data that is more representative of the real data by providing the generator only one feature vector of this dataset. This feature is then used as

feedback in the discriminator, enabling it to distinguish between fake data (corresponding to 0) and real data (corresponding to 1).

In this architecture, we use the publicly available NSL-KDD dataset [112] [113], a refined version of KDDcup99 [114]. NSL-KDD is solved redundant records and duplicate data issues in training set in KDDcup99 dataset [114] [115]. Moreover, this issues affects the performance of evaluate system. The proposed approach is used NSL-KDD dataset for training and testing that is comprised of normal and attacks data. The proposed technique consists of following steps. First, split dataset into training set and testing set, shown in Table 5.1, is made up of 125,973 data samples in the training set and 22,544 samples in the testing set. The testing set is also comprised of additional attacks that are not in the training dataset. This dataset has 41 different features to define each threat as shown in Table 5.2. Second, perform pre-processing the NSL-KDD dataset should be converted to binary, since the neural networks can only process this type of data. Once converted, the dataset can feed into the neural network model as an input layer. Preprocessing this dataset is done by hand, similar to other techniques such as the flag, service, and protocol types, and is converted as a number from 1 to 100. For example, the flag pre-processing technique uses OTH=76 and REJ=77 [8, 116].

Table 5.2. Overview of NSL-KDD Dataset.

	<b>Normal</b>	<b>Attacks</b>	<b>Total</b>
<b>NSL-KDD Train</b>	67343	58630	125973
<b>NSL-KDD Test</b>	9711	12833	22544

## CHAPTER 6: EXPERIMENTAL RESULTS

The proposed framework is implemented in Python and all experiments are performed in the Keras library [117]. Keras is a high-level neural network API and self-contained framework for deep learning. It supports scikit-learn features such as grid search, and cross validation. The framework was evaluated on the NSL-KDD dataset. The analytical model was developed using MATLAB.

### 6.1 Experimental Setting

**Generator Network Setting:** The G is designed with a fully connected layer with 40 neurons. We then reshape the output of the fully connected layer into a size of  $(8 \times 5)$  before feeding it into two convolutional layers. The three layers are fully connected, as shown in Figure 5.3. We employ batch normalization [118] in some layers to normalize the inputs into zero-mean and unit variance to make the learning faster. We train the G model using the stochastic gradient descent (SGD) optimization algorithm with a mini-batch size of 128. The learning rate is set to 0.01 and the momentum at 0.9 for 150 epochs. The hyperbolic tangent (tanh) activation function applies for all layers. We use the L2 norm regularizer to prevent overfitting with a weight decay of 0.001.

**Discriminator Network Setting:** We train the D model using Adam Optimizer with a learning rate of 0.001 with momentum. The mini-batch is 128,  $\beta_1 = 0.5$ , and  $\beta_2 = 0.99$ , which helps stabilize the training. Adam optimizer has shown faster convergence than SGD. We employ dropout with a rate of 0.5 for fully connected layers to combat overfitting. The sigmoid output is a scalar value of the probability of whether data is real

or an attack. For the real data, the scalar output is more than 0.5, and for attacks, the output is less than 0.5.

The weights of all of the layers in G and D networks are initialized according to the Xavier initialization [119] technique and biases are set to zero. The input features of each vector is normalized between -1 and 1.

## **6.2 Convolutional Neural Networks (CNNs)**

There is significant design research on deep Convolutional Neural Networks (CNNs) layers to achieve improved accuracy [120-122]. In [120], the authors performed an experimental study on depth, or the total number of layers in a network. The author kept time constraints constant while only increasing the depth. This practice resulted in an overall performance reduction, having more layers makes the network more difficult to optimize and more prone to overfitting. Moreover, the accuracy becomes either stagnant with increased depth or much reduced. Literature has shown that while the training errors tend to decrease, errors increase with low accuracy after a while [120].

Since deep networks are mostly used for complex data with multiple classes, we used a simple binary classification dataset in our proposed architecture: the number of CNN layers is set to three to obtain a high-performing network. Experimentally, increasing CNN layers leads to inaccuracy while also requiring a higher computational time and cost. A high-performance, optimized architecture is obtained with three CNN layers to maintain accuracy of results while also minimizing overhead and overfitting, as shown in Table 6.1. Increased CNN layers can affect the accuracy and provide a high loss function based on

data generated from the generator network (G). The loss function for the generator is computed by using the feedback from D. We use stochastic gradient descent (SGD) with a learning rate of 0.01 on over 150 training iterations to minimize loss. Table 6.1 shows that the accuracy increases with a minimum number of layers, with the optimum accuracy achieved with three. The results in Figure 5.1 illustrate that the quality of data generated improves by increasing the accuracy and minimizing the loss function of the G. The G network is updated based on the output feedback from the D network until it generates more accurate data that the D accepts as real.

Table 6.1. Accuracy Comparison for Different Layers of CNN Architectures

<b>Number of CNNs Layers</b>	<b>Accuracy</b>
6 or more	82 %
5	84%
4	86%
3	~87%

### **6.3 Confusion Matrix**

The confusion matrix is applied to evaluate the performance and effectiveness of the proposed generator network and the original dataset NSL-KDD. For this purpose, the Accuracy Rate (AR), False Positive Rate (FPR), True Positive Rate (TPR/Recall), and F-measure ( $F_1$ ) are applied and computed by following formulas, numbered 6.1, 6.2, 6.3, and 6.4. TP, TN, FP, and FN are the number of true positive, true negative, false positive and false negative cases, respectively.



$$AR = \frac{TP+TN}{TP+TN+FN+FP} \quad (6.1)$$

$$FPR = \frac{FP}{FP+TN} \quad (6.2)$$

$$TPR/Recall = \frac{TP}{TP+FN} \quad (6.3)$$

$$F_1 = \frac{2(P*R)}{P+R} \quad (6.4)$$

$$P = \frac{TP}{TP+FP} \quad (6.5)$$

$$ErrorRate = 1-AR \quad (6.6)$$

### 6.3.1 Full Feature of NSL-KDD Dataset

In this section, 40 features of NSL-KDD dataset are used to evaluate the performance of the proposed approach. Figure 6.1a shows the confusion matrix between the testing target output and the predicted output for the generated data from the G network. The G network achieved a better binary distribution while also improving the accuracy and decreasing the classification error. In addition, TN and FP are two main criteria for evaluating the performance of the G network data compared with the NSL-KDD dataset results.

Figure 6.1a, shows the confusion matrix between the testing target output and the predicted output for the generated data from the G. The G network achieved a better binary distribution while also improving the accuracy and decreasing the classification error.

Additionally, TN and FP are two main criteria for evaluating the performance of our G network data compared with the NSL-KDD dataset results.

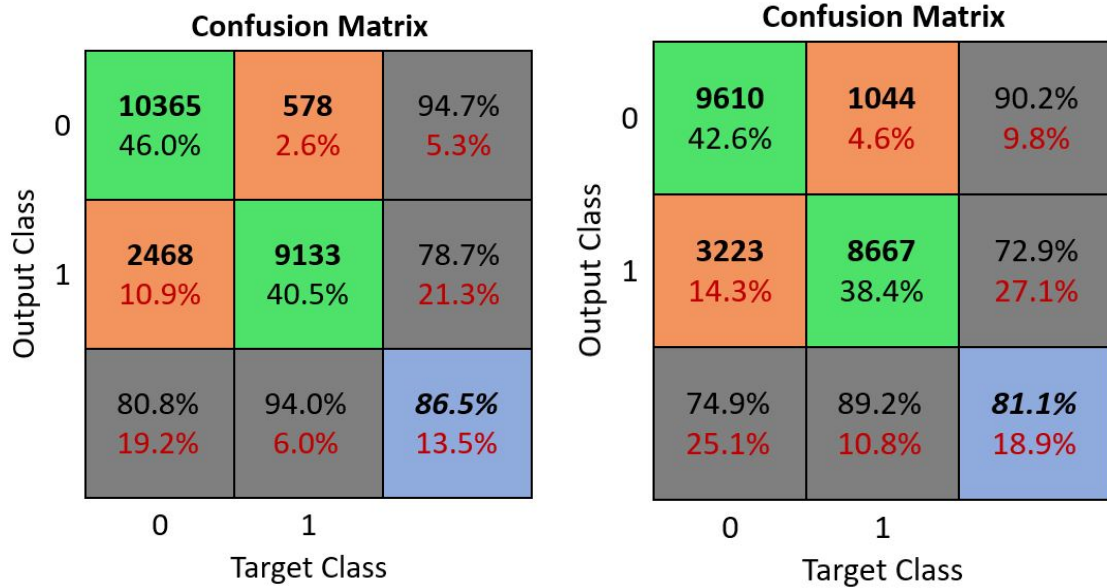


Figure 6.1. (a) Generated data in the proposed Generator Network. (b) Original Dataset (NSL-KDD).

The results show that FP is reduced from 14.3% to 10.9%; it also has lower FPR compared to the original dataset and existing algorithms. The original dataset produced a low accuracy (81.1%) with a high FPR (27.1%) and FP (14.3 %), as shown in Figure 6.1b. Precision (P) is a measure of accuracy achieved in the positive prediction of the class calculated from equation 6.5.

The Recall (R), or TPR, is a measure of whether or not actual observations will be predicted correctly, and is obtained with equation 6.3. The low precision and high recall show that most positive examples are correctly recognized due to a decrease in FN. The F-measure (F1) is the harmonic mean that measures the quality of the classifications between the average of P and R as given in equation 6.4.

The aim is to provide a high level of adversarial system on our generator model, one that is much better than the attack samples and will result in an increase in accuracy and decrease in the error rate. FPR occurs when the results are incorrectly predicted as positive when they are indeed negative, an outcome which is reduced in the proposed model, obtained in equation 6.2. The experimental results show that the proposed generator network gives better accuracy and a robust representation of data with the ability to reduce the error rate from 17.4% to 10.9%.

The Error Rate (ER) can be calculated from the accuracy result. The accuracy is the number of correct classifications divided by the total number of classifications. The ER will be obtained by equation 6.6. The results obtained from the proposed G network were evaluated based on the error rate, FPR and F1, and then compared with the NSL-KDD dataset and Artificial Neural Network (ANN) approach. Table 6.2 shows the limitations of the dataset and the ANN technique due to a high error rate in FPR and low accuracy.

