The University Of Sheffield.

An Energy Efficient Data Architecture for Wireless Sensor Networks

Fesal Baxhaku

Supervisors: Prof. George Eleftherakis,
Prof. Eleni Vasilaki

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

The University of Sheffield
Faculty of Engineering
Department of Computer Science

January, 2023

Declaration

I, the author, confirm that the Thesis is my own work. I am aware of the University's Guidance on the Use of Unfair Means (www.sheffield.ac.uk/ssid/unfair-means). This work has not been previously been presented for an award at this, or any other, university.

Acknowledgments

I want to express my genuine thankfulness to my supervisors, Prof. George Eleftherakis and Prof. Eleni Vasilaki, for their continuous and invaluable support, patience, and encouragement throughout my Ph.D. journey. It's hard to say how thankful I'm for all these years you have guided and advised me and for always being available with your kind and generous words. Thank you!

Thank you to my mother for always caring and guiding me, and my father who would have been proud today. Special thanks to my wife Almira for always supporting and encouraging me throughout these years. Every single day you make my day easier. I cannot forget to thank my two Brothers for always encouraging me and helping in this journey. You were my greatest supporters on all these years. I'm very grateful to all my family members for kindness, love and support you gave me.

Abstract

We live in a technological generation surrounded by interconnected sensors that can collect and distribute immense amounts of data on a daily basis. These data would have a better connotation and would have been more practical if sensor-based networks allowed us to capture and monitor the characteristics of physical objects from a highly dynamic environment. At this point, sensor-based networks could substantially enhance their applicability if machines process and interpret vast amounts of data correctly, an essential characteristic of scalable and interoperable wireless sensor network architectures. Through this research project, a) We will identify and evaluate wireless sensor network architectures enabling applications from a highly dynamic environment. Then a data architecture will be proposed to enhance machine-to-machine (M2M) communication and human understanding, considering the issues and challenges of sensor networks. The future proposed data architecture will overcome the existing data frameworks' limitations identified in the literature review. b) The significant contribution of the research is to propose energy-efficient data collection models (the first layer of data architecture) that will reduce data transmissions using prediction models between nodes in sensor networks. The proposed models intend to predict values at the sink node using coefficients built and transmitted by sensor platforms. The goal is to build models that improve the energy of battery-powered sensory devices by reducing data transmissions and recovering values at sink nodes using the same coefficients of models while ensuring data integrity. Furthermore, the models are evaluated using real data sets from real sensor networks with the following metrics; RMSE, MAE, MSE, data reduction percentage, and energy savings.

Keywords: *Data Architecture, Data Annotation, Wireless Sensor Networks, Ontologies, Data Prediction, Data Reduction, Energy Efficiency in WSN.*

Contents

List of Figures

Chapter 1

Introduction

We live in the new era of computing, where billions of nodes connect unpredictably, create their mutual relations, and share information with other nodes within the network. The relations vary over time and make what is called a *dynamic network*. From the real-world perspective, plenty of network examples can be used as a model to mimic dynamic networks and reproduce their structural properties [1]. Protein-protein interaction networks, human-human interactions, transportation networks, neural networks in biology, and animal populations are just a few examples that can be analyzed and applied to dynamic networks. These systems, even though they have always existed, they came to attention when *time* became an integral part of the network for a formal treatment of systems [2]. The most significant transformation is happening in computer science with wireless protocols, low-cost wireless sensory devices, smartphones, and other intelligent devices.

Compared to the early years of computing, where most of the systems can be considered static, and relatively predictable [2], with the new technological advancements (tiny sensory devices, wireless protocols) and mobile users constitute an era of dynamicity and unpredictability. According to Cisco, [3], the number of machines/devices connected to the Internet is expected to be around thirty billion by 2030. All these devices will have sensing capabilities to gather and share their data with the peers in the network. Internet of Things envisions huge managed devices with sensors that connect to the Internet. Case studies of smart cities, smart farming, and intelligent hospitals are already present. On the other hand, the Internet of Things expects further connections; people, processes, things, and information which altogether create the Internet of Everything (IoE) [4]. In this paradigm, people might be nodes on the Internet, data will provide insightful information when linked and combined, and "things" expect a connection with one another and the Internet. Finally, the correct process is necessary to deliver accurate service to the right user efficiently.

At this point, systems cannot perform their tasks flawlessly and predict everything. Rather, many possible interactions among "things" spontaneously move in the environment. Computer scientists and engineers are putting much effort into developing algorithms that self-assemble into desired structures, such as the case of programmed DNA molecules [5] or spontaneous local activities in cortical development that reveal large-scale network structure [6]. All these challenging and ambitious applications are dynamic.

1.1 A variety of context

A network is a group of interconnected entities with established relations that can exchange data. It is considered dynamic if the network, specifically the relations, changes over time.

Most of the time, the term dynamic network is used as a synonym for complex systems. Either it is used as a means to describe huge distributed devices operating in a wireless network that deals with dynamic environments or a network in a broader sense to describe network systems from real life. Kuhn et al. [7] describes dynamic networks as an enormous number of nodes, totally decentralized, where each node knows only its local peers. Casteigts [8] distinguishes dynamic networks that consist of wireless entities; smartphones, laptops, sensors, robots, and complex networks, which include a broad range of networks such as those from social sciences, biology, transportation, etc. In the first category fall, networks that allow occasional connections up to unrestricted. Complex networks provide a means to describe a wide range of domains where connections among "things" are neither purely random nor purely regular; they produce a massive amount of data available for analysis in identifying different phenomena.

It is the motivation that distinguishes dynamic versus complex networks. The network complexity is described via mathematical models, while the dynamic network aims to solve interaction issues among entities within the network. In general, the term dynamic network is used by numerous disciplines in an approach to describe complex systems [9]. Following the above, a complex system describes interactions between system components, individuals, or players [10]. When two or more components work together, and the input they receive produces results that cannot be achieved from the components individually, it makes a system complex [11]. Specifically, how simple entities with no initial structure organize themselves to create patterns collectively without any central controller and use the information to learn and evolve [12]. Sagut and McGrath [13] distinguish a) complicated systems composed of individual parts that somehow follow a pattern and b) complex systems, which despite the options to follow some patterns, the interactions between parts are constantly changing.

Hereafter, for the purpose of this study the term complex system is going to be defined as *"A vast number of nodes/components with no well-defined infrastructure and completely decentralized able to interact with local peers spontaneously, and have the ability of the system to load balance when a massive number of users access the applications/network"*

In this matter, the working environment is highly dynamic, meaning that nodes may enter or leave the network unpredictably over time. The interactions among components happen instantly without any notice. Further, the number of users accessing the network(application) must be handled, and the system must process the required service by such users, even during congestion time. Additionally, the tasks are executed in a distributed manner, and the system evolves and adapts to all the changes in the environment. Such a system is of enormous scale, high heterogeneity, and interdependence that characterizes complex systems [13, 14]. One example of such a complex system is the case of the Internet of Things, specifically Wireless Sensor Networks.

1.2 Motivation

1.2.1 Wireless Sensor Networks

Wireless Sensor Networks consist of a considerable number of battery-powered sensor nodes connected to one another and the internet that can collect data and exchange data with the peers and the world. The volume of generated data by these embedded devices with sensing capabilities is extremely high. They have a high potential to technologically advance, automate and create many business possibilities [15, 16]. They could substantially enhance the decision-making by humans and machines themselves. It can enhance machine-to-machine (M2M) communication, striving to process all those data to create even more data, but at a sophisticated level, thus producing more practical knowledge. This mode will create a continuous cycle where information is continually processed and be available in the suitable form for the right consumer (machine or human) in the near future. In order to happen all this, it is required by the modern sensor networks to provide flexibility to operate in a decentralized mode and without a well-defined infrastructure.

There is an expanded demand for WSN solutions that operate in highly dynamic surroundings and enable the design of an infrastructure of a WSN dynamically and autonomously, specifically in conditions where well-defined and well-designed infrastructures do not exist or are not preferable. Those cases will most likely become the primary trend in the future because of the lack of well-defined situations and thus well-designed solutions which operate in the best feasible mode for those problems.

The data creates an even greater demand for interoperability than ever before to help machines to process all those data created from highly dynamic circumstances, as described above. Although, there is a clear need to consider all issues and challenges imposed because of the nature of WSNs.

This emerging need demands a distinct level of abstraction to bridge the gap between sensory devices and applications and improve the M2M communication to create more sophisticated applications. First, the ample amount of data observed and collected from sensor networks creates the need for a well-defined syntax and encodings (syntactic interoperability) to enable the communication and exchange of such observations in the most efficient way between the interconnected things. Secondly, the raw data cannot be automatically identified or appropriately processed by other devices if there are no precise, well-defined semantics. The raw data must be altered into information to boost the extraction of high-level wisdom and enhance the decision-making process either by machines or humans.

1.2.1.1 Energy Efficiency in WSNs

Energy efficiency is a critical concern among many researchers in the field of Wireless Sensor Networks. The nodes in Wireless Sensor Network have limitations in energy as well as processing. In order to operate in the open environment that is often critical (emergency monitoring, fire explosions, crop field monitoring etc.), or open environments like monitoring sea/ocean water quality, it is necessary to increase the network lifetime. Specifically, it is necessary to improve energy of sensor nodes while gathering sensed information and trans-

mitting them to the right node and then to the right application. In many cases it becomes infeasible to replace batteries in sensor nodes and thus not being able to monitor and sense the objects and the surroundings.

Current energy and processing constraints in sensor nodes creates difficulties for the machine to machine communications and advancing further existing applications [17, 18]. In order to automate and create more autonomy in existing systems and applications, it is a demand to reduce power consumption and prolong network lifetime. There are already great studies with regard to energy efficiency mainly focused on network typologies and few on data collection methods.

1.3 Aim and Objectives

This work aims to identify and adopt the right network architecture that will enable well-designed solutions in extremely dynamic environments, especially in situations where well-defined and well-designed infrastructures do not exist or are not preferable. Then it will propose a data architecture that will enable the efficient communication of information in this network considering all the challenges imposed by sensor networks stated above. A significant contribution lies in improving energy efficiency in the data collection phase while reducing transmission. The applicability of the proposed models will be validated with a real-world sensor data sets. The objectives are as follows in priority order:

- adopts a robust network architecture that enables mobile ad hoc network solutions and affords an extremely high number of sensor nodes and concurrent users in real-time to form a stable architecture for further enhancements.
- proposes a data architecture for the adopted network architecture to enable syntactic and semantic data interoperability considering the limitations of Wireless Sensor Nodes (bandwidth, memory, CPU, and energy).
- using the selected architecture after the thorough investigation, propose a new data framework mentioned, that enhances machine-to-machine communication and understanding in the adopted network architecture.
- improve energy efficiency in the network architecture by reducing data transmission via dual prediction models
- evaluate the algorithmic models using real use case data sets
- evaluate the efficiency of the proposed models; energy efficiency, data accuracy, data processing

1.4 Structure

The structure of the research work is as follows: Chapter II provides the literature background of the challenges and the current state of the art of sensor network architectures that

can work in a highly dynamic environment. Further, we present the existing standards, discuss the data semantics, and evaluate existing data architectures. Moreover, we describe the methodology of the research proposal. Chapter III evaluates and adopts the right network architecture from the challenges identified from the background literature review. Then, we propose a new extended model architecture for the adopted network architecture. After analysing, adopting and extending the network architecture, we introduce a new data architecture addressing some of the Sensor Networks' challenges identified in the literature review. The following three chapters propose new approaches for improving energy efficiency using prediction algorithms. Finally, future work concludes the research work of the current thesis.

Chapter 2

Literature Review

IoT vision presents exciting opportunities to almost every activity we perform on daily basis, while unique challenges need to be tackled to make it happen. Under a dynamic global network architecture every "thing" is connected, is accessible in real time, and is able to act intelligently upon the available information of the surrounding environment. The pervasive presence of enormous heterogeneous devices with sensing capability allows to uniquely identify "things", communicate data with other peers in the network which cooperatively are able to achieve a common goal. The most basic technology is RFID [19] whose functionality is to uniquely identify track and monitor "things" using tags. Another technology with high impact in IoT is Wireless Sensor Network (WSN) which uses sensors to sense physical properties of "things", monitor them and actuators can be used for interactions with objects under monitoring. Other than this, smart phones, tablets, cloud computing and a number of technologies make what so called IoT [20]. One particular technology with highest impact in IoT is Wireless Sensor Network (WSN)

2.1 Understanding Wireless Sensor Networks

Wireless sensor networks are an emerging solution for real-time wide-area monitoring in many fields of interest like agriculture, healthcare, military, and other interesting domains. The applicability of this technology to a range of domain problems comes as a result of satisfactory features and advantages in the market as compared to other monitoring technologies.

In this section, we introduce wireless sensor networks as the main contributors towards the Internet Of Things, problems that need to be tackled by the network architectures, the massive amount of data produced by the sensing devices, including the issues imposed by such massive data production. Further, we will discuss network architectures for WSN and already existing data architectures where the comparison study will be given.

2.1.1 Wireless Sensor Network

A large number of sensor nodes can be connected Wirelessly using one of the radio frequency technology Bluetooth, ZigBee, or other constraint communication protocols, and form what

is called Wireless Sensor Network. Sensors connect with sensor platforms, measure the important features of the objects being monitored and transmit measurements whenever changes are detected.

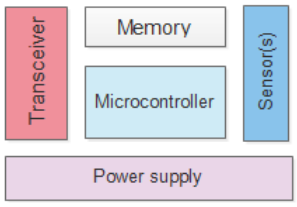


Figure 2.1: Sensor Nodes Components

There are two forms of sensor deployments when observing the phenomena [21]: a) distance monitoring and b) nearby field. The sensor platform accepts the sensed data as time series and can process some data (very limited). As depicted in Figure 2.1, each node can store some raw data in memory, transmit the data through a radio transceiver, and process such information received by the transceiver through a microcontroller. Each sensor node depends on a source of energy that enables the devices to keep running. The energy in sensor nodes is limited and yields from small batteries attached to the device. Taking into account the attached battery as an energy source along with the tiny device size, the node itself has constraint processing, memory, and bandwidth for exchanging the data [17, 21, 22, 23].

2.1.2 Application Areas

The ability of WSN to work in any environment by sensing real-time data adds applicability to a broad spectrum of applications starting from environmental to complex industrial process automation. One of the early adoptions of sensor networks is in the military domain, the project from Defense Advanced Research Project Agency (DARPA)[24]. Today, there are limitless applications built up using WSN; environmental monitoring, healthcare applications [25, 26], smart home, agriculture [27, 28], industrial application [29, 30, 31], energy control system, and other commercial applications [24, 32] as we have summarized in Figure 2.2.

Among the benefits of using WSN is the possibility to self-organize and create a multi-hop network. In case of a node failure, they can be fault tolerant. Since nodes can be mobile, each node can quickly enter and abandon the network. From an economic perspective, they are cheaper in today's market. The price of a single WSN node was around $ 5 in 2021 [33]. All these features make them very advantageous compared to other technologies.

2.1.3 Middleware for Wireless Sensor Networks

The emerging need for developing applications using WSN requires new software and tools that facilitate the integration and communication of sensor platforms (which are distributed), gateways (nodes), and end-user applications. Such tools and software will allow management of the complexities by providing abstractions at different levels, facilitating the development process, and reducing programming overhead. Indeed these tools are referred to as middleware and are often used as the solution that stands between the sensor layer and application layer and ensures effective communication among them.

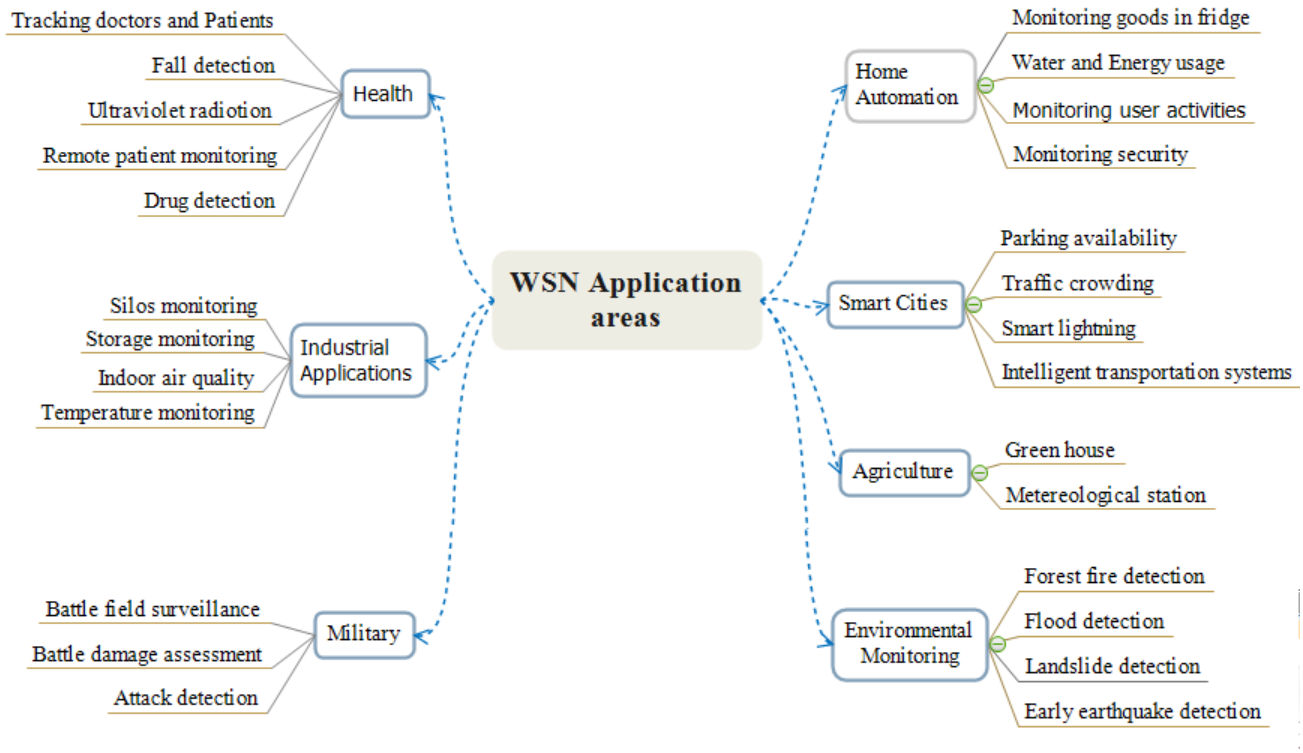


Figure 2.2: WSN Application areas

What is middleware?

The term middleware dates back to 1970 that is found in the edition of the Dictionary of Computers [34]. It refers to middleware as the software which facilitates the installation and development process not provided by operating systems. The term was not popular until the 1990's, and rarely can be found in the literature. In the early 1990s, the term was mainly associated with relational databases, while later, this was not the case [35, 36]. After this period, the term middleware was used to signify the software layer between operating systems and applications and provides abstractions that overcome heterogeneity and hide the complexities of distributed systems.

Campbell [36] uses the notion of middleware to describe the software layer, which stands between distributed systems and the application layer. It can be seen in Bakken [37], who identifies the primary functions of middleware as a layer that hides the complexities and overcome the heterogeneity of inherently distributed systems. A software layer between applications and operating systems that eases the programming process by providing abstractions to transactions and remote procedure invocation is defined as middleware [38]. We can see diverse and versatile definitions of the middleware notion. In most cases, it is used as a software layer that abstracts, integrates, facilitates, and connects one or many applications with the underlying hardware layer.

Considering heterogeneous communication protocols and the diversity of sensor platforms

from the open market (including different operating systems), there are particular needs for WSN middleware discussed in [39]. In the context of the discussion, Atzori et al. [40] suggest three research areas to be tackled in realizing the vision of IoT; sensors, middleware, and the Knowledge base, which must act together toward the vision of connected things in the Internet. Generally speaking, challenges that easily connect things and simplify the M2M communication and understanding, which easily allow the development of domain-specific applications, will lead to a desired state of IoT. In order to allow this to happen, there is a compelling demand for layers of abstractions in the middleware. Still, too much abstraction often brings complexities and difficulties[1]. Specifically, this can be a detriment to performance and flexibility. These issues are also echoed in [41], who identifies challenges intended for WSN middleware, including support in the development, deployment, maintenance, and execution. Hlabishi et al. [42], from WSN middleware perspective, acknowledge network management and heterogeneous-sensor networks as solid issues that need to deal with. All because WSN includes node movement and route changes in connecting neighbors [43]. Therefore, the concern for node abstraction is also noted in [44]. Katasanov et al. [45] utilize agent technologies as a model to deal with the preceding criteria, while semantic technologies for enabling interoperability among devices/machines.

Hadim and Nadir in [46] state that a middleware layer for a WSN network dictated by WSN characteristics, in order to be successful, must be able to manage limited power low energy sensor nodes, their communication and offer support to thousands of connected nodes that might be mobile. Furthermore, it must support the management of highly dynamic networks (many regional networks) and enable heterogeneous hardware solutions with diverse communication protocols, dynamic network organization, and real-time services. Finally, data integration, aggregation, and security are significant factors to take into consideration. Meanwhile, Romer et al. [47] suggest that WSN middleware features should integrate diverse applications by providing application knowledge in nodes, integrating adaptive fidelity algorithms, and supporting time and location data. Additionally, Yang et al. [48] adds the localization algorithms and performance requirements as design middleware issues. Razzaque et al. [49] in the architectural requirements of IoT middleware suggest; interoperability (network, syntactic, and semantic interoperability), programming abstraction, autonomy, adaptivity, context-awareness, and distribution requirements.

Traditional distributed middlewares like CORBA and DCOM do not comply with the requirements of WSN network [50] and thus are not suitable for use in this particular environment. Mainly, the communication in these architectures follows a synchronous model relying on "request/response," which is different since WSN, in most cases, follows an event-driven asynchronous model and is often prone to node failure due to unexpected changes or energy loss.