Table 6.2. Comparison of Proposed Approach with Different Approaches

<b>Method</b>	<b>FP</b>	<b>FPR</b>	<b>F-Measure</b>
Original Data [112]	14.3%	27%	81.8%
Artificial Neural Network (ANN) [123]	17.4%	31%	81.6%
<b>SWSNM Approach</b>	<b>10.9%</b>	<b>21%</b>	<b>87.2%</b>

In this section, we provide the results of our method and the discussion. The generator network produced attack samples that were more realistic and accurate than the

original dataset. The intelligent detector model was able to filter and detect between the normal and attack data. The proposed networks were reliable in detecting an attack.

Table 6.3. The Comparisons of Accuracy Rate for Proposed Approach with Existing Approaches on NSL-KDD Dataset

<b>Method</b>	<b>Accuracy</b>
SVM [113]	69.52%
Decision Tree [113]	81.5%
DMNB with RP [124]	81.47%
SOM [125]	75.49%
ANN based IDS [123]	81.2%
<b>SWSNM Approach</b>	<b>86.5%</b>

We compared the performance of our approach alongside existing methods that use the NSL-KDD dataset with 40 features. In Table 6.3, the ML algorithms are simulated to perform this comparison. As shown in Table 4, the proposed model achieves significantly better accuracy with a lower error rate. The performance of ML techniques optimized accuracy over the NSL-KDD dataset. For example, the accuracy of support vector machine (SVM) [113] and decision tree [113] are much lower compared to other ML techniques [113]. Panda *et al.* [124] introduced Discriminative Multinomial parameter learning using Naïve Bayes (DMNB) with a supervised filter called Random Projection at the second level. The authors achieved 81.47% accuracy in their system. Ibrahim *et al.* [125] implemented self-organizing map (SOM) with a very low accuracy rate. The ANN [123] reported that their accuracy was similar to other ML techniques.

### 6.3.2 Features Selection

Feature reduction is applied by using principal component analysis (PCA). The goal of PCA is to select the most significant feature and reduce the dimensionality of the data into 20 features while keeping the variation in the NSL-KDD dataset as much as possible. Figure 6.2a shows that the G network generates 86.4% accurate data with the FPR of 18.5% in comparison to the original dataset with 76.3% accuracy and FPR of 33.8% as shown in Figure 6.2b. The results in Table 6.4 show that FP is reduced from 14.3% to 8.7% of the data generated from G network with 40 features, due to the efficiency of the GAN algorithm.

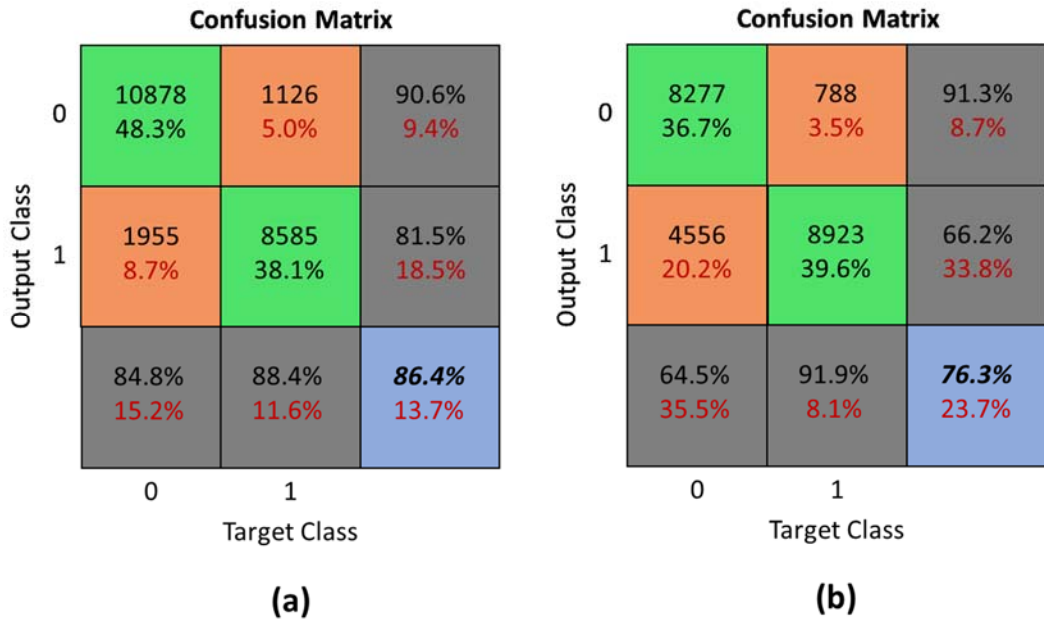


Figure 6.2. (a) Generated data from G Network with 20 features. (b) Original Dataset (NSL-KDD) with 20 features.

Table 6.4. Comparison Results between Proposed G Network with Original Dataset (NSL-KDD) with Only 20 Features.

<b>Method</b>	<b>FPR</b>	<b>FP</b>	<b>F<sub>1</sub></b>
NSL-KDD Dataset	33.8%	20.2%	75.5%
<b>SWSNM Approach</b>	<b>13.1%</b>	<b>4.8%</b>	<b>86.7%</b>

In Table 6.5, different of ML algorithms are simulated to carry out comparative analysis with 20 features. It can be observed that SWSNM produced a much higher accuracy when the selection features are applied. Moreover, in Table 6.5, the F<sub>1</sub> for SWSNM is higher than NSL-KDD dataset, more specifically FPR is reduced from 33.8 % (for the NSL-KDD dataset) to 18.5% (for SWSNM).

Table 6.5. Comparison of Accuracy Rate of SWSNM with other ML method with 20 Features

<b>Method</b>	<b>Accuracy</b>
SVM	78.7%
Decision Tree	81.1%
AdaBoost	77.6%
Original Dataset	76.3%
<b>SWSNM Approach</b>	<b>86.4%</b>

## 6.4 Data Visualization

The T-distribution stochastic neighbor embedding (t-SNE) is a machine learning algorithm used to visualize the structure of very large data [126]. The visualization produced by this algorithm is significantly better on almost all datasets. We used t-SNE to

visualize the output data of our model's results, compared it with the original dataset for both full feature (40 features), and reduced feature (20 features). The aim is to take a set of points in high-dimensional space and find the correct representation of those points in a lower-dimensional space (2D). The t-SNE builds a probability distribution over pairs of high-dimensional data in such a way that similar data have high probability of being selected, while dissimilar have small probability of being selected. It minimizes the divergence between the two distributions. Suppose a given dataset of objects  $x=(x_1,x_2,\dots,x_N)$  in which each point has a very high dimension and function  $d=(x_i,x_j)$  computes a distance between pair of objects then convert it into two-dimensional data  $x_j=(x_1,x_2,\dots,x_N)$ . The similarity of data point  $x_j$  to data point  $x_i$  is the conditional probability  $P(j|i)$ , and  $D$  is the number of data points obtained, as represented in equation 6.7. The t-SNE aims to learn a  $d$ -dimensional map of  $y_i=(y_1,y_2,\dots,y_N)$  that reflects the similarities of  $P_{ij}$ .

$$P_{ij} = \frac{P_{ji} + P_{ij}}{2D} \quad (6.7)$$

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k,m} (1 + \|y_k - y_m\|^2)^{-1}} \quad (6.8)$$

The similarity measure  $q_{ij}$  of two points  $y_i$  and  $y_j$  is defined in equation 6.8. The  $t$ -distribution can withstand outliers and is faster in evaluating data. The original dataset and the data generated from G network contained a high number of dimensions along which the data is distributed. The NSL-KDD dataset, shown in Figure 6.3a and 6.4a reveal poor

visualization in comparison to the data generated through the proposed G network, as evident from Figure 6.3b and 6.4b. The experiment show that G network has produced accurate data and achieved diversity with more coverage of data distribution. The NSL-KDD dataset has poor diversity and less coverage of the data distribution.

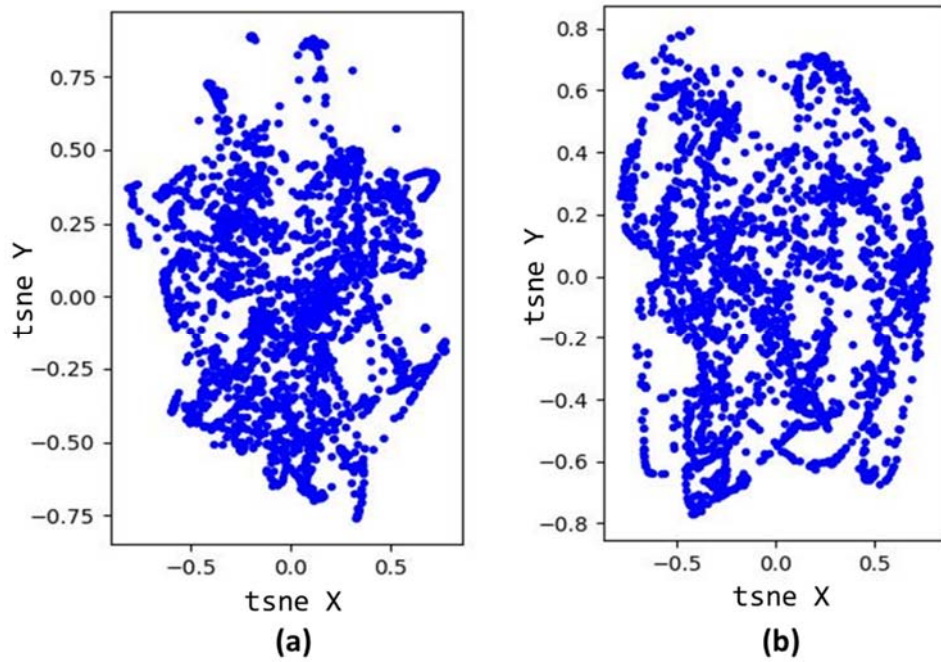


Figure 6.3. t-SNE Visualization with full features. (a) Original Dataset (NSL-KDD) and (b) Generated data in proposed SWSNM.



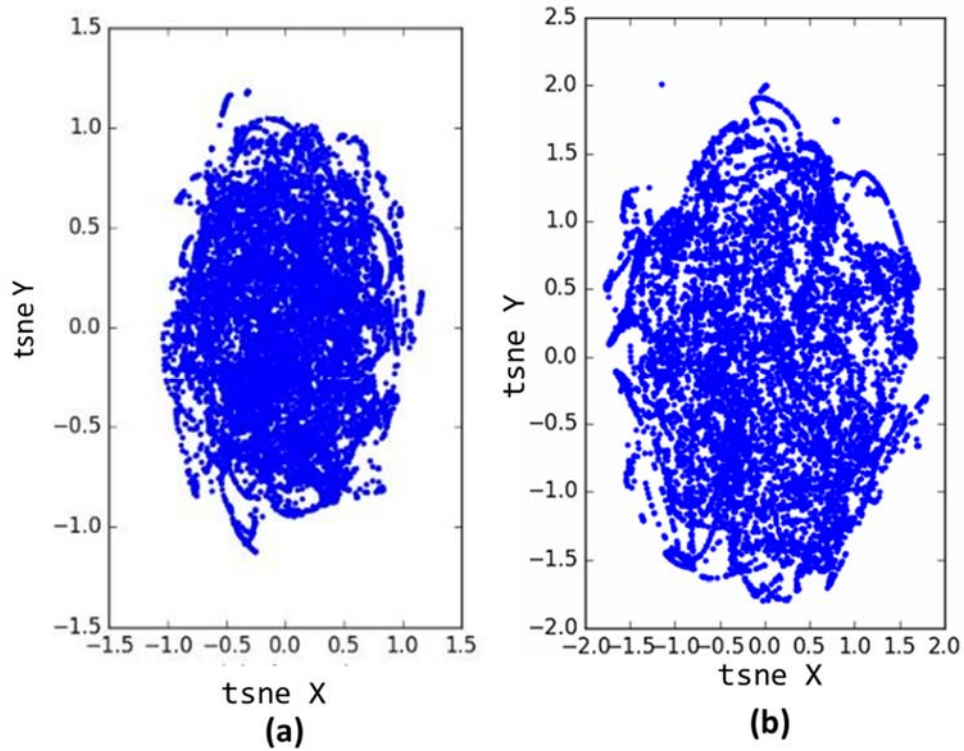


Figure 6.4. t-SNE Visualization with 20 Features. (a) NSL-KDD Dataset and (b) Generated data in proposed SWSNM.

## 6.5 Refeeding the Generated Data

In this section, the generated data with accuracy of 86.5% obtained from the generator network is re-fed to generate new data. The G network is able to generate a better quality data and takes much less time than the first time training. Figure 6.5 shows the confusion matrix results after 150 iterations. It is crucial to consider the FP rate since it represents the cost of learning. The aim is to have a high TP rate (high benefits) and a low

FP rate (low costs). Figure 6.5 shows that the G network is capable of generating accurate data with 85.1% accuracy and much lower FPR of 20.4%.

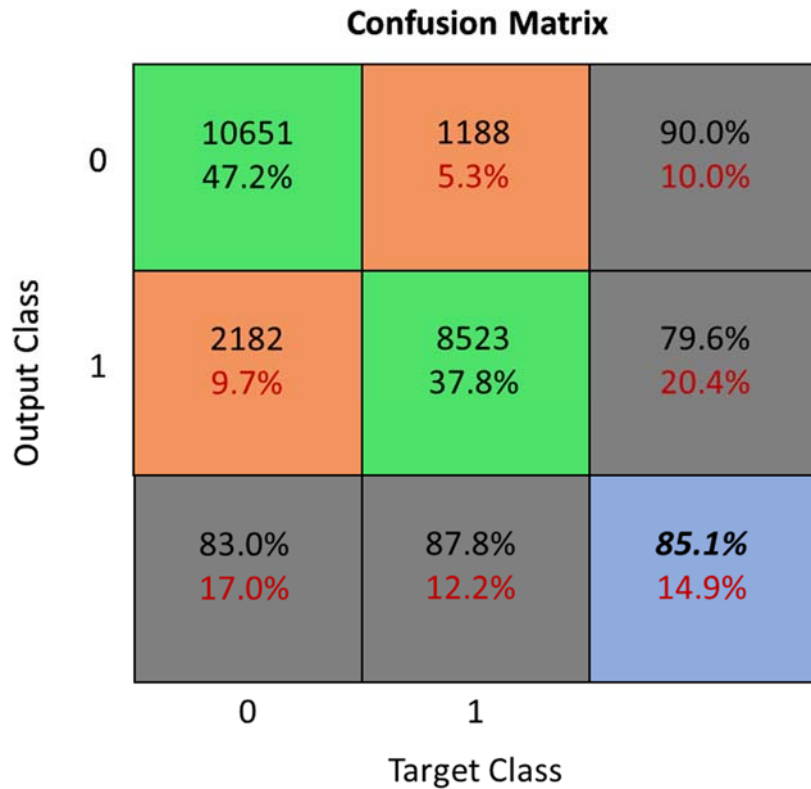


Figure 6.5. Re-feeding the Generated Data into Generator Network.

## 6.6 Evaluation of the SWSNM

We evaluate the capacity of the proposed wireless sensor network middleware (WSNM) based on machine learning for adaptive evolution through a component in middleware called adoption. Adoption allows the addition of new sensor nodes during runtime in a secure manner. Mechanisms such as security, flexibility, and fault-tolerance

must be considered during middleware implementation [18]. Numerous standard algorithms are applied to detect node failure [127, 128]. Peng Jiang [129] proposed a distributed fault detection approach capable of checking node failure through an exchange of data between neighboring nodes within the network. However, this scheme is not suitable for diagnosis or the detection of accuracy with small number of neighboring nodes [129].

Sensor nodes are prone to failure due to energy constraints and environmental factors that frequently affect the network topology. In our contribution, we consider the message freshness mechanism, which ensures that the existing data is new and guarantees that no adversary uses old data (messages). Moreover, new sensors are easily deployed by considering the forward and backward secrecy mechanism [2]. Forward secrecy restricts nodes from failing or leaving the network with future data. Backward secrecy does not allow any node to join the network to read any previous transmitted data [2].

Most existing security algorithms are impractical for WSNs due to the resource constraints in nodes. We applied a unique, unsupervised learning technique on the middleware to secure the entire network. The proposed middleware supports and adapts to node failure and node mobility without affecting the performance of the overall network. We designed a scalable middleware where the network has the capability to grow in size while continuing to meet network's security requirements. Middleware based on machine learning techniques can not only minimize the probability of node failure, but also eliminate the need for a network redesign. The intelligent discriminator (D) is capable of detecting attacks and diagnosing failed nodes and abnormal data. In case of incorrect

readings from nodes, the information is sent to D. The D will consider this reading as faulty and remove the node from the network because it can negatively affect the performance and accuracy of the network.

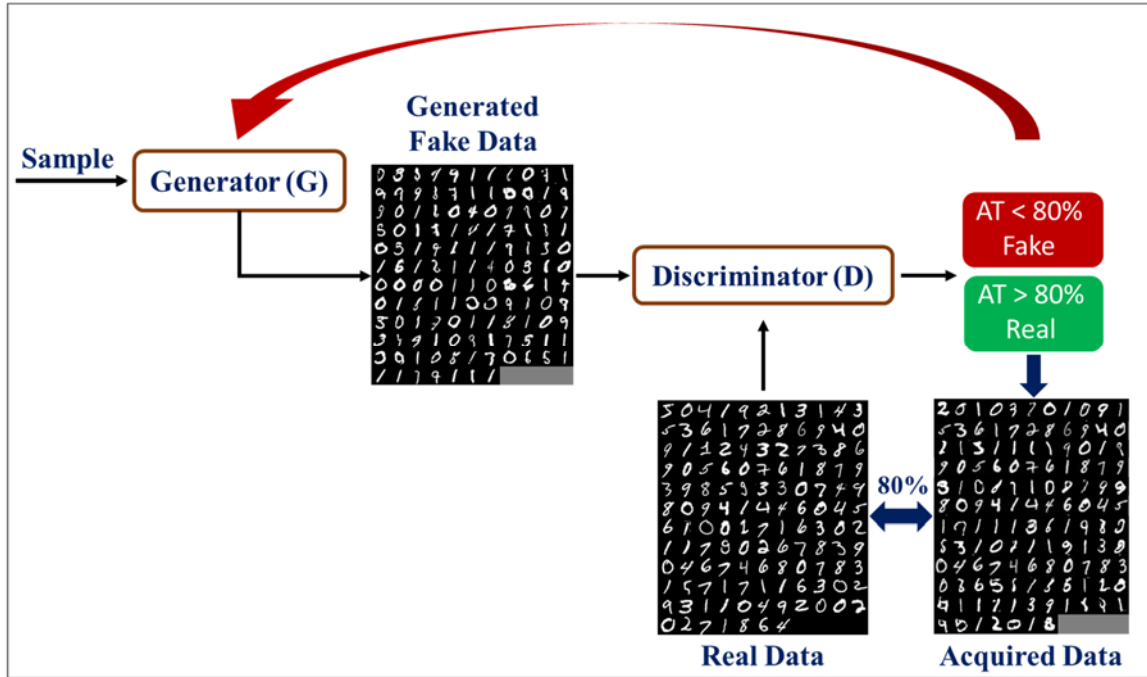


Figure 6.6. Generate Accurate Data Scenario and Detecting Errors for each Iteration.

The proposed architecture has the capability of re-feeding the output data into the generator depending on the accuracy of the results. This is done through a comparison check of the result with the desired data. Empirically, we investigated the proposed architecture by testing the discriminator network (D) on data that came from the generator and contained errors. As a result, we found that the D is capable of rejecting any erroneous data. The MINST dataset [130] is applied to represent the simulation scenarios of the proposed architecture. For example, if the output data from the generator is fake and less than the set accuracy threshold (AT) of 80%, the discriminator network automatically sends

it back to the generator for further iterations, as shown in Figure 6.6. Similarly, if data at each iteration is deterministic but the final data results in error and is not real, the network rejects the final data containing errors and feeds the most recent accurate data to the generator until the obtained result is error-free.

## **CHAPTER 7: EFFICIENT GAN BASED SWSNM**

An intelligent unsupervised learning algorithm for developing secure wireless sensor networks middleware (SWSNM) is introduced. SWSNM provides an efficient, secure communication between the sensor nodes and the base station with minimal power consumption, increased probability of successful data delivery, and an improved network lifetime. The proposed approach eliminates the need to use fake sensor nodes by introducing unsupervised learning algorithms into WSNs.

This approach is capable of addressing the anonymity of real data communication by incorporating real data from the sensor nodes with fake data generated by generator network to confuse the adversary. The main goal of the G network is to generate fake data very similar to the real data, and then combine the fake data into the real data before diffusing it to the base station.