Recent proposals of middleware architectures aim to optimize resources, offer scalable solutions in a highly dynamic environment and address interoperability issues by exploiting semantic technologies. In the same way, they aim to facilitate WSN application development by providing levels of abstraction. The following section provides a classification of WSN middlewares architectures proposals.

[1]https://www.joelonsoftware.com/2002/11/11/the-law-of-leaky-abstractions/

2.2 WSN/IoT requirements

2.2.1 Architectural Requirements

1. Dynamics and reorganization - Nodes in the network may happen to be mobile, and they often enter and leave the network in an unpredicted way. In such cases, the SN should be able to adapt and self-organize its structure to offer flexibility and availability of resources, including the sensed data.

2. Context-aware - The system continuously monitors the physical object, attaches necessary data for time, location, and space, and acts autonomously or even suggests to users possible actions based on the perceived situation. In complex domains such as air traffic control, automobile driving, and emergency services, an entity must recognize the entities' operations (nodes) in a Cyber-Physical System.

3. Location-aware - Location awareness is the ability of a sensor node to identify its position in relation to a particular point of reference using coordinates.

4. Distributed - WSN network covers a wide area by distributing the nodes all over the field that need to be connected.

5. Autonomy and Self-organization - is required to extend the IoT at a global scale. Middleware could perform tasks without external input, optimize resources and expand automatically depending on the need. In such cases, overlay networks and algorithms achieving self* properties must be implemented in the middleware to enhance performance and provide semi/full autonomous systems. For instance, if a node generates false values, the system may detect and stop sending erroneous data, indicating a self-optimized network by preserving energy usage. Self-organization is another valuable property where a collection of entities/agents interact and coordinate their actions to achieve the global goal [51] efficiently.

6. Interdependence [18] and distributed data processing - Sometimes, cooperation between nodes is necessary to achieve the desired result for a particular task. Such sensor nodes should be able to coordinate their activities when operating in the environment.

7. Proactiveness - building an intelligent system does not simply mean that an entity acts in response to the environment but has a goal-oriented behavior that will know when to initiate its actions [15].

2.2.2 Functional Requirements

1. Resource-constrained - sensor nodes are small devices with limited memory capacity, processing, and bandwidth.

2. Energy Management - all WSN nodes depend on battery power, and the quick harvest of energy means no support for communication. Energy management is a critical

factor that should be considered when building intelligent systems using WSN. Recent research studies aim to optimize energy consumption on constrained sensory devices [34, 35, 36].

3. Device Discovery and Management - each device deployed in the network should allow other devices to know its presence and vice-versa; all neighboring devices connected in the network are reachable. When discovery is made, the middleware allows requesting the services offered by the sensory device.

2.2.3 Non-functional Requirements

1. Interoperability - is required to handle the dynamics and explore the business value of continuously generated data. The interoperability can be achieved when end-users, applications, and primarily machines commonly understand the meaning of data and can efficiently process and interpret those data [52]. Agent technologies should adopt semantic web technologies to facilitate decision-making, ease the discovery of resources, and provide accurate results to end-user queries. This interoperability sought to be; a) syntactic and b) semantic.

2. Mobility - middleware requires to support network dynamics and movement of sensor nodes. Each node is connected wirelessly and can move within the network range.

3. Fault Tolerance - It is required to ensure normal functioning due to power restrictions or hostile environments where the networks are deployed.

4. Ultra Scalability - Middleware should be capable of dynamically injecting the service functionalities, providing service discovery and composition, and enabling efficient communication with other entities in the network and the Internet. This way, algorithms must process voluminously generated data by sensors, and the persistence layer needs to store those data and immediately respond with accurate results to end users' queries. Furthermore, managing billions of connected things presents a significant challenge. The only way to keep the trend is to scale quickly with a set of rich behaviors and intelligent algorithms with self* properties and not burden developers with implementation tasks whenever a new device is added. Thus, scalability is required in all three dimensions: a) Size (increasing networks, users etc.) b) Geography (communication distance posing no problems), and c) Administration (optimization should occur without too much effort and should be automatic)

5. Heterogeneity - Different vendors produce sensory devices that may consist of different operating systems, that in turn demand abstractions from middleware for integrating each device in the network and providing the services to end users. The more devices are integrated, the more the demand for data interoperability is created. As the information generated by sensory devices might have different data models, a middleware should provide an interface that accepts the information streams and converts them

to a common standardized format that is accepted among all entities in the network. The heterogeneity should be considered in terms of a) hardware devices b) network, c) programming languages, and d) data formats.

6. Intelligence - the amount of data streams produced by the sensor network is exceptionally high. Designing a middleware that could bring Intelligence closer to the nodes, such as reducing transmission of unnecessary raw data and ensuring that the system is at an optimal level, is very desirable.

7. Robustness and openness - having open systems allows us to understand the technologies involved and could enable cooperation from diverse technologies, including devices from different systems. However, this will open the research direction with security and privacy. On the other hand, having a system that can operate continuously despite numerous failures that could happen (node failure, broken links, noisy and faulty physical world realities) is more than desired for future intelligent applications.

8. Concurrency - virtualization of sensory devices would give numerous applications access to the same device without the necessity to deploy similar devices for the same purpose.

2.2.4 Application Oriented Requirements

1. Diverse applications - a wide variety of applications would be able to access the middleware by using various technologies from different vendors. In that case, the middleware will also create the need to address the appropriate domain-specific requirements.

2. Real time - in a monitoring environment with the wireless sensor nodes, industries/ organizations collect real-time data about their interested "thing" properties and process in real-time that information that will result in an appropriate decision.

3. Precision - some specific domains, like industrial automation, smart automotive, healthcare, and other related industries, require high accuracy and no latency at all. For instance, a self-driving car requires data immediately without delay because of the actions to be taken instantly, otherwise, a crash might happen with other objects nearby. As a result, having very low latency and enough bandwidth to transmit the necessary data are crucial for building future smart apps.

4. Seamless connectivity and stability - The near future requires fast data processing techniques and high connection speed with quality service.

5. Security - nodes will collect heterogeneous information, including personal human data, their location, medical data, preferences, and similar sensitive data. It is necessary to build security protection at a different middleware level, including the regional networks: from the physical layer where we collect data, every communication encryption, and middleware level, if possible, the detection of an attack and their diagnosis, or privacy issues if they are broken.

6. Transport Systems Requirements

    - Delay sensitivity - Vehicles moving at high speed have a crucial requirement regarding communication delay, which could be in micro to milliseconds. In cases when bandwidth is restricted, and a delay occurs, the life of surrounding people would be at risk.
    - High-speed mobility - In the automation industry, it is critical to guarantee high precision using ad hoc connections at a very high speed.

7. Health Care

    - Confidentiality - saving patients' data requires a model that restricts data access by keeping data secret and ensuring confidentiality.
    - Data Visualization - data visualizing techniques that professional caregivers understands still need progress. Data will be meaningless if the visualization does not follow a certain standard.
    - Trustworthiness - providing high-quality measurements and data delivery with no latency. In healthcare, the level of trustworthiness must be very high. Trustworthiness has several other consequences concerning radio communication which can lead to additional loss of packets or noisy data when metal doors and walls are present.

8. Marine Environment

    - Higher Water resistance - the more profound the water level, the stronger the resistance.
    - Strong robustness - sensor nodes need to work in an environment with high waves, typhoons, and different tides.
    - Higher energy consumption occurs due to long distances in communicating the data while monitoring the environment (ship mobility, container tracking, etc.)
    - Sensor coverage issues - the maritime environment is inadequate for deploying sensing devices in ocean areas, which might result in coverage problems.
    - Unstable line-of-sight - between transmitter and receiver becomes critical due to antenna oscillation, waves, and moving "objects."

9. Industry Challenges

    - Mixing legacy infrastructure with IoT - Industry is still not ready to modernize all of its infrastructure with modern IoT, and one of the challenges is to find solutions that both; legacy and modern infrastructure are able to operate together. The legacy issue indicates the need to leverage data by producing the necessary tools for integrating the data from both; systems and new tools that efficiently process data.

- Unpredicted situations - further development of intelligent algorithms is necessary to respond to unpredicted behaviors from industrial machines.
- High-Speed Internet Protocols - information transformation from machine to machine should be higher than what current protocols are offering.
- Cyber security and property security - today, the industry faces security challenges and cyber-attacks that could cause an economic loss. Protecting products, customers, marketing strategic data, and even more to avoid the misbehavior of machines requires additional security enhancements.

10. Smart City

    - Long range communication - Smart city consists of a very noisy environment that creates signal interferences and provides environmental barriers with several other obstacles. It requires long-range communication to devastate such issues.
    - Bandwidth consumption - smart city produces tons of data that will consume the bandwidth fast.
    - Ultra-dense device deployments - due to dynamism, noisy environment, crowds, and obstacles appearing in the city, the need for dense sensor node deployment is required.
    - Functional extensibility - has to do with adding new functionalities/services to the existing ones.

WSN is characterized by many sensory devices and network protocols (Zigbee, Xbee, 5G, Bluetooth, etc.) with applicability to a wide range of domains. Challenges already present on the Internet are valid to WSN, but there are far more issues because of the characteristics and limitations imposed by WSN. Those challenges are summarized in the following section.

2.3 Challenges of Dynamic WSN/IoT

As the previous section introduces the general challenges of IoT, for the purpose of this study, the challenges will be categorized according to the definition of dynamic systems. Thus, the following section discusses the challenges of the dynamic system:

- **Ultra Scalability** - Middleware should be capable of scaling in several perspectives:
    - Functional scalability - is the ability of the middleware to easily add new functionalities and scale with new services and new functionalities with minimum effort.
    - Geographical scalability (GS) - in complex systems, GS represents the ability of middleware to maintain performance and efficiency regardless of when it is expanded from a local area to a distributed geographic pattern. It also means that the distance of components will perform well when communicating and collaborating.

  - Load scalability - presents the capability of the distributed system to create a balanced load when the number of inputs is growing or when the number of inputs is light. It shows how easily it accommodates the load of the middleware architecture by modifying components, adding new ones, or removing them.
  - Administrative scalability - is the capability of the middleware architecture to work for a number of organizations/institutions/users.

- **Heterogeneity -** Heterogeneity is a critical aspect of IoT, where complex systems often need to sustain with many diverse standards, devices, and applications. So starting from the hardware up to the application level, the heterogeneity includes:
  - Heterogeneous devices - multiple vendors produce devices with different capacities including devices addressing application-specific requirements.
  - Heterogeneous data streams - information can be in binary, plain text, or structured, such as JSON or XML. Thus, the complex system needs to access such information and be able to process them.
  - Communication protocols - since WiFi cannot be directly applied to sensory devices with already known limitations (memory, processing, and energy), a variety of lightweight wireless protocols exists and aim to work in such resource-limited devices.
  - Operating system - as multiple vendors are producing sensory devices, there may be cases where different operating systems are used by different vendors, which also indicates that different programming languages may be used.

- **Autonomy** - the system should adapt to unpredictable changes while also hiding its complexity from users. It should perform its tasks without external input and do them automatically [53]. IBM proposes four properties for autonomic computing: self-healing, self-optimization, self-configuration, and self-protection [54]. The self* properties are expanded further by Prosland [54] and Nami and Bertel [55].
  - Self-organization: patterns arising at a global system arise from the interactions of low-level components [53]. It allows the creation of a structured network for nodes who enter the network spontaneously.
  - Self-adaptation: the system is able to adapt itself to changing conditions without too much effort.
  - Self-healing: discovery from faults in an automatic manner.
  - Self-optimizing: automatic optimization of the resources to function properly as defined in the requirements.