### **7.1 Network Model**

The network is composed of the sensor nodes, the base station, and fake data from the generator network. The nodes are distributed randomly with the same power, resources, and computational capabilities. The nodes collect information about an event and combine their data with fake data before transmitting it to the base station. The fake data that is generated from the generator network should be identical to the real data from the sensor node. The base station has a higher capacity in terms of power and resources than other sensor nodes in the network.

## 7.2 Generating Fake Data

The generative adversarial networks (GANs) algorithm is applied to generate fake data that is identical to real data, to secure the network through the D network[131]. Alshinina and Elleithy provide more details about this technique in[131]; injecting fake data into the real data for each node during the lifetime of the network, instead of using fake nodes to generate dummy data, has a positive impact on energy consumption and network throughput. The real data is hidden within the fake data between which the adversary cannot distinguish. By applying this technique, the data is transmitted to the base station in a secure manner. The discriminator network (D) should be able to distinguish between the real data and fake data and filter it, before sending it to the client or end user.

We evaluate the proposed algorithm by feeding the G network data that can either be normal or attack data. The G network is able to generate fake data and then append it with the real data. The sensor node should do the above step before sending any data to its neighbor or the base station. Finally, the data will pass through D network, as shown in Figure 7.1 [132]. The D network then evaluates and filters the data, both real and fake, even if both sets of data are very similar to each other. After that, only the real data is transmitted to the end user. The diagram and process of GANs based on intelligent WSNM is shown in Figure 7.2.

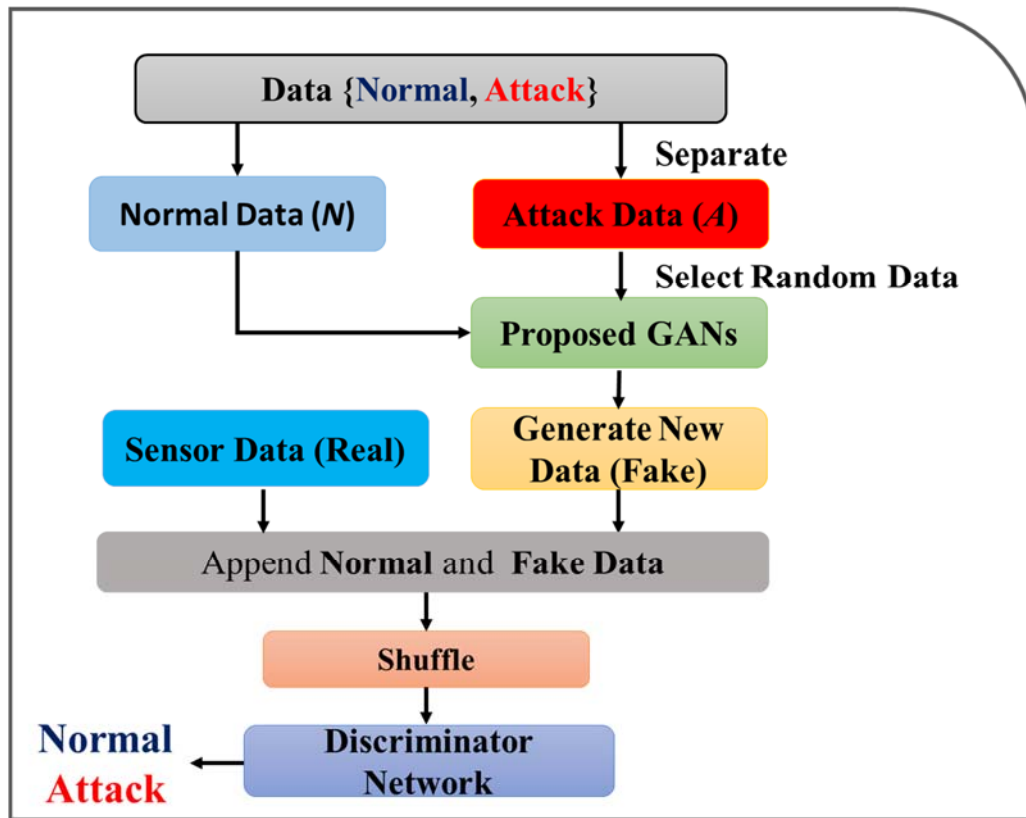


Figure 7.1. The Scenario of Proposed SWSNM Approach



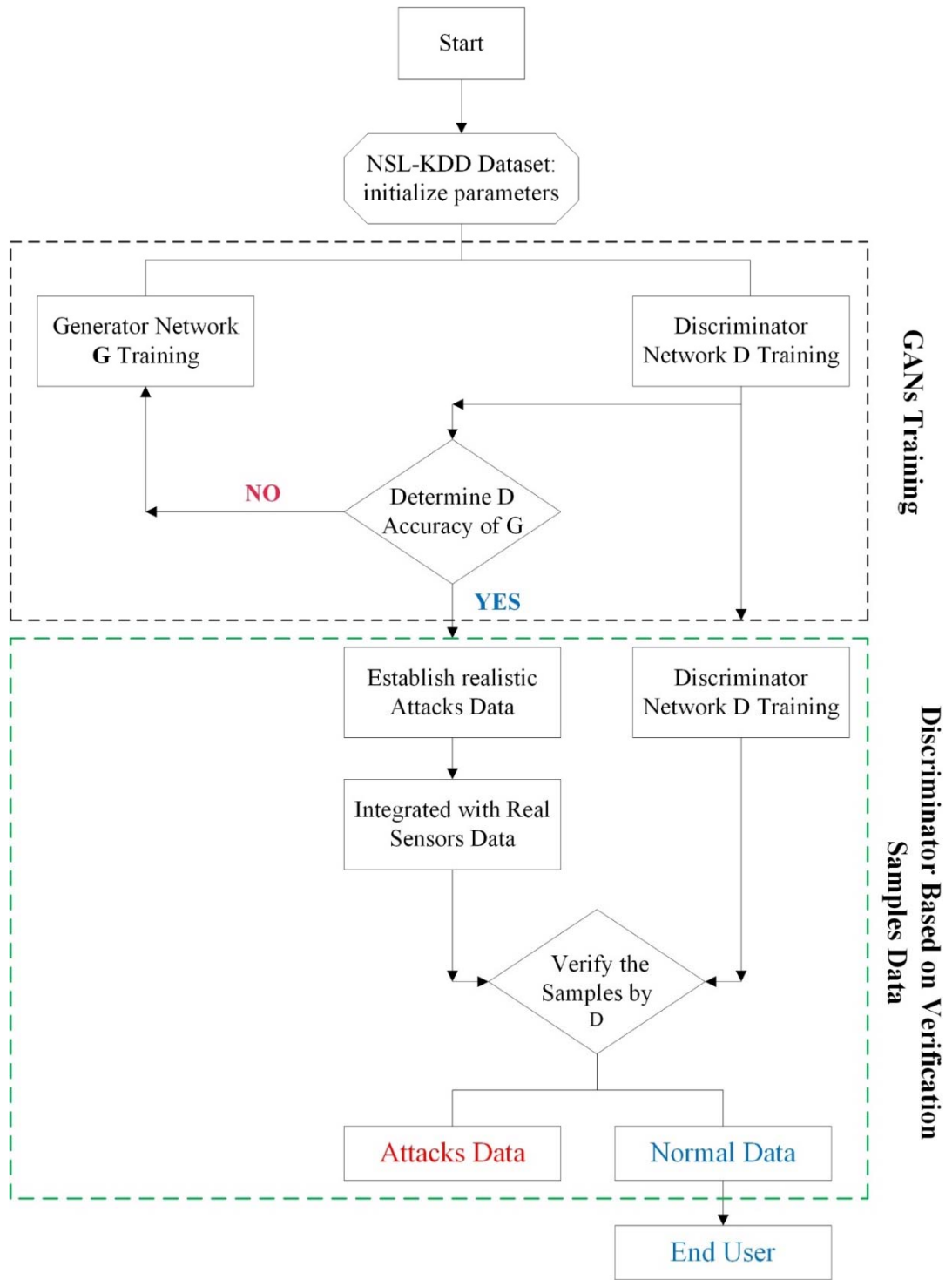


Figure 7.2. Diagram of SWSNM based on GANs.

### 7.3 Simulation Test for Fake Data and SWSNM

In our simulation, the size of the network is  $1500 \times 1500 \text{ m}^2$  using NS2. The WSN involves 150 sensor nodes with a transmission range of 40 meters. The initial energy of the nodes is set to 6 joules. The maximum energy consumption of the sensor nodes for receiving (Rx) and transmitting (Tx) the data is set to 14 mW and 13.0 mW, respectively. Sensing and idle nodes have 10.2 mW and 0.42 mW, respectively. The maximum simulation time is 45 minutes, and the pause time is 20 seconds for phase initialization before starting the simulation. During the testing phase, the GAN takes about 20 seconds to distinguish between real and fake data. Extensive experimental evaluation on this approach ensures that the discriminator network is robustly capable of protecting the network from any attackers or malicious nodes. It improves the security of the network without compromising on the network delay.

The proposed network composed of sensor nodes and base station (BS) is distributed randomly with the same power, resources, and computational capabilities. These nodes collect information about an event and embed their data with fake data from the G network before transmitting it to BS. The BS has a higher capacity in terms of power and resources than other sensor nodes within the network.

The main objective of the simulation is to monitor the network and secure data communication from both internal and external attacks. Extensive experimental evaluation on this approach ensures that the discriminator network is robustly capable of protecting the network from any attackers or malicious nodes. It improves the security of the network without compromising on the network delay [132].

The biggest challenge in WSNs is when the attacker compromises a node by targeting the network resources. For this purpose, the propose SWSNM approach generates fake data identical to the real data from the sensors in the network area, and then joins the real and fake data before sending it to the base station through routers. The network consists of 12 mobile nodes and 138 static nodes. We assume that the nodes that drop all packets passing through them are malicious nodes. When it receives an indication of dropped packets, the algorithm assigns a malicious flag to those nodes. The location of each of the malicious node within the network is calculated and those nodes are replaced with static (normal) nodes [133].

### 7.3.1 Power Consumption

A comparison of SWSNM with and without malicious nodes with Eagilla approach are shown in Figures 7.3, 7.4, and 7.6. Figure 7.3 shows the average amount of energy consumed by the nodes within the network. It is clearly seen that when the malicious nodes are replaced with new static nodes, the energy consumption of the network is reduced.