- **Emergent properties** [56]: complex systems poses emergent properties which arise by studying the system overall and not parts individually. More than a complete overview of the individuals is required to predict the system's behavior overall. In IoT, the cases of autonomous driving cars, smart traffic lights, smart grids, etc., are

complex systems that need emergent properties [57]. One requirement to be fulfilled as an emergent behavior is to have self-organization property and, specifically, realize autonomic properties.

- **Robustness -** A critical factor of the middleware is to react to disruptions of broken links and node failures. Failing nodes have an immediate effect on the accuracy of the received values that may require new settings for the nodes or new network configuration in case of node failure.

- **Interoperability -** is required to handle the dynamics and explore the business value of continuously generated data. Interoperability (syntactic and semantic) is achieved when end-users, applications, and most machines typically understand the meaning of data and efficiently process and interpret those data. This interoperability sought to be;

  - Syntactic interoperability: a common structured data format like XML or JSON enabling machine processing of information.
  - Semantic interoperability: the ability to share the meaning in machine-to-machine communication.

- **Energy efficiency -** due to the dynamic environment, node mobility, and small-sized power supply, the demand for energy optimization and efficient energy usage in wireless sensor networks is crucial. To have energy efficiency, several domain problems need to be touched; data transmission, routing of information, processing, and localization strategy. For instance, transferring the data to the failing nodes requires retransmission [71]. Recently, much investigation has been done to improve the energy efficiency in sensor networks.

- **Memory and Processing -** The miniaturization of the sensory devices and the energy limitations directly affect the processing and storage of these devices (which are constrained). Storing the data internally is impossible, while processing the data is too limited.

2.4 Architectural Middleware Classification for WSN

A demand prevails to classify a taxonomy and check the advantages and disadvantages of each architectural approach that can work in a highly dynamic environment in delivering middleware services for WSN. This classification derives from an investigation of the research literature of previous reviews in the middleware approaches for WSN network [49, 58, 59, 60, 61, 62]. This categorization reflects how closely each architectural approach is able to address the issues and challenges of sensor networks and their potential for future applications.

This categorization includes:

- **Database Oriented.** The entire sensor network is seen as a virtual distributed database.

- **Service Oriented.** The design paradigm follows SOA architecture approach, where the applications are built in the form of services.
- **Agent Oriented.** Agent-oriented programming (AOP) is used to develop modular programs for WSN control and coordination.
- **Virtual machine based.** Virtual Machine is placed on nodes within the network, allowing programming support and providing abstractions of the OS.
- **Tuple Space oriented.** Each sensor node in the network acts as a shared memory consisting of tuples that can be accessed concurrently by other nodes in the network.
- **Event based.** In event-based middleware, components, applications, and all the other participants interact through events. Each event has a type, as well as a set of typed parameters whose specific values describe the specific change to the producer's state.
- **Rule based.** In the rule-based paradigm, the execution of an instruction is based on the rules or knowledge applied to the initial receiving data to inherit new information.

The following section examines the previous work investigated in this area which summarizes the approaches identified by different research studies in Table 2.1. Then, we will provide the pros and cons of each paradigm which details the reasoning of this thesis regarding the most suitable paradigm for Wireless Sensor Networks.

Table 2.1: Taxonomy for Wireless Sensor Network middleware

Resources	**DB**	**Service Orien.**	**Tuple**	**Agent**	**VM**	**Event-based**	**Rule-based**	**App. driven**	**other**
Molla et al. [59] :2006	✓	✓	✓			✓			
Sugihara et al. [63] :2008	✓				✓				✓
Razzaque et al. [49] :2016	✓	✓	✓	✓	✓	✓		✓	
Wang et al. [62] :2008	✓			✓		✓	✓		✓
Hadim & Mohamed [46] :2006	✓			✓	✓			✓	
Kuorilehto et al. [64] :2005	✓				✓			✓	✓
Li & Moh [65] :2014	✓	✓			✓	✓		✓	
Chelloug et al. [61] :2017	✓	✓		✓					

Kuerilehto et al. [64] distinct operating system, middleware, and virtual machine-based architectures. Operating system and VM architectures mainly operate at the node level, while the middleware performs at the network level. It follows that the middleware architectures are classified into database approach, application QoS and based on the context of the surrounding environment, which are implemented using tuple spaces or mobile agents. Tuple spaces are considered communication methods for task assignments rather than an architectural model.

Hadim & Mohamed [46] propose another classification for sensor networks based on programming support and programming abstraction. The latter describes the sensor network

as we view it, a layer over the network that changes as we view it. Programming support classifies; VM machine middleware, database, modular (agent-oriented), application-driven, and message-oriented approaches. Contrary to Kuerilehto et al. [64], the VM approach is represented as a middleware approach that allows modular programming and distributed modules throughout the network using tailored algorithms, which later the VM interprets the modules. Database abstraction perceives the whole network as a virtual database, which allows users to unleash queries via a friendly interface and get replies that most often are approximate results. The message-oriented approach allows asynchronous communication in a distributed sensor network by employing a publish-subscribe mechanism.

A broader classification is given by Razzaque et al. [49], which categorizes middlewares according to the design approach. The classification consists of: a) event-based, b) service-oriented, c) VM-based, d) agent-based, e) tuple-spaces, f) database-oriented, and g) application-specific. Database-oriented, agent-based, and tuple spaces are previously presented in the above sections. Message-oriented approaches are treated under the event-based approaches. The events are disseminated from sensor nodes to the end-user application programs. Most commonly, the communication model is publish/subscribe mechanism that is also part of message-oriented middleware and thus falls in the same category. On the other hand, service-oriented approaches are built following the SOA approach where loose coupling, service compose-ability, and remote access are integral to these middlewares.

Another mode not discussed previously is the one noted by Wang et al. [62], which follows a rule-based approach and is represented by FACTS middleware [66]. Likewise tuple-space approach, each node acts as a shared memory, structuring the data in tuples and encompassing local rules. Then, a component consisting of a set of rules allows the execution of occurrences by receiving the information from local nodes. Authors in [62] give a more rigorous taxonomy of the architectures for WSN. They divide into programming abstractions and implementation features. The first involves the design approach to be tackled, while the second provides implementation features, including coordination and context features.

Other research reviews ([59, 61, 63, 65]) provide a similar classification of middleware for WSN, all of which are previously discussed.

2.4.1 WSN Middleware Advantages & Disadvantages

The middleware design approaches discussed previously provide a means to develop high-level applications in the domain of the WSN network easily. Each paradigm is distinguished for its own powerful features and has its pros and cons where Figure 2.3 summarizes the advantages and disadvantages of each paradigm, which will also be discussed according to the criteria previously identified in the literature while addressing the important issues of a generic (not application-specific) WSN middleware addressed in the Introduction section of this report; dynamic environment, ultra-scalability, heterogeneity, robustness, autonomy and node constraints. First, we will discuss the paradigm in general, which would best fit the criteria specified, and then a detailed overview of the middlewares following such design approach is discussed.

In designing the Ubiroad middleware, Terziyan et al. [67] identifies interoperability, mo-

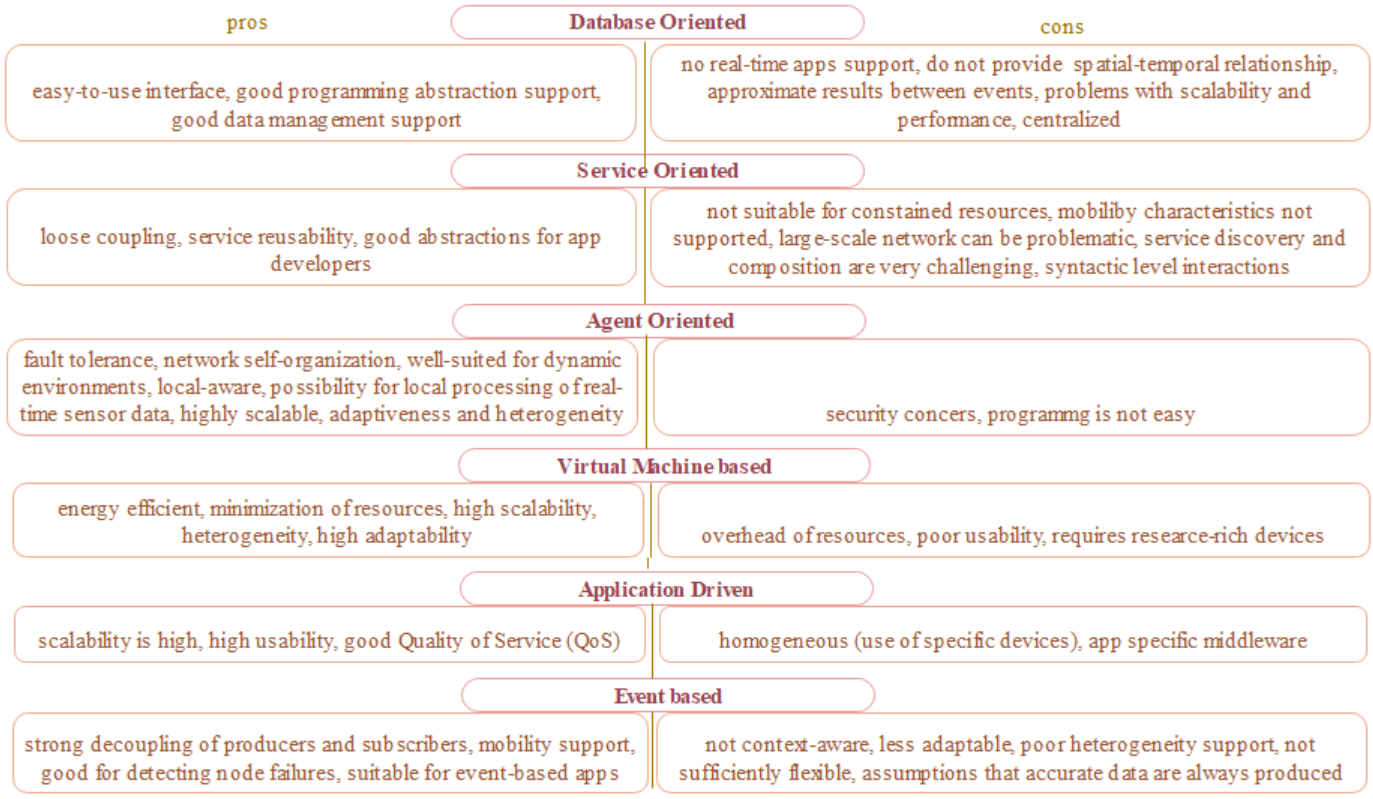


Figure 2.3: Middleware Design Approaches Advantages and Disadvantages

bility, and heterogeneity as middleware design feature that needs to provide mobile services to the end users. Following that, as the number of devices continually increases and along with that increases the management of such massive growth, authors in [68] propose autonomic and resilient service provisioning support by the middleware.

It is evident that current middlewares are only able to address some of the WSN requirements and challenges imposed by their characteristics and application domains they operate. For instance, Razaque et al. [49] identifies several architectural challenges that are not/partly provided by current middlewares, such as interoperability, autonomy, adaptiveness, and programming abstractions. Alternatively, Anne et al. [69] remark service discovery, scalability, and security & privacy as concerns for the middlewares.

Of course, the next generation of IoT should be able to deal with all of the above design issues for enabling next-generation IoT systems. Notably, the super-massive increase of connected things will require autonomy for systems that must have self-management but also remove complexity barriers for further growth.

The following table summarizes middleware paradigm comparison mainly based on the dynamic challenges of IoT.

Table 2.2: Comparison Criteria for the WSN paradigms

DB	C, E, NRT
Service-oriented	D, S, SI, SRT, DA (hard coded)
VM	D, HS, SRT, S, partially RA
Agent-based	DA, Aut, R, SRT, D, SO, HS, E, SI, partially RA
Event-based	C, E, S, HRT,
Application-based	Aut, HRT, C, D, HS
Legend	Aut: Autonomous, E: Energy efficient, C: Centralized A: Availability, DA: Dynamically adaptive, SRT: soft real-time, HRT: hard real time HS/S: Highly Scalable/Scalable SO: self-organized, D: decentralized SemI: Semantic Interoperability NRT: Not real time SI: syntactic interoperability

2.4.2 Paradigm Satisfaction Desiderata

Previous sections identified a set of paradigms, discussed their advantages and disadvantages, and explicitly defined the criteria under which the middleware paradigm is compared. Table 2.2 recaps the supported requirement of each middleware paradigm. It can be concluded that service-oriented, agent-based, application-oriented, and VM-based address more specified requirements for the purpose of this study.

DB approach. Centralized approaches are commonly adopted by the industry [70] due to operational costs and low capital. All the resources reside in a primary data center where IoT users connect remotely to a registry in the form of a client-server and query the data from the central database. They make it possible to respond to user queries and extract meaningful information, most often with approximate results. The most popular is GSN [71], which is also implemented in other projects like OpenIoT [72]. A great feature of this architecture is the Plug & Play capability. Despite the success in the industry because of operational cost and historical data, they provide, today, WSN applications require real-time monitoring, whereas centralized middlewares are timeliness[73]. Next, in the centralized solutions, it takes a lot of work to handle scalable sensor network dynamics due to the mobility of nodes and environment dynamics which often may require local collaboration to accomplish a task based on the collected data. For instance, recently, we have self-driving cars that requires to instantly make decisions locally, rather than waiting for someone to press a button and wait for the following action. It is also challenging to handle and manage applications and services dispersed geographically. Finally, the main data center becomes a single point of failure, and attack [74].

VM approach. VM-based are very good at providing abstraction and scalability at the application level. It allows the developers to develop applications in small modules, which are then distributed throughout the network. Considering that each node holds a VM that interprets the modules [49], the nodes themselves should be resource-rich enough to run the VM modules correctly [75]. On the other hand, installing VM would minimize energy consumption and resource usage [76]. Overall, VM architectures, together with service-oriented approaches, follow a predefined and deterministic mechanism for resource composition, which does not scale well in ultra-scale and dynamic environments [49].

Event-based. In the event-based approach, applications receive data transmission when a specific event occurs. Generally, events consist of a set of typed parameters that determines whether there is a change in the state of the component being monitored. Usually, the sensed information of the producer is assumed to be accurate. The kind of event approach is message-oriented middleware which depends on the publish/subscribe mechanism. The architectural model follows an asynchronous communication model, which is also considered more suitable than the synchronous communication model [46]. Most of the requirements addressed by even-based approaches cover non-functional requirements; real-time performance, availability, scalability, resilience, and security [77]. These imply to perform well in mobile and reactive apps, while the support for interoperability, context-awareness, and adaptability are limited [49].

Application-oriented. Application-oriented middlewares are based on a set of requirements of a specific domain that try to satisfy the quality of service by following an architecture that fits the network of a specific application. Building another application for a different domain requires addressing new requirements and even starting implementation from the beginning.

Service-oriented. Presents an industry supported standard technology. The SOA, together with Agent-oriented approaches, facilitates the modeling and development of today's complex systems by offering modular solutions (separating the concerns into smaller pieces); one through services and later through agents. SOA would provide better scalability through load balancing by creating individual clusters, while the Agent-oriented paradigm (AOP) is known for scalable solutions. Following the SOA-based architecture seems inconvenient for Sensor Networks due to the node limitations. As a result, it fails to comply with SOA requirements for high processing power and memory for message processing [69]. Further, SOA research challenges such as heterogeneity, mobility, adaptability, awareness, security, and privacy [22] should be further addressed.

Agent-based. The agent-based programming paradigm is highly flexible and has a lot to offer to the IoT. There is evidence from studies [78, 79] that this is the most suitable technology to meet the IoT needs for building robust, complex, real-life enterprise applications. In a distributed environment like IoT, complexity and scalability are handled by separating its components and decomposing the larger problem into subproblems by assigning each subproblem to an agent. Also, agents can coordinate the tasks accordingly and provide efficient solutions via distributed computing, bandwidth, and power usage [80].

In an environment where agents operate, the ability to adapt to a rapidly changing environment and dynamics which consist of limited resources [80, 81] is high. Agents might know in a precise manner with whom, and when to collaborate, so the goal is accomplished. A natural way of modeling complex systems is through small autonomous components interacting with each other towards the goal achievement [82]. The more autonomous components are built, the better management of complexity. Usually, various software modules (or mobile agents) in distributed systems run on the distinct local network and become part of that network throughout their lifetime. Due to the mobility of agents [82], there is potential to change the state dynamically depending on the environment and thus capable of performing well relevant to the new changing environment.

Another dimension where agents play a significant role is the ability to do data processing and data extraction in sensor nodes. Data processing near the nodes filters the unnecessary raw data and further turns the raw data into information in a region of interest. The possession of such properties allows global energy management for the whole network [83].

Although AOP and SOA present the most promising paradigm by addressing most of the requirements of today's IoT, the differences are as follows: Regarding Interactions, the difference between Service-oriented and Agen-oriented approaches is the difference between syntactic and semantic level interactions. SOA interactions are bound mainly at "syntactic level"; a service invokes a function that is exposed by another function, and after a given time, it retrieves the results [82]. In the agent-oriented paradigm, the interactions are bound to semantic and knowledge levels. The semantic interaction allows agents to communicate and engage with the problems cooperatively until the desired results are achieved. Consequently, we may give more responsibilities and create more autonomous systems so that human intervention remains at the lowest level possible in our future e-business applications and enterprise integration.

Admittedly, agents have the ability to provide autonomy in WSN systems if the information is shared and appropriately processed. A very promising approach for the system's autonomy plays knowledge representation languages [84], which allows modeling, representing, and processing of the environment and sensed information. In particular, the combination of an agent-based paradigm with semantic Web and Web services is very promising for the future of IoT. There are several studies [85, 86] showing the relevance and benefits of such a combination.

2.5 Agent oriented Middleware for Wireless Sensor Networks

An agent is an autonomous software that is capable of performing some actions with a certain degree of autonomy in a particular environment in order to accomplish the given tasks. In Agent-oriented middleware, multiple software agents are dispersed throughout the network and enable the execution of simultaneous applications. Agents are likely to store their state while moving from one node to another node and thus making them very suitable for decentralized systems that tolerate partial failure [87]. In Wireless Sensor Network, an agent can represent real-life entities such as a doctor in a hospital, a temperature monitoring center, a self-driving car, etc. For instance, when querying a particular truck, an agent would give its location, depicting the truck on the road. It is demonstrated that agents can also reduce data transmission and thus save up roughly 90% of data transfer time [88]. The energy savings happens because the data will be transferred only when an agent requires it. It indicates that they eliminate the data redundancy and data transfer overhead within the network.

We already emphasized the importance of agent-oriented architectures for Wireless Sensor Networks using particular criteria and the advantages compared to other paradigms. The current section elaborates on agent-oriented architectures that help us to find the best

architecture later in chapter 3 according to the criteria described in section 2.3.

2.5.1 Multi-agent based approach

Multi-agent approach provides the necessary tools and methods to model complex distributed systems, such as the case of WSN, with some capabilities in performing tasks either individually or collaboratively under certain rules or by learning via input gathered from the monitoring environment. Thus, a multi-agent system consists of numerous software agents that collaboratively solve the tasks, operating beyond their individual capacities [89]. Each agent likely performs processing tasks individually, reasons over the data, and can provide visionary solutions via intelligent algorithms integrated into each agent or via intelligent algorithms capable of solving complex issues provided at the global level, resulting from facts exchanged by local agents.

Thus, knowledge sharing among agents promotes better processing time, task realization, energy savings, and bandwidth utilization. They also help in selecting the optimal path for information traveling in a dynamically modifying network structure. The above statement shows better utilization of WSN resources and contributes to better decision-making in distributed and dynamic environments. For instance, in (semi)autonomous vehicles, agents efficiently process sensed local data and share the knowledge with other agents for better decision-making by avoiding possible accidents or in a crowded city; they allow to select the best route to arrive faster at the destination.

2.5.2 Bio-inspired approach

Biological systems and nature are a great source of inspiration for designing, developing, and applying biological and chemical principles in several engineering problems, which are introduced for the first time by Eigen [90]. These principles come from chemical systems by studying nanostructures of biological molecules or studying interactions of proteins in diverse surfaces [91] and biological systems by studying the swarm behaviors. Self-organization biological systems provide interesting properties that emerge spontaneously via local interactions of individual elements in order to solve complex problems at the global level. For instance, ants are limited in the functions they perform. Nevertheless, they are able to solve very complex problems when they function in colonies, capable of creating superhighways while carrying food. These genuine, spontaneous, and self-organization behaviors have found a suitable solution to a wide range spectrum of applications. Specifically, many algorithms and tools have emerged and been applied to various problems in WSN. Dressler and Akan [92] classify Bio-inspired solutions in three main categories; a) Bio-inspired computing consisting of optimization algorithms, b) Bio-inspired systems that deal with architectural paradigms, and c) Bio-inspired networking for building large-scale, distributed applications. We mainly study bio-inspired solutions from an architectural perspective addressing WSN requirements.

2.5.3 Existing Agent-Oriented Architectures for WSN

In what follows, we list the agent-oriented WSN architectures extracted from the literature review.

Agilla [93] is an agent-based middleware designed to support self-adaptive applications. Agilla is built on top of TinyOS and Mate. It is composed of moving agents that save their current state using tuple space. The autonomy of each agent is guaranteed by the use of local tuple spaces where every node in the network consists of tuples that make it possible for other agents who visit that particular node to read/update the saved information. It makes use of a template pattern for gathering information in tuple spaces. The main idea of Agilla was to deploy on a network with no pre-installed apps. In this middleware, agents adapt the application's behavior from the environment they operate and reconfigure it. Also, agents might interact with nearby agents or remotely access information using tuple spaces. Apparently, agents process information and make decisions locally, reducing the data communication and making Agilla an appropriate solution for nodes requiring energy efficiency. However, programmability and code management are very challenging due to low-level language abstractions.