In the proposed approach, the energy consumed during data transmission as well as during sleep and idle modes are taken into account. The energy consumption is obtained from equation 7.1. We assume that the energy consumed by node  $j$  has bits of packets to transmit/receive while the node is active. Further, both sleep and idle modes are counted and  $n$  is the total number in the network.

$$\sum_{j=1}^n \frac{\text{Total energy consumed at node } j}{n} \quad (7.1)$$

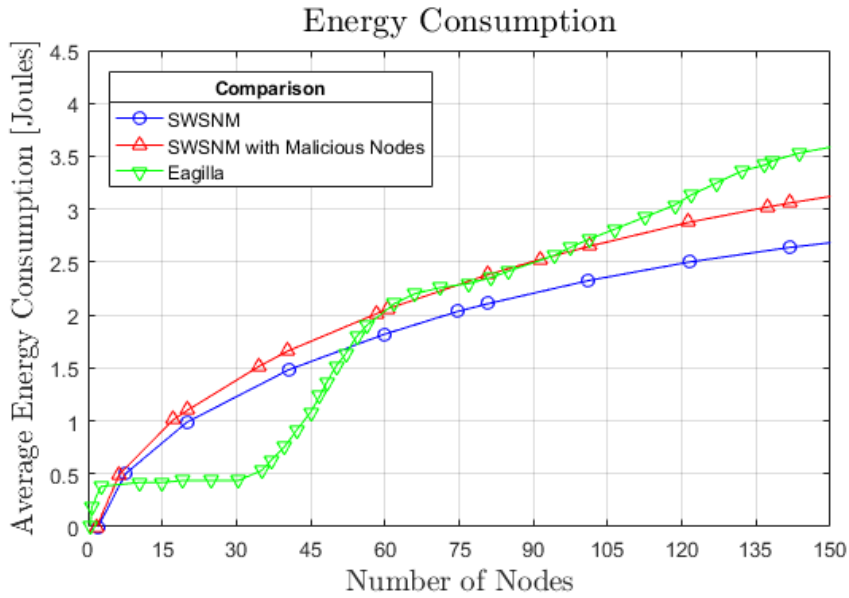


Figure 7.3. Energy consumption for SWSNM with Eagilla Approach

Figure 7.3 shows the average energy consumption for SWSNM (with and without malicious nodes) [133] and the Eagilla approach [44]. It is seen that removal of the malicious nodes reduces the energy consumption of the network. The energy consumption curve for Eagilla [43] is rather interesting. While the energy consumption is much less for small number of nodes (a little over 30 nodes), significant increase in the energy consumption is seen at network size ranging between 30 and 60 nodes. This shows that in terms of the energy consumption, the Eagilla approach [43] is only feasible for small number of nodes.

### 7.3.2 Throughput

Throughput is defined as the amount of data that is transmitted from source nodes to the destination or base station, within a certain time, obtained in 7.2. Figure 7.4 shows a comparison of network throughput for each of the three cases. As seen from Figure 7.4, the throughput of the network without malicious nodes is significantly higher than the network with malicious nodes. The SWSNM outperforms the Eagilla [43] by a significant margin.

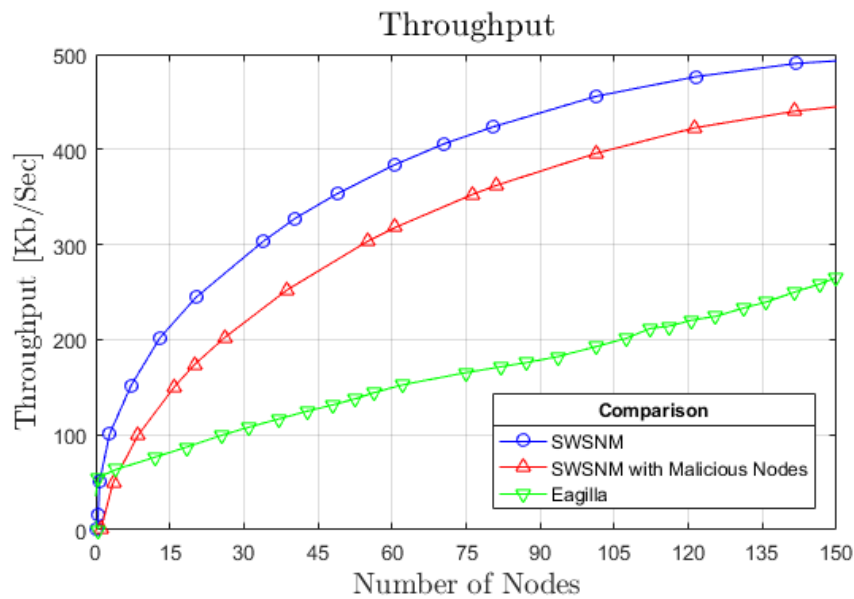


Figure 7.4. Throughput for SWSNM and Eagilla.

$$\text{Throughput} = \frac{\text{Number of bytes received at base station}}{\text{Total number of bytes transmitted at source nodes}} \quad (7.2)$$

### 7.3.3 End-to-End Delay

The end-to-end delay (EED), obtained in equation 7.3, is another important parameter to evaluate the performance of the proposed approach. In equation 7.3, it is noteworthy that the EED is obtained by summing the delays of all the nodes and averaged

over total number of nodes. The delay of each node is calculated through equation 7.4 and normalized by the total number of packets by the given node  $j$ . Figure 7.5 shows that the EED increases until a certain time (~32 minutes) and stays fairly constant after that. It is noteworthy that while the trends are similar, the SWSNM shows significantly lower end-to-end delay when compared to that with the existence of malicious nodes (SWSNM w/10 MN). The end-to-end delay of the Eagilla [43] approach is comparable to the proposed SWSNM. The delay of node  $D_j$  is obtained in equation 7.4; the  $D_{rec}^j$  represents arrival time at the destination for packet  $p$ , and  $D_{snd}^p$  is transmission time at the source node. Where  $n$  is the total number of nodes in the network.

$$EED = \frac{\sum_{j=1}^n D_j}{n} \quad (7.3)$$

$$D_j = \frac{\sum_{p=1}^{pkt} D_{rec}^p - D_{snd}^p}{\text{Number of packets by node}_j} \quad (7.4)$$

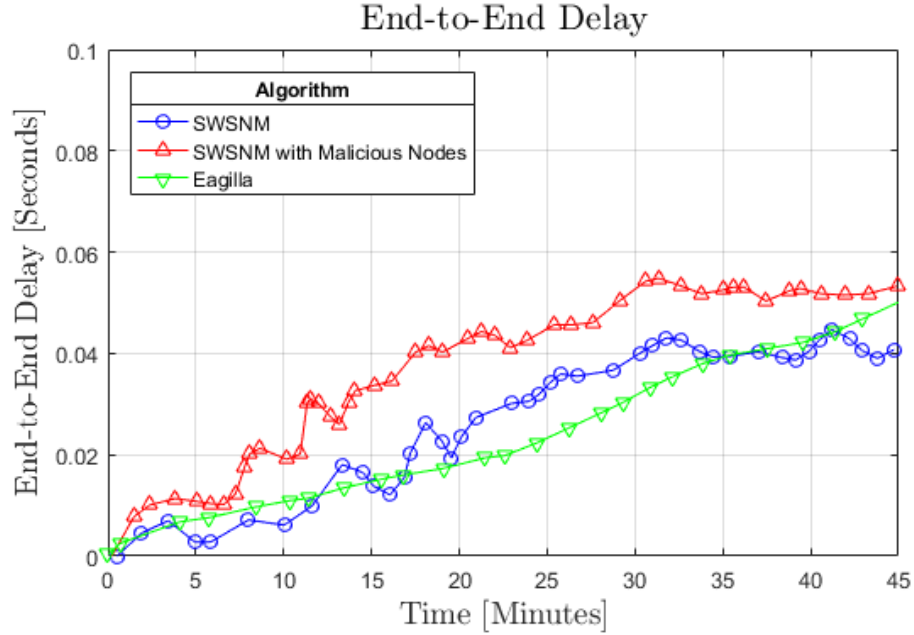


Figure 7.5. End-to-End Delay for SWSNM and Eagilla.

SWSNM comparison of three factors; energy consumption, throughput, and the end-to-end delay, is shown in Tables 7.1 and 7.2 for SWSNM with malicious nodes (MN) and Eagilla [43], respectively. For the ease of comparison, one location or time is selected for each of the variables. Table 7.1 shows that 14.9% more energy is consumed when the network included malicious nodes compared to that with no malicious nodes. Similarly, more than 10% throughput was increased when all malicious nodes were replaced with a 24.5% lower end-to-end delay. It can be inferred that if the probability of malicious nodes is higher in the network (for example 20%), percentage differences in the calculated variables are expected to be much larger. SWSNM comparison with the Eagilla [43] approach, shown in Table 7.2, reveals significant percentage differences for energy consumption (27.7%) and throughput (49%) while the end-to-end delay is comparable (4.87%) for both approaches.

Table 7.1 Comparison Table of Proposed SWSNM Approach with and without Malicious Nodes.

	<b>Location/ Time</b>	<b>SWSNM</b>	<b>SWSNM w/ 10 MN</b>	<b>%Diff.</b>
<b>Energy Consumption</b>	140 <sup>th</sup> Node	2.638 J	3.056 J	13.7%
<b>Throughput</b>	140 <sup>th</sup> Node	490.39	440.25	10.2%
<b>End-to-End Delay</b>	35 Minutes	0.04	0.053	24.5%

Table 7.2. Comparison Proposed SWSNM and Eagilla Approaches

	<b>Location/ Time</b>	<b>Eagilla [21]</b>	<b>SWSNM</b>	<b>% Diff.</b>
<b>Energy Consumption</b>	140th Node	3.6 J	2.6 J	27.7%
<b>Throughput</b>	140th Node	250	490.39	49.0%
<b>End-to-End Delay</b>	35 Minutes	0.042	0.039	4.87%



## CONCLUSIONS

Wireless sensor networks (WSNs) are an essential medium for the transmission of data for numerous applications. In order to address power consumption, communication, and security challenges, middleware bridges the gap between applications and WSNs. Most existing middleware does not completely address the issues that significantly impact WSNs' performance. Thus, our contribution proposes unsupervised learning for the development of middleware to provide end-to-end security for the system. The proposed algorithm consists of a generator and a discriminator network. The generator is capable of creating fake data to confuse the attacker and resolving imbalanced data by generating more data to balance the proportion between the normal and attack data classes. We render the discriminator to be a powerful network that can easily distinguish between two datasets, even if the fake data is very close to real samples. Extensive testing on the NSL-KDD dataset with different supervised learning techniques and comparisons shows that our generator model provides a better accuracy of 86.5% with a lower FPR. Additionally, we employed the t-SNE algorithm and normal distribution to compare the output results of our generator to the original dataset. The results show that the proposed generator performs very well with data visualization and normal distribution while the original, conventional dataset NSL-KDD performed worse with both algorithms.