Impala [94] is a middleware designed explicitly for ZetbraNet project, which is a wildlife tracking project for tracking nodes and peer-to-peer communication techniques. Impala allows applications to achieve modularity, adaptability, and repairability in WSN via efficient updates that come into modular pieces on the fly (through a wireless transceiver).

Impala is suitable for sensor networks that are deployed in a harsh environment that requires minor updates but not too frequently since the overhead is present when delivering a specific event. It follows that they are efficient when there is a need for complete software upgrades or re-installation.

ActorNet [95] is a mobile agent platform for WSN designed to support concurrent applications in constraint sensor nodes. It provides code migration functionalities similar to Agilla but with higher efficiency and interoperability support. In addition to the above, ActorNet provides virtual memory, context switching, and multitasking for better execution support when multiple applications access nodes with limited resources. A drawback of ActorNet is fault tolerance that comes from the service discovery mechanism, which is a local broadcast protocol that requires a node to know all the neighboring nodes, which introduces an extra overhead in the network.

UbiRoad [96] presents a semantic middleware following an agent-driven architecture for context-aware smart road environments. This middleware includes flexible collaboration between smart road devices and services and the interoperability among heterogeneous in-car and roadside devices. Semantic interoperability is obtained with regards to a) data-level interoperability by developing a paradigm of resource-oriented networking and b) protocol-level interoperability by utilizing context-aware, adaptable and reconfigurable composite

service networks. In addition, agents in Ubiroad allow monitoring of various components of the network, which by monitoring resources internally, including the interactions with other resources in a smart road environment, contributes to self-management capabilities.

MAPS [97] is an agent-oriented middleware for WSN applications based on Sun SPOT technology. MAPS offers a set of services that supports the management of agents and services for easy access to the node resources. Agents' actions follow the multi-plane ECA-based model where each action is triggered by certain events under certain rules. MAPS framework is particularly designed for devices with limited resources and suits well for Sun SPOT nodes that are Java programming compliant.

MASPOT [98] extends the abstraction of MAPS and adds the capability for code migration to Sun SPOT sensor nodes. Providing migration capability brings a better range of applicability of Sun SPOT sensor nodes compared to MAPS. Compared to MAPS, agents' communication in MASPOT is fulfilled via a broadcast protocol which introduces an extra overhead in the network.

TinyMAPS [99] is a Java-based framework designed to work with the Sentilla sensor platform. It derives from MAPS middleware by enhancing functionalities to work with nodes having more limited capabilities compared to Sun SPOT. TinyMAPS is less efficient than MAPS when agents communicate with each other but improve migration when the data payload is low (under 58 bytes).

UBIWARE [100] is a middleware for integrating heterogeneous IoT resources and providing semantic interoperability between the connected components and the applications. It follows an agent-oriented paradigm that is developed on top of Java Agent Development Framework (JADE [2]). UBIWARE allows the development of complex, distributed systems that are flexible and extensible. One of the main principles is to support automatic discovery, followed by orchestration, choreography, invocation, and execution of various meaningful services. Although it does not look ideal for Wireless Sensor Networks, it seems a promising approach for IoT by addressing several IoT requirements.

Emergent Distributed Bio-Organization (EDBO) [74] presents a bio-inspired agent-oriented middleware for WSN. Inspired by the concept of gaining the desired properties at a higher level through the interactions happening at a lower level, EDBO aims to tackle most of the challenges of distributed systems and today's IoT requirements. Similar to UBIWARE, it intends to discover resources from decentralized but unstructured networks. UBIWARE follows a proactive routing strategy, while EDBO utilizes emergent behaviors. Discovery in EDBO middleware is achieved via BioBot, which can interact with the cyber-physical system (CPS) and handle the end user queries. BioBots can interact with each other in order to achieve the desired result. Furthermore, EDBO has the ability to offer a scalable, robust

[2]http://jade.tilab.com/

solution in distributed systems without explicit engineering using bio-inspired algorithms for replication and organization of the activities.

Self-Adaptive Middleware for Wireless Sensor Networks (SAMSON) [101] is a middleware for Wireless Sensor Networks that addresses the autonomic model requirements such as self-configuration, self-healing, and self-optimization and, as a result, will facilitate the creation of self-adaptive WSN middleware applications. It follows the MAPE-K [102] model proposed by IBM for autonomic computing. Their reference architecture (RA) allows them to map elements of the RA to software components and generate source code for the sink nodes and nodes in the WSN network. The code is platform-specific and can be deployed to nodes running the Contiki operating system. The SAMSON self-adaptivity takes into account the nodes with faults, disconnection, and power concerns.

Distributed Internet-like Architecture for Things (DIAT) [103] propose a distributed layered architecture able to tackle the challenges of automation through dynamic service creation, security, zero-configuration, heterogeneity, and interoperability. The idea is to run the so-called "IoT daemon" in every object that consists of some processing power and memory. This daemon is divided into three layers: virtual object layer (VOL), composite virtual object layer, and service layer. The first layer (VOL) of the IoT daemon deployed in device A can communicate with the second layer of each daemon deployed in other devices. This layer allows for overcoming the interoperability issues among the system. In situations where limited capabilities exist, such as WSN, the daemon provides limitations to the features that can be offered to those particular nodes. However, DIAT allows the creation of dynamic services with some "manual" configurations in the platform.

EAGILLA[104] is an enhanced alternative to Agilla middleware. The Eagilla middleware supports mobile agents that can move around the nodes (not fixed like agilla). The communication of moving agents goes on globally by supporting multicasting in WSNs. This communication between agents and remote agents is similar to agilla and is based on tuple spaces. Eagilla is different from Agilla in supporting several nodes from different vendors, which addresses the heterogeneity issue of WSN. Eagilla is in its early stages, and it provides simulation results.

Sensomax [105] is an agent-oriented middleware dedicated to working with Sunspot nodes implemented with Java platforms (Java SE and ME). In Sensomax, it is possible to access nodes by multiple applications concurrently. Each application is treated as a process, and these processes can access all the resources. The resources are categorized as shared resources on a global level with all the network nodes, locally with members of the same cluster, which consist of cluster head and system properties. In Sensomax, every node can become a cluster node that can communicate with the other cluster nodes and the gateway. There is no distinction between nodes with resource constraints and nodes without constraints. Also, this middleware is application domain-specific, using Sunspot nodes.

Bisnet [106]. BISNET presents a biologically-inspired middleware for WSN. The middleware is inspired by the bee colony mechanism and implements certain agent behaviors; pheromone emission, energy exchange, replication, migration, and death. For instance, when an agent reads the sensor readings, it converts to energy which is shared among agents for maintaining a balance of energy and also used for pheromone emission that alerts other agents of environmental changes. When the pheromone exceeds a specific threshold value, agents can replicate themselves, and as a result, the monitoring node is replaced by the replicated agent while the parent moves to the neighboring node. The migration is used in multi-hop transmission for transmitting data to the base station, while agent death happens due to a lack of energy.

FIoT (Framework for IoT) [107]. Presents an agent-based framework for building self-adaptive and self-organized IoT systems. It is constructed using MAS and machine learning techniques for making decisions based on the collected information. The main focus of the middleware is to provide autonomy to the smart object in the sense that they do not need human administration. Each smart object is controlled by an agent who collects the data from it and sends the data to the agent controller. Based on the defined features of the problem domain (which should be provided by the designer), the adaptive agent executes and process the information. The framework looks promising by employing evolutionary and predictive algorithms for improving the performance and the adaption to the dynamics of IoT. On the other hand, the framework does not consider the constraints of the WSN and IoT. Smart objects are considered devices equipped with sensors, actuators, processing, and networking capabilities embedded with intelligence. These smart objects can either sense and interpret their own generated information or information from nearby devices.

ACOSO (Agent-based COoperating Smart Objects) [103]. ACOSO follows an agent-oriented and event-driven paradigm for monitoring cooperative smart objects (CSO) (similar to FIoT middleware). CSO in ACOSO utilizes the event-driven proactive architecture following a message-based and publish/subscribe communication model. The middleware layer which implements the CSO architecture is built using agent frameworks. ACOSO is specifically designed for developing and managing SO, which demands proactive and reactive solutions. In particular, exploiting the regional knowledge of smart objects allows them to react to outer stimuli, perpetrate their inference rules, and fulfill specific goals.

Al-Sakran [108] proposes an agent-based framework for intelligent traffic monitoring. The system integrates RFID and Wireless Sensor Network technology to monitor several explicit parameters for this domain. The collected information is gathered by a software agent that is integrated into the device, stores the data locally, or forwards the information to the other agents for further processing, which allows making decisions for the monitoring segment. The framework is domain-specific and works in a decentralized environment that addresses IoT's mobility and heterogeneity requirements. The framework is simulation-based, tested for a number of segments, and it is good to see how it performs in real-life scenarios.

Table 2.3: Agent oriented middleware approach comparison

Name	Dynamics	Autonomy	Robust and Avail.	Scalability	Self-org.	Energy	Mem. and Proc.
Agilla	✓	✓	✓	✓	✓	***X***	✓
Impala	✓	***NS***	***NS***	✓	***NS***	✓	***NS***
ActorNet	✓	***NS***	NIP	✓	NIP	✓	✓
UbiRoad	✓	✓	✓	✓	✓	NIP	NIP
MAPS	✓	NIP	NIP	✓	NIP	***NS***	✓
MASPOT	✓	✓	✓	✓	***NS***	***NS***	✓
TinyMAPS	✓	NIP	✓	✓	NIP	✓	✓
UBIWARE	✓	✓	✓	✓	NIP	NIP	NIP
EDBO	✓	✓	✓	✓	✓	NIP	NIP
SAMSON	✓	✓	✓	✓	✓	✓	NIP
DIAT	✓	NIP	NIP	✓	***NS***	***NS***	***NS***
EAGILLA	✓	✓	✓	✓	NIP	NIP	NIP
Sensomax	✓	✓	NIP	✓	***NS***	***NS***	✓
Bisnet	✓	✓	✓	✓	✓	✓	✓
FIoT	✓	✓	NIP	✓	✓	***NS***	***NS***
ACOSO	✓	NIP	NIP	✓	***NS***	***NS***	***NS***
Al-Sakran	✓	***NS***	***NS***	✓	***NS***	***NS***	***NS***
Legend: NIP: No Information Provided, NS: Not supported ✓: Support							

2.5.4 Discussion

The previous sections have provided an overview of sensor networks concepts, applications, and challenges. Following that, a review of middleware architectural paradigms was conducted, and a taxonomy of the approaches was presented. The middleware paradigm advantages and disadvantages are presented in section 2.4.1, and table 2.2 that gives us the knowledge for the most approximate approach that can address the requirements of the future WSN middleware with regards to the comparison criteria presented in section 2.3.

Database approaches provide a familiar and easy implementation of the wireless sensor network (compared to other approaches) with a user interface for issuing queries in a central database and returning the approximate results. They are often restricted to working with specific hardware like TinyDB[109] and unsuitable for dynamic environments.