The proposed GANs algorithm eliminates the need for fake sensor nodes, which consume more power and reduce both throughput and the lifetime of the network. In our experiment, we evaluated the efficiency of the proposed SWSNM and compared the security of the generated data with real data by using the D network. The results show that

even if the G generates real data, it can be easily detected by D network. In this case, the D network is capable of detecting attack data. The simulation results demonstrate that the proposed approach provides stronger security mechanism by detecting and replacing malicious nodes which leads to lower energy consumption, higher throughput, and an increased probability of successful data delivery to and from the base station.

In the future, real-time implementation of the SWSNM approach in more complex and layered networks could be implemented. The performance of the SWSNM approach, when scaled to larger number of nodes, would reveal key differences in security mechanisms in comparison to similar (more conventional) techniques. The SWSNM has the potential to significantly improve the overall performance of the wireless sensor networks.

## REFERENCES

- [1] A. Shchzad, N. Hung Quoc, S. Y. Lee, and L. Young-Koo, "A comprehensive middleware architecture for context-aware ubiquitous computing systems," in *Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05)*, Jeju Island, South Korea, pp. 251-256, 2005.
- [2] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 2, pp. 2-23, Second Quarter 2006.
- [3] Y. W. Law, J. Doumen, and P. Hartel, "Benchmarking block ciphers for wireless sensor networks," in *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, Fort Lauderdale, FL, USA, pp. 447-456, 2004.
- [4] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," presented at the Proceedings of the 3rd international symposium on Information processing in sensor networks, Berkeley, California, USA, pp.259-268, 2004.
- [5] A. A. Pirzada and C. McDonald, "Secure routing with the AODV protocol," in *Communications, 2005 Asia-Pacific Conference on*, Perth, WA, Australia, pp. 57-61, 2005.
- [6] S. Bhargava and D. P. Agrawal, "Security enhancements in AODV protocol for wireless ad hoc networks," *IEEE 54th Vehicular Technology Conference. VTC Fall 2001*. Atlantic City, NJ, USA, pp. 2143-2147, 2001.

- [7] S. Capkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Miami, FL, USA, pp. 1917-1928, 2005.
- [8] P. Aggarwal and S. K. Sharma, "Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection," *Procedia Computer Science*, vol. 57, pp. 842-851, 2015.
- [9] J. A. Jeyanna, E. Indumathi, and D. S. Punithavathani, "A Network Intrusion Detection System Using Clustering and Outlier Detection," *International Journal of Innovative Research in Computer and Communication Engineering(IJIRCCE)*, vol. 3, no. 2, pp. 975-982, 2015.
- [10] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38-43, 2004.
- [11] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, pp. 52-73, Second Quarter 2009.
- [12] S. Roy, N. Maitra, J. Nath, S. Agarwal, and A. Nath, "Ultra Encryption standard modified (UES) Version-I: Symmetric Key Cryptosystem using generalized modified Vernam Cipher method, Permutation method and Columnar Transposition method," in *Proceedings of IEEE sponsored National Conference on Recent Advances in Communication, Control and Computing Technology (RACCCT)*, pp. 29-30, 2012.
- [13] A. Swaminathan, B. S. Krishnan, and M. Ramaswamy, "A Novel Security Enhancement Strategy for Improving the Concert of CDMA Based Mobile Ad-Hoc

- Network," *International Journal of Modern Electronics and Communication Engineering (IJMECE)* vol. 5, pp. 1-8, 2017.
- [14] A. Kaur and S. S. Kang, "Attacks in Wireless Sensor Network-A Review," *International Journal of Computer Sciences and Engineering*, vol. 4, no. 5, pp. 157–162, 2016.
- [15] R. D. Shinganjude and D. P. Theng, "Inspecting the Ways of Source Anonymity in Wireless Sensor Network," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*, pp. 705-707, 2014.
- [16] G. Xu, W. Shen, and X. Wang, "Applications of Wireless Sensor Networks in Marine Environment Monitoring: A Survey," *Sensors*, vol. 14, no. 9, pp. 16932-16954, 2014.
- [17] S. Hadim and N. Mohamed, "Middleware for Wireless Sensor Networks: A Survey," in *1st International Conference on Communication Systems Software & Middleware*, New Delhi, India, pp. 1-7, 2006.
- [18] R. Alshinina and K. Elleithy, "Performance and Challenges of Service-Oriented Architecture for Wireless Sensor Networks," *Sensors*, vol. 17, no. 3, p. 536, 2017.
- [19] J. Al-Jaroodi and A. Al-Dhaheri, "Security issues of service-oriented middleware," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 11, no. 1, pp.153-160, 2011.
- [20] B. Bhuyan, H. K. D. Sarma, and N. Sarma, "A survey on middleware for wireless sensor networks," *Journal of Wireless Networking and Communications (JWNC)*, vol. 4, pp.7-17, 2014.

- [21] S. Hadim and N. Mohamed, "Middleware: middleware challenges and approaches for wireless sensor networks," *IEEE Distributed Systems Online*, vol. 7, no. 3, pp. 1-1, 2006.
- [22] M. Lopez-Ramos, J. Leguay, and V. Conan, "Designing a Novel SOA Architecture for Security and Surveillance WSNs with COTS," in *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Pisa, pp. 1-6, 2007.
- [23] M. Azarmi *et al.*, "An End-to-End Security Auditing Approach for Service Oriented Architectures," in *IEEE 31st Symposium on Reliable Distributed Systems (SRDS)*, Irvine, CA, pp. 279-284, 2012.
- [24] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, and G. M. L. Sarné, "A trust-aware, self-organizing system for large-scale federations of utility computing infrastructures," *Future Generation Computer Systems*, vol. 56, pp. 77-94, 2016.
- [25] A. Comi, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarné, "A partnership-based approach to improve QoS on federated computing infrastructures," *Information Sciences*, vol. 367–368, pp. 246-258, 2016.
- [26] M. Hammoudeh, S. Mount, O. Aldabbas, and M. Stanton, "Clinic: A Service Oriented Approach for Fault Tolerance in Wireless Sensor Networks," in *Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM)*, Venice, pp. 625-631, 2010.
- [27] M. Hammoudeh, A. Kurz, and E. Gaura, "MuMHR: Multi-path, Multi-hop Hierarchical Routing," in *International Conference on Sensor Technologies and Applications (SENSORCOMM 2007)*, Valencia, pp. 140-145, 2007.

- [28] B. Bhuyan, H. K. D. Sarma, and N. Sarma, "A survey on middleware for wireless sensor networks," *Journal of Wireless Networking and Communications*, vol. 4, no. 1, pp. 7-17, 2014.
- [29] Y. Sahni, J. Cao, and X. Liu, "MidSHM: A Flexible Middleware for SHM Application Based on Service-Oriented Architecture," in *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, Oxford, pp. 126-135, 2016.
- [30] S. Chien-Chung, C. Srisathapornphat, and C. Jaikaeo, "Sensor information networking architecture and applications," *IEEE Personal Communications*, vol. 8, no. 4, pp. 52-59, 2001.
- [31] W. B. Heinzelman, A. L. Murphy, H. S. Carvalho, and M. A. Perillo, "Middleware to support sensor network applications," *IEEE Network*, vol. 18, no. 1, pp. 6-14, 2004.
- [32] X. Koutsoukos, M. Kushwaha, I. Amundson, S. Neema, and J. Sztipanovits, "OASiS: A Service-Oriented Architecture for Ambient-Aware Sensor Networks," in *Composition of Embedded Systems. Scientific and Industrial Issues: 13th Monterey Workshop 2006 Paris, France, 2006 Revised Selected Papers*, F. Kordon and O. Sokolsky, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 125-149, 2008.
- [33] I. Amundson, M. Kushwaha, X. Koutsoukos, S. Neema, and J. Sztipanovits, "OASiS: a service-oriented middleware for pervasive ambient-aware sensor networks," *Pervasive and mobile computing journal on middleware for pervasive computing*, Institute for Software Integrated Systems, Vanderbilt University, Tech. Rep. ISIS-06-706, 2006.
- [34] J. e. e. Leguay, M. LopezRamos, Kathlyn JeanMarie, and V. Conan, "An efficient service oriented architecture for heterogeneous and dynamic wireless sensor networks,"

- in *33rd IEEE Conference on Local Computer Networks (LCN)*, Montreal, Que, pp. 740-747, 2008.
- [35] B. L. Corre, J. Leguay, M. Lopez-Ramos, V. Gay, and V. Conan, "Service Oriented Tasking System for WSN," in *Developments in E-systems Engineering*, London, pp. 64-69, 2010.
- [36] A. Coronato, "Uranus: A Middleware Architecture for Dependable AAL and Vital Signs Monitoring Applications," *Sensors*, vol. 12, no. 3, pp. 3145, 2012.
- [37] Y. Bai, H. Ji, Q. Han, J. Huang, and D. Qian, "MidCASE : A Service Oriented Middleware Enabling Context Awareness for Smart Environment," presented at the *2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, Seoul, pp. 946-951, 2007.
- [38] M. Ananthi and M. R. Sumalatha, "Integrating WSN with web services for patient's record management using RFID," *2013 3rd IEEE International Advance Computing Conference (IACC)*, Ghaziabad, pp.605-609, 2013.
- [39] K. K. Khedo and R. Subramanian, "A service-oriented component-based middleware architecture for wireless sensor networks," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 9, no. 3, pp. 174-182, 2009.
- [40] A. Malatras, A. Asgari, and T. BaugÉ, "Web Enabled Wireless Sensor Networks for Facilities Management," *IEEE Systems Journal*, vol. 2, no. 4, pp. 500-512, 2008.
- [41] A. Sleman and R. Moeller, "Micro SOA Model for Managing and Integrating Wireless Sensor Network into IP-Based Networks," in *2nd International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*, Liverpool, pp. 137-142, 2010.