Virtual machine approaches provide a good model to overcome the heterogeneity of the hardware platforms but implementing a new platform that runs with a virtual machine requires a significant amount of time. Then, the platform is able to change the behavior accordingly to the environmental conditions. On the other hand, installing the VM requires resource-rich capabilities.

Service-oriented middlewares provide modularity of the IoT systems with a high potential to address several issues of IoT. This approach is highly recognized in several studies of IoT middlewares [40]. Often, these middleware approaches are combined with other approaches, such as OpenIoT with GSN middleware [110]. Alternatively, the service-oriented approach cannot provide fine-grained modularization in the constrained node platforms, which in turn restrict performing tasks in a highly dynamic environment that requires knowledge to adapt to the environment changes.

Agent-oriented approaches are seen as a solution to perform well in a highly dynamic environment where the infrastructure is not well-defined, and changes in the network occur very often. By applying knowledge through collaborative interactions locally or globally, agents can be seen as external controllers of the whole network without human intervention.

Given the above, disparate agent-oriented middlewares were discussed in section 2.5.3. A comparative study based on the comparison criteria was summarized in table 2.3. Each of these approaches has its own merits and limitations. This comparative study is done so that a given middleware fulfills such dynamic requirements. This analysis is extended and discussed in more detail in chapter 3, which presents one of the discussion themes of this thesis.

Finally, as can be seen from the research literature, agent-oriented middleware, even though with high potential to deal with the specified criteria (the increasing demand for ultra-scalable middleware solution that is able to work in a highly dynamic environment, changing state of sensors and entities themselves) there are still limitations in resource discovery and applying real-world knowledge which yields with advanced reasoning capabilities and fault detection of the devices, for instance, in a dynamic environment in the city where cars enter and leave the parking slot, how one can get an exact answer that a nearby parking area in the city has a free parking spot. Recent studies apply the semantic context of real-world entities using Ontologies which is the topic of the next sections.

2.6 Data Requirements for Wireless Sensor Networks

Much effort is spent integrating WSN and providing service to end users by addressing specific requirements. Less research is done on interpreting and enabling machine-to-machine understanding of sensed data context. Real-world sensed data are different (temperature of the room, environment, vehicle, human, inside an airplane, tunnel.), and accessing, understanding, processing, and interpreting data from various sources is tremendously important. Considering the vast volume of data generated by IoT devices, particularly by Wireless Sensor Networks, many business opportunities, industrial improvement, and real-life activities would be improved if these data were adequately exploited. In 2012, IoT generated 2.5 quintillion bytes daily, while in 2025, 72% of IoT devices are expected to be connected and generate two zettabytes of data.

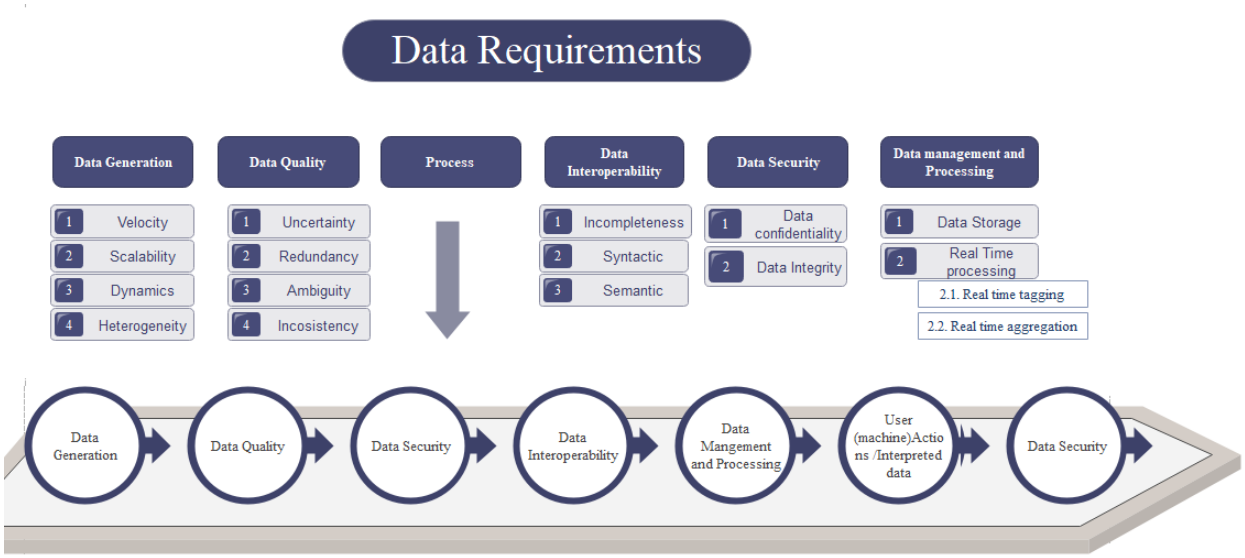


Figure 2.4: Data Requirements and characteristics

The volume of generated data is overgrowing, and building applications using monitoring technologies typically involves additional real-time requirements. Yongrui Qin (see [60]) summarizes data characteristics in four categories: Data generation, quality, and interoperability. Although these are valid for WSN data, there are more practical issues that need to be considered. Since sensor nodes have limited capabilities, data format, real-time processing, and tagging are essential factors for generated data. We list the important data features and characteristics for future WSNs middleware in what follows.

2.6.1 Data Characteristics and Requirements

The number of everyday connected objects is 20.6 billion in 2020, which consists of 2.7 connected devices per person (7.6 billion people in total) [3] with 14.1 ZB[4] generated data per year. As this number continuously grows, the domain of sensors still needs attention for standardization. Standards for physical and electrical feature representation, data processing and message parsing procedures, and the possibility for offering Plug & Play capabilities [111].

Data processing [112, 113], quality of data [113], and storage [112, 113] becomes a big challenge with the enormous increase of connected sensory devices attached to daily objects. The data will play a significant role in every field of interest when they are exploited correctly. Machines need the collected data provided by various sensors to be understandable. The need for structured data enabling machine interpretability and reasoning is essential [114]. It requires attaching further semantics to the actual raw data.

a) **Data generation**:

1. Velocity - depending on the monitoring environment, data can be generated at a different speed. Some applications require every second to transmit the sensed value, which will reduce the energy consumption of the nodes, or the data can be generated slowly enough that could result in possible loss of information.

2. Scalability - generating continually raw data from hundreds or thousands of nodes drastically increases the data volume. Considering this, the system's performance and governance of generated data will be an issue.

3. Dynamics - this means that due to the mobility and environment dynamics, the need for location and time to attach to the sensed information is required.

4. Heterogeneity - having a diversity of vendors and applications of different kinds creates the demand to deal with different formats and structures of information from specific domains.

5. Plug & Play - providing the sensor node the opportunity to attach sensors into it without further configuration is greatly appreciated.

b) Data quality:

1. Uncertainty & Accuracy - due to the noise and influence of the environment, sensors could produce values that are not correct or accurate.

2. Redundancy - monitoring an environment could make it possible to deploy similar sensor nodes in nearby places that will collect similar values. Also, sensors would produce the same values for a period of time due to unchanged values, which again creates Redundancy.

[3] https://mitechnews.com/internet-of-things/how-big-is-iot-20-6-billion-connected-devices-by-2020/

[4] https://www.forbes.com/sites/joemckendrick/2016/11/13/with-internet-of-things-and-big-data-92-of-everything-we-do-will-be-in-the-cloud/#231f5d154ed5

3. Ambiguity - data could be interpreted differently from two different things.

4. Inconsistency - when a phenomenon is monitored by two or more sensors, the collected information would result in Inconsistency. Inconsistency is due to Accuracy, packet loss, noise, or other factors involved in the sensing process [113].

c) Data interoperability:

1. Incompleteness - WSN networks are mobile and dynamic that cooperate with each other to achieve a common goal. In such a cooperative environment, it is vital to see/distinguish the missing information that a specific node has produced.

2. Syntactic - machines need to communicate to each other the sensed values for taking actions or fulfilling the provided goal. Devices must share a common data format for communicating with each other.

3. Semantics - apart from communicating the information, the possibility of understanding and interpreting the common meaning of such information is crucial for future dynamic applications.

d) Data Management and Processing

1. Data Storage - storing continuously generated data for use in later phases is an essential aspect for historical reasonings or building systems smart enough so that when a similar situation appears during phenomena monitoring, it could predict and prevent not happening the same thing.

2. Real-time Data Processing - a huge amount of real-time data streams will be produced by sensing devices, and having data processing mechanisms to get insights is required in order to utilize applications to their best. It involves:

   (a) Real time tagging - the data coming from sensors are unstructured, and the possibility of identifying the data origin and noisy data requires extra information to be attached.

   (b) Real time aggregation - processing real-time information on a scheduled time and finding important patterns is desirable for future dynamic systems.

2.6.2 Data Requirements for the research study

For the purpose of this study, we will take into consideration the following characteristics with regards to data:

- Syntactic and Semantic Interoperability - current and future middlewares requires interoperable systems where the data can be easily processed and understood by machines.

- Accuracy - the data comming from sensor nodes needs to be accurate.

- Velocity for energy efficiency. How to improve energy efficiency by reducing data transmission in between the nodes. It is important that sensor nodes send as fewer data as possible and not frequently.

2.7 Syntactic & Semantic Interoperability: Standards and Trends

Wireless sensor nodes are composed of a number of sensors, a transceiver, a radio, a power source, and memory. They are small devices with limitations in memory, communication, processing, and energy consumption [115, 116]. Further, sensor nodes are sensitive and prone to physical damages that will produce unrelated values or often fail when they are deployed in harsh environments. Taking into account that sensory devices are used to measure the physical properties of the objects being observed, several research challenges need to be addressed. An emerging research field of these challenges are associated with data: lightweight format to be used, quality of data, syntactic and semantic interoperability, redundancy, and scalability. Indeed, these challenges become more difficult to deal with when there are plenty of manufacturers who produce these kinds of devices, and each of them has its own way of describing things.

Accordingly, there is a need for data standards that are able to describe physical and electrical features and measurements, enabling plug & play capabilities [111], providing syntactic and semantic interoperability, and offering data scalability by always taking into consideration resource limitations. Providing such a standard allows us to easily filter unnecessary data or detect erroneous ones, which yields to better decision-making. It is also important for energy saving by not transferring inaccurate data to other nodes. In the case of the minimum energy or outside noise directly influencing sensors, wrong data will be generated that could easily be detected.

As long as the sensor node energy lifetime is critical in a WSN, it demands little packet size, and less frequency of packet delivery is necessary. The less data are transmitted, the more energy efficient the node will be. Sending 100 bits of data from node to node consumes 5 μJ [117]. Since the data needs to be transmitted from node to node until it reaches the sink node, the energy will be wasted in the whole network.

2.7.1 Sensor Data Standards

Having open data standardizations for describing sensors and their capabilities in the domain of WSN provides a mean for interoperable systems, stimulate market competition, prevents parties from controlling a standard, and avoid the need to be stucked in a specific architecture for solving problems that will result in innovation and differentiation with the provision of better services. A goal of this section is to present sensor data standards, discuss them according to the specific layer categorization and then represent what they have done for data interoperability.