- [42] F. Kerasiotis, C. Koulamas, and G. Papadopoulos, "Developing wireless sensor network applications based on a function block programming abstraction," in *IEEE International Conference on Industrial Technology (ICIT)*, Athens, pp. 372-377, 2012.
- [43] K. Lingaraj, R. V. Biradar, and V. C. Patil, "Eagilla: An Enhanced Mobile Agent Middleware for Wireless Sensor Networks," *Alexandria Engineering Journal*, 2017.
- [44] E. Cañete, J. Chen, M. Díaz, L. Llopis, and B. Rubio, "USEME: A Service-Oriented Framework for Wireless Sensor and Actor Networks," in *Eighth International Workshop on Applications and Services in Wireless Networks (aswn 2008)*, Kassel, pp. 47-53, 2008.
- [45] E. Cañete, J. Chen, M. Díaz, L. Llopis, and B. Rubio, "A Service-Oriented Middleware for Wireless Sensor and Actor Networks," in *Sixth International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, NV, pp. 575-580, 2009.
- [46] K. Pandey and S. V. Patel, "A Novel Design of Service Oriented and Message Driven Middleware for Ambient Aware Wireless Sensor Network," *International Journal of Recent Trends in Engineering (RTE)*, vol. 1, pp. 313-317, 2009.
- [47] F. Aijaz, S. M. Adeli, and B. Walke, "A service-oriented approach for in-network computations in wireless networks," in *International Conference on Wireless and Optical Communications Networks (WOCN)*, Cairo, pp. 1-6, 2009.
- [48] G. Anastasi, E. Bini, A. Romano, and G. Lipari, "A service-oriented architecture for QoS configuration and management of Wireless Sensor Networks," in *IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, Bilbao, pp. 1-8, 2010.

- [49] M. M. Faghieh and M. E. Moghaddam, "SOMM: A New Service Oriented Middleware for Generic Wireless Multimedia Sensor Networks Based on Code Mobility," *Sensors*, vol. 11, no. 11, p. 10343, 2011.
- [50] F. Sheikhha and M. E. Moghaddam, "Service-Oriented Wireless Multimedia Sensor Network Middleware Using Infra-Red Cameras," in *Third International Conference on Mobile, Ubiquitous, and Intelligent Computing (MUSIC)*, Vancouver, BC, pp. 230-235, 2012.
- [51] E. Avilés-López and J. A. García-Macías, "TinySOA: a service-oriented architecture for wireless sensor networks," *Service Oriented Computing and Applications*, vol. 3, no. 2, pp. 99-108, 2009.
- [52] V.Vanitha, V.Palanisamy, N.Johnson, and G.Aravindhbabu, "LiteOS based extended service oriented architecture for wireless sensor networks," *International Journal of Computer and Electrical Engineering (IJCEE)*, vol. 2, pp. 432-436, 2010.
- [53] J. Ibbotson *et al.*, "Sensors as a Service Oriented Architecture: Middleware for Sensor Networks," in *Sixth International Conference on Intelligent Environments (IE)*, Kuala Lumpur, pp. 209-214, 2010.
- [54] H. Abangar, P. Barnaghi, K. Moessner, A. Nnaemego, K. Balaskandan, and R. Tafazolli, "A service oriented middleware architecture for wireless sensor networks," *Centre for Communication Systems Research, University of Surrey, Guildford, UK, Future Network & MobileSummit 2010 Conference Proceedings, IIMC International Information Management Corporation*, 2010.

- [55] E. Aguilar, A. J. Torralba, L. Collar, and D. Villalba, "A Service Oriented Wireless Platform for Acquisition and Control (SOWPAC)," in *39th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Vienna, pp. 5444-5449, 2013.
- [56] M. S. Aslam, S. Rea, and D. Pesch, "A vision for Wireless Sensor Networks: Hybrid architecture, model framework and service based systems," presented at the Fifth International Conference on Digital Information Management (ICDIM), Thunder Bay, ON, pp.353-358, 2010.
- [57] B. Alkazemi, E. Felemban, A. Abid, and F. Al-Zahrani, "Middleware model for Wireless Sensor Networks," in *International Conference on Multimedia Computing and Systems (ICMCS)*, Tangier, pp. 67-71, 2012.
- [58] L. Q. Zhuang, J. B. Zhang, Y. Z. Zhao, M. Luo, D. H. Zhang, and Z. H. Yang, "Power-aware service-oriented architecture for wireless sensor networks," in *31st Annual Conference of IEEE Industrial Electronics Society (IECON)*, Raleigh, NC, USA, pp. 1-6, 2005.
- [59] N. Hua, N. Yu, and Y. Guo, "Research on Service Oriented and Middleware Based Active QoS Infrastructure of Wireless Sensor Networks," in *10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN)*, Kaohsiung, pp. 208-213, 2009.
- [60] R. Eltarras, M. Eltoweissy, and M. Youssef, "Towards evolving Sensor Actor NETWORKs," in *Proceedings of the IEEE INFOCOM 2008 Conference on Computer Communications Workshops*, Phoenix, AZ, pp. 1-6, 2008.
- [61] F. Pramudianto, J. Simon, M. Eisenhauer, H. Khaleel, C. Pastrone, and M. Spirito, "Prototyping the Internet of Things for the future factory using a SOA-based

- middleware and reliable WSNs," in *IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, Cagliari, pp. 1-4, 2013.
- [62] P. Gorski, F. Golasowski, R. Behnke, C. Fabian, K. Thurow, and D. Timmermann, "Wireless Sensor Networks in Life Science applications," in *3rd International Conference on Human System Interaction*, Rzeszow, pp. 594-598, 2010.
- [63] D. I. Tapia, J. A. Fraile, S. Rodríguez, J. F. de Paz, and J. Bajo, "Wireless Sensor Networks in Home Care," in *Bio-Inspired Systems: Computational and Ambient Intelligence: 10th International Work-Conference on Artificial Neural Networks, IWANN 2009, Salamanca, Spain*, pp. 1106-1112, 2009.
- [64] K. Ganapathy, B. Priya, B. Priya, Dhivya, V. Prashanth, and V. Vaidehi, "SOA Framework for Geriatric Remote Health Care Using Wireless Sensor Network," *Procedia Computer Science*, vol. 19, pp. 1012-1019, 2013.
- [65] F. Pu, T. Gao, J. Pan, J. Li, and C. Li, "SunShine: A Service-Oriented Architecture Based on Wireless Sensor Network for Health Monitoring and Tracking," in *International Conference on Biomedical Engineering and Computer Science (ICBEC)*, Wuhan, pp. 1-4, 2010.
- [66] K. Ganapathy and V. Vaidehi, "Medical intelligence for quality improvement in Service Oriented Architecture," in *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*, Chennai, Tamil Nadu, pp. 161-166, 2011.
- [67] R. Kyusakov, J. Eliasson, J. Delsing, J. v. Deventer, and J. Gustafsson, "Integration of Wireless Sensor and Actuator Nodes With IT Infrastructure Using Service-Oriented Architecture," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 43-51, 2013.

- [68] Q. Duan, "Applying the Service-Oriented Architecture for Network Discovery and Selection in the Next Generation Wireless Mobile Networks," in *International Conference on Network-Based Information Systems*, Indianapolis, IN, pp. 380-385, 2009.
- [69] J. C. Liu and K. Y. Chuang, "WASP: An innovative sensor observation service with web-/GIS-based architecture," in *17th International Conference on Geoinformatics*, Fairfax, VA, pp. 1-6, 2009.
- [70] J.-C. Liu, K.-Y. Chuang, and C.-F. Ye, "A Highly Flexible System for Smart Home Sensor Networks," in *Fourth International Conference on Genetic and Evolutionary Computing (ICGEC)*, pp. 775-778, 2010.
- [71] A. Ghobakhlou, A. Kmoch, and P. Sallis, "Integration of Wireless Sensor Network and Web Services," in *Proceedings of the 20th International Congress on Modelling and Simulation*, Adelaide, Australia, pp.838-844, 2013.
- [72] A. Ghobakhlou, P. Sallis, and X. Wang, "A service oriented wireless sensor node management system," in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, Montevideo, pp. 1475-1479, 2014.
- [73] S. Sawant, J. Adinarayana, S. Durbha, A. Tripathy, and D. Sudharsan, "Service oriented architecture for wireless sensor networks in agriculture," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Proceedings of the ISPRS Congress, Melbourne, Australia*, pp. 467-472, 2012.
- [74] M. S. Aslam, S. Rea, and D. Pesch, "Service Provisioning for the WSN Cloud," in *IEEE Fifth International Conference on Cloud Computing*, Honolulu, HI, pp. 962-969, 2012.

- [75] S. V. Patel and P. Kamlendu, "Design of SOA Based Framework for Collaborative Cloud Computing in Wireless Sensor Networks," *International Journal of Grid and High Performance Computing (IJGHPC)*, vol. 2, no. 3, pp. 60-73, 2010.
- [76] K. K. Pandey and S. V. Patel, "Design of SOA based service stack for collaborative wireless sensor network," in *2009 Fifth International Conference on Wireless Communication and Sensor Networks (WCSN)*, Allahabad, pp. 1-5, 2009.
- [77] A. Pratap Singh, O. P. Vyas, and S. Varma, "Flexible Service Oriented Network Architecture for Wireless Sensor Networks," *International Journal of Computers Communications & Control*, Wireless Sensor Networks (WSN), Flexible Service Oriented Network Architecture (FSONA). middleware architecture, interoperability, localization vol. 9, no. 5, p. 13, 2014.
- [78] S. Li, S. Zhao, X. Wang, K. Zhang, and L. Li, "Adaptive and Secure Load-Balancing Routing Protocol for Service-Oriented Wireless Sensor Networks," *IEEE Systems Journal*, vol. 8, no. 3, pp. 858-867, 2014.
- [79] N. Komoda, "Service Oriented Architecture (SOA) in Industrial Systems," in *4th IEEE International Conference on Industrial Informatics*, Singapore, pp. 1-5, 2006.
- [80] M. Thoma, K. Sperner, and T. Braun, "Service descriptions and linked data for integrating WSNs into enterprise IT," *2012 Third International Workshop on Software Engineering for Sensor Network Applications (SESENA)*, Zurich, pp. 43-48, 2012.
- [81] F. C. Delicato, P. F. Pires, L. Pirmez, and T. Batista, "Wireless Sensor Networks as a Service," *17th IEEE International Conference and Workshops on Engineering of Computer Based Systems*, Oxford, pp. 410-417, 2010.

- [82] T. Yu, Q. Chen, Q. Li, R. Liu, W. Wang, and W. Liu, "A System for Web-Based Interactive Real-Time Data Visualization and Analysis," *IEEE Conference on Commerce and Enterprise Computing*, Vienna, pp. 453-459, 2009.
- [83] M. Parhi, B. M. Acharya, and B. Puthal, "An effective mechanism to discover sensor web registry services for wireless sensor network under x-SOA approach," *Trendz in Information Sciences & Computing(TISC2010)*, Chennai, pp. 197-201, 2010.
- [84] M. Parhi, B. M. Acharya, and B. Puthal, "Discovery of sensor web registry services for WSN with Multi-layered SOA framework," *2nd International Conference on Computer and Communication Technology (ICCT-2011)*, Allahabad, pp. 524-530, 2011.
- [85] A. Amokrane, Y. Challal, and A. Balla, "A Secure Web Service-Based Platform for Wireless Sensor Network Management and Interrogation," *2011 Conference on Network and Information Systems Security*, La Rochelle, pp. 1-8, 2011.
- [86] S. R. a. T.Senthil, "Efficient SOA-based Network Management Architecture in Wireless Sensor Networks," *International Conference on Web Services Computing (ICWSC)*, pp. 50-54, 2011.
- [87] H. Hellbrück, T. Teubler, and S. Fischer, "Name-Centric Service Architecture for Cyber-Physical Systems (Short Paper)," *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*, Koloa, HI, pp. 77-82, 2013.
- [88] A. Kovacevic, J. Ansari, and P. Mahonen, "NanoSD: A Flexible Service Discovery Protocol for Dynamic and Heterogeneous Wireless Sensor Networks," *2010 Sixth International Conference on Mobile Ad-hoc and Sensor Networks*, Hangzhou, pp. 14-19, 2010.

- [89] L. F. Herrera-Quintero, F. Maciá-Pérez, D. Marcos-Jorquera, and V. Gilart-Iglesias, "Wireless Sensor Networks and Service-Oriented Architecture, as suitable approaches to be applied into ITS," *2012 6th Euro American Conference on Telematics and Information Systems (EATIS)*, Valencia, pp. 1-8, 2012.
- [90] J. Miranda *et al.*, "A Wireless Sensor Network for collision detection on guardrails," *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, Istanbul, pp. 1430-1435, 2014.
- [91] X. Wang, J. Wang, Z. Zheng, Y. Xu, and M. Yang, "Service Composition in Service-Oriented Wireless Sensor Networks with Persistent Queries," *2009 6th IEEE Consumer Communications and Networking Conference*, Las Vegas, NV, pp. 1-5, 2009.
- [92] F. C. Delicato, P. F. Pires, L. Rust, L. Pirmez, and J. F. d. Rezende, "Reflective middleware for wireless sensor networks," *20th ACM symposium on Applied computing (ACM SAC)*, Santa Fe, New Mexico, pp. 1155-1159, 2005.
- [93] H. Zhou, Z. Huang, and G. Zhao, "A service-centric solution for wireless sensor networks," *2010 5th International Conference on Communications and Networking (ICST)*, China, Beijing, pp. 1-5, 2010.
- [94] S. Y. Shah, B. Szymanski, P. Zerfos, C. Bisdikian, C. Gibson, and D. Harries, "Autonomous configuration of spatially aware sensor services in service oriented WSNs," *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, San Diego, CA, pp. 312-314, 2013.



- [95] G. Moritz, C. Cornelius, F. Golatowski, D. Timmermann, and R. Stoll, "Differences and Commonalities of Service-Oriented Device Architectures, Wireless Sensor Networks and Networks-on-Chip," *2009 International Conference on Advanced Information Networking and Applications Workshops*, Bradford, pp. 482-487, 2009.
- [96] B. Upadhyaya, Y. Zou, H. Xiao, J. Ng, and A. Lau, "Migration of SOAP-based services to RESTful services," *2011 13th IEEE International Symposium on Web Systems Evolution (WSE)*, Williamsburg, VI, pp. 105-114, 2011.
- [97] H. Cao and J. Chen, "Service-Oriented Transparent Interconnection between Data-Centric WSN and IP networks," *2010 International Conference on Electrical and Control Engineering*, Wuhan, pp. 1884-1887, 2010.
- [98] Mohammad Ali Shamalizadeh, S. Shamshirband, M. Amiri, and S. Kalantari, "Security in Wireless Sensor Networks Based On Service-Oriented Architecture," *Australian Journal of Basic and Applied Sciences*, vol. 5, pp. 694-701, 2011.
- [99] M. E. F. Maia, L. S. Rocha, and R. M. C. Andrade, "Requirements and challenges for building service-oriented pervasive middleware," *Proceedings of the 2009 international conference on Pervasive services*, London, United Kingdom, pp. 93-102, 2009.
- [100] L. Kwong Yuen, P. Thi Khoi Anh, and Z. Tari, "Efficient SOAP binding for mobile Web services," *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)*, Sydney, NSW, pp. 218-225, 2005.
- [101] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996-2018, 2014.

- [102] N. Ahad, J. Qadir, and N. Ahsan, "Neural networks in wireless networks: Techniques, applications and guidelines," *Journal of Network and Computer Applications*, vol. 68, pp. 1-27, 2016.
- [103] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier Detection Techniques for Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 159-170, 2010.
- [104] D. Janakiram, V. A. Reddy, and A. V. U. P. Kumar, "Outlier Detection in Wireless Sensor Networks using Bayesian Belief Networks," *2006 1st International Conference on Communication Systems Software & Middleware*, New Delhi, India, pp. 1-6, 2006.
- [105] J. Branch, B. Szymanski, C. Giannella, W. Ran, and H. Kargupta, "In-Network Outlier Detection in Wireless Sensor Networks," *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, Lisboa, Portugal, Portugal, p. 51, 2006.
- [106] S. Kaplantzis, A. Shilton, N. Mani, and Y. A. Sekercioglu, "Detecting Selective Forwarding Attacks in Wireless Sensor Networks using Support Vector Machines," *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, Melbourne, Qld., Australia, pp. 335-340, 2007.
- [107] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter Sphere Based Distributed Anomaly Detection in Wireless Sensor Networks," *2007 IEEE International Conference on Communications*, Glasgow, UK, pp. 3864-3869, 2007.
- [108] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is "Nearest Neighbor" Meaningful?," in *Processing of the 7th International Conference Database*

- Theory (ICDT), Jerusalem, Israel*, C. Beeri and P. Buneman, Eds. Berlin, Germany: Springer, pp. 217-235, 1999.
- [109] I. Goodfellow *et al.*, "Generative adversarial nets," in *Processing Advances in neural information processing systems*, pp. 2672-2680, 2014.
- [110] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Processing Advances in Neural Information Processing Systems*, pp. 2234-2242, 2016.
- [111] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," [Online]. Available: *arXiv preprint arXiv:1511.06390*, 2015.
- [112] University of New Brunswick. *NSL-KDD dataset*. [Online]. Available: <http://nsl.cs.unb.ca/nsl-kdd>.
- [113] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, pp. 1-6, 2009.
- [114] University of California, Irvine. (1999). *KDD Cup 1999*. [Online]. Available: <http://Kdd.Ics.Uci.Edu/Databases/Kddcup99.html>, 1999.
- [115] S. Hettich and S. Bay, "The UCI KDD Archive " *Department of Information and Computer Science, University of California, Irvine, CA, USA, Tech.Rep.,1999*, vol. 152. [Online]. Available: <http://kdd.ics.uci.edu>,1999.
- [116] L. Ray, "Determining the Number of Hidden Neurons in a Multi Layer Feed Forward Neural Network," *Journal of Information Security Research*, vol. 4, no. 2, pp. 63-70, 2013.

- [117] F. Chollet. (2015). Keras. [Online]. Available:<http://github.com/fchollet/keras>
- [118] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, pp. 448-456, 2015.
- [119] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249-256, 2010.
- [120] K. He and J. Sun, "Convolutional neural networks at constrained time cost," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, pp. 5353-5360, 2015.
- [121] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in neural information processing systems*, pp. 2377-2385, 2015.
- [122] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, 2016.
- [123] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," *2015 International Conference on Signal Processing and Communication Engineering Systems*, Guntur, India, pp. 92-96, 2015.
- [124] M. Panda, A. Abraham, and M. R. Patra, "Discriminative multinomial Naïve Bayes for network intrusion detection," *2010 Sixth International Conference on Information Assurance and Security*, Atlanta, GA, USA, pp. 5-10, 2010.
- [125] L. M. Ibrahim, D. T. Basheer, and M. S. Mahmood, "A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self organization map (SOM) artificial

- neural network," *Journal of Engineering Science and Technology*, vol. 8, no. 1, pp. 107-119, 2013.
- [126] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579-2605, 2008.
- [127] S. S. Singh and Y. B. Jinila, "Sensor node failure detection using check point recovery algorithm," *2016 International Conference on Recent Trends in Information Technology (ICRTIT)*, Chennai, India, pp. 1-4, 2016.
- [128] G. Sumalatha, N. Zareena, and C. Raju, "A review on failure node recovery algorithms in wireless sensor actor networks," *arXiv preprint arXiv:1407.0009*, 2014.
- [129] P. Jiang, "A New Method for Node Fault Detection in Wireless Sensor Networks," *Sensors*, vol. 9, no. 2, p. 1282, 2009.
- [130] Y. LeCun.(1999). "The MNIST database of handwritten digits," [Online] Available: <http://yann.lecun.com/exdb/mnist/>,1999
- [131] R. Alshinina and K. Elleithy, "A highly accurate machine learning approach for developing wireless sensor network middleware," *2018 Wireless Telecommunications Symposium (WTS)*, Phoenix, AZ, pp. 1-7, 2018.
- [132] R. Alshinina and K. Elleithy, "Efficient Unsupervised Learning to Secure Communication for Wireless Sensor Network Middleware," *2018 Wireless Telecommunications Symposium (WTS)*, Phoenix, AZ, pp. 1-5, 2018.
- [133] R. A. Alshinina and K. M. Elleithy, "A Highly Accurate Deep Learning Based Approach for Developing Wireless Sensor Network Middleware," *IEEE Access*, vol. 6, pp. 29885-29898, 2018.