SensorML [118] provides an XML schema for describing sensor meta-data, and its processes, while O & M is a standard that describes the observations. Both present standards under the umbrella of SWE group. A combination of SensorML and O & M [119] gives good modeling to describe the sensory information and thus create syntactic interoperability.

The IEEE 1451 is a set of standards able to interconnect smart transducers with the systems. It provides numerous functionalities [120] (like self-identification, self-description, self-diagnosis, location-awareness, time-awareness, data processing, etc.), and thus (semi)automation could be achieved. Here, the transducer electronic data sheet (TEDS) is introduced as a specification for sensory information, able to be attached to memory in the EEPROM of smart transducers.

Amon [121] is a general-purpose open data format for describing and exchanging sensor/metering/monitoring devices data. It describes the data capabilities of devices, metering points, and entities.

SenML is a lightweight media type for representing sensor measurements and related data, with the focus devices with limited capabilities [122]. It is designed to carry multiple measurements and the possibility that the packet size could remain under 80 bytes. The supported protocols are Constraint Application Protocol (CoAP) and HTTP.

Echonet [123] is an open standard supporting heterogeneity of home appliances and the sensors integrated into such types of equipment. Many vendors are already using Echonet in the Japanese market in the domain of smart homes.

Device Kit is an OSGi technology that serves as a data model for devices. It also simplifies the development process for future applications when hardware characteristics are unknown [124].

Device Description Language (DDL) [111] provides an XML schema for describing sensors and related devices by exposing them as services. It is a reference for Service Oriented Device Architecture (SODA).

2.7.2 Scope of the Standards

In a world full of different types of sensors, ensuring the heterogeneity of the devices and making them fully functional is not quite easy. The applicability and the scope of each previously mentioned data format in the prior section are different. Table 2.8 presents functionalities they provide in different layers starting from the physical to the application layer. Indeed, these functionalities are represented for some of the standards in [111], and we are extending them with additional data formats.

Operating environment covers the description of the physical characteristics of the devices (form factor) and operating environment. This information, even though not highly important for the developer can serve as a form to check whether the device works properly under specific conditions.

Physical layer describes the electrical characteristics, mechanical interfaces, and other form factors of physical devices (size, shape, etc.).

Units and block layer (Pins and ports) describes pin wiring, pinout, and timing of signals. It will allow us to describe the pins and ports of the sensors with the platform.

These three layers are less important for the developer and, consequently, for the application point of view since there is not much that could be done using such information.

Events and Protocol Layer is the one that describes the parsing procedures of the signal being processed. These procedures need an analog-to-digital converter (ADC) for the analog voltage port or a string for the digital port. Most of the standards that need to tackle the issue of the Plug & Play capability need to be able to provide specifications of the receiving signal. Whenever a sensor is connected to the platform, it generates a timed pulse that is translated into a unique identifier. This identifier maps to an address space that will cause the platform to download the driver for the specific sensor and automatically start receiving measurements for the monitoring object. To enable the Plug & Play capabilities, only TEDS, DDL, and Device Kit are the standards fulfilling such requirements.

Functional layer provides descriptions for the semantics of the receiving signal. For instance, whenever a signal is sent by a sensor, this is converted by ADC to the appropriate value (i.e., 30 degrees Celius). These semantics are the most important features for developers and, thus, a target of all the standards described above. In this layer, the measurements and units are described as time, location, and other important features of the monitoring objects.

Network layer describes interfaces for wireless communication and protocol stack. Since the majority of the sensors do not have wireless communication capabilities, most of the standards do not include descriptions for this layer.

Data service layer exposes the device interfaces to the outside world as a service over the network. This layer is supposed to manage the communication protocol and data format together with the semantics where it is possible to access the device's data over the network. This layer has not been part of the abovementioned standards.

Table 2.4: Sensor descriptions of standards in different layers (Y=Yes, N=No)

Description	IEEE 1451	DDL	Device Kit	Sensor ML	Echonet	Amon	SenML
Operating environment layer	N	Y	N	N	N	N	N
Physical layer	N	Y	N	N	N	N	N
Units and block layer	N	Y	N	N	N	N	N
Event and protocol layer	Y	Y	Y	N	N	N	N
Functional layer	Y	Y	Y	Y	Y	Y	Y
Network configuration and protocol layer	Y	N	Y	N	N	N	N
Interoperable data service layer	N	N	N	N	N	N	N

We see in Table 2.4 that the only standard able to describe how environmental variables affect the results of the device is DDL. In IEEE 1451 and Device Kit, the lower layer is the Event and protocol layer. We also know that the data carried by sensing devices are just raw readings, and if no parsing procedure exist, there is no practicable way to understand such readings.

Although, under the assumption that parsing procedures are implemented by vendors for each sensor node, almost all the standards choose the lower bound layer, the functional one,

which is also considered mandatory. In this layer, the semantics of the receiving signal is described (i.e., 30 degrees Celsius). Some standards like IEE 1451 and device kit define networking capabilities, while the last layer is not in the scope of any standard. The networking layer is the discussed topic among researchers whether it should be included in the standards [111] since most of the sensors do not have networking capabilities. Nodes themselves are responsible for data transmission.

The majority of manufacturers describe the capabilities of the device (error rate, max., and min. values, device operation condition mode, parsing procedures, sensor features, etc.) using their specific model, and often they do not follow a particular standard already mentioned in this report. Hence, the scope of the majority of the presented standards is to model the observed sensor data and their values (i.e., room temperature, its value, timestamp, etc.). Basically, the chosen standard should allow the modeling of the data so that the packet length remains small and provide simple semantics to enable later the increase of the knowledge with an alternative format with expressive power.

2.7.3 From Plaintext to Data Interoperaility

The increasing number of sensors continually produces a vast amount of raw data. More than values are needed to properly exploit the resources and extract new knowledge from the raw data. Machines require formats that easily could be processed, understood, and even interpreted. The desired format needs to scale well and carry enough information but always considers the constraints of the sensor nodes. For instance, in a home automation scenario, the home temperature is required to maintain cooling and air conditioning automatically. Further, to create intelligent homes, several parameters could be gathered, such as humidity, moisture, and temperature, and based on these observed parameters, it is possible to reason and get more insights that could adapt to user preferences. Nevertheless, all this information requires to have small packet size that could easily travel from node to node until it reaches the sink node. Except for the need to measure real-time values and communicate these data to other machines, the data need to be machine processable and interpretable. It is insufficient to have just temperature value, but there is a need to get more insights from that value (whether the value is in Celsius or Fahrenheit degree, whether it is room, outside, or body temperature).

To enable machine processing and provoke syntactic data interoperability, encoding formats such as XML and JSON are well-formed with a standardized structure. They allow us to model observed raw data in a structured way so that machines can easily process such information. The XML is the common encoding basis for SenML, SensorML, Device Kit, and Echonet. In fact, Echonet device specification is a logical model of the information or essentially a dictionary of devices where each device represented in XML is transformed into an Echonet object. Different encoding formats are supported by SenML (XML, JSON, EXI, and CBORCBOR[5] or textual modeling. formats) depending on device constraints and the developer preferences. DDL uses plain text, which is not the appropriate form for enabling machine processing and enhancing interoperability among machines.

[5]Concise Binary Object Representation: http://cbor.io/

Table 2.5: Features os Sensor Description Standards

Standard Name	**Data Format**	**Semantic Support**	**Units**	**Data types**	**Design perspective**	**Constraint devices**	**Supported devices**	**Good for**
SensorML	XML	Yes (if integrated with O & M)	Yes	CDT	Data Oriented	No	Intelligent Sensors	Describing complex devices
IEEE 1451 (TEDS)	IDL	No	Yes	CDT	Modular	Yes	Intelligent Sensors	Manufacturing the design devices interface
Amon	JSON	No	Yes	PDT	Data Oriented	No	Devices	Industry (metering/ monitoring)
SenML	XML, EXI, JSON, CBOR	Very limited	Yes	PDT	Data Oriented	Yes	Sensors, Actuators	Very constraint devices
Echonet	XML (Object - dictionary of devices)	Limited	Yes	PDT	Class	Yes	Home Appliances	Home automation
Device Kit	DKML	No	Yes	PDT	Modular	No	Sensors, Actuators	Exposing devices as services (Eclipse plugin)
DDL	Plain text	N/A	Yes	PDT	Data Oriented	Yes	Sensor and Devices Communication	Manufacturing and process industries (Plug & Play)
*PDT = Primitive Data Types *CDT = Complex Data Types								

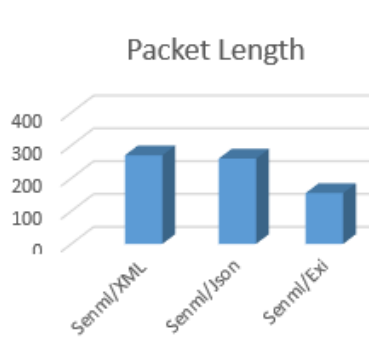
The possibility of sending several measurements in a single packet without exceeding bandwidth is important. Several sensors can be equipped in a single sensory device, which will need all the observed properties to be delivered to the end node. It is essential to choose a lightweight format that could easily be expanded with semantics and carry all the measurements simultaneously within its boundaries.

Regarding syntactic Interoperability, the efforts have been focused on employing SensorML from the OWC group. Recently, many middlewares have used SenML as a lightweight format for their Wireless Sensor Network architectures. The former consist of a set of model languages that define sensors and their services by exposing them to the web. However, additional effort is required to automate the process; sometimes, it takes effort to implement the proposed technologies.

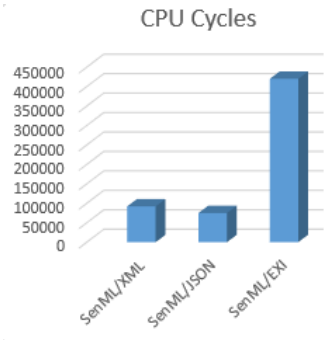
Regarding scalability, implementing XML as a data format for real-time application development promises extensibility and flexibility. At the same time, managing information as XML raises three issues related to enterprise scalability and robustness. The first is that XML is not designed for fast information retrieval. The second is that XML is a verbose method for exchanging data over a highly constrained network, which becomes a problem when exchanging a large amount of data daily. The third issue is that XML elements are not defined as native data types. The steps required for converting XML-tagged data into predefined data types degrade the performance of a real-time WSN application system. On the other hand, JSON is a lightweight format with simple syntax, easy to process, and easily

can be extended with other data.

Dealing with node constraints and energy efficiency, EXI and CBOR are the preferred formats for the transmission of the data in binary form due to the small packet size. EXI format allows the delivery of packets easily from node to node, can process the data locally, and making decisions requires a lot of CPU resources which is not convenient for the CPU-constrained platform. It creates difficulties in detecting erroneous data from quality data, and it will break the hard constraints of real-time monitoring. On the other hand, it is possible to model the data in JSON format that will include a small amount of data, including basic semantics, so that later these data could easily be transformed into more sophisticated knowledge that will allow machines to interpret. XML is another alternative, as most standards have proposed. However, XML is considered heavyweight compared to JSON [125, 126]. Xiang et al. ([22]) evaluate the formats through experiments with the formats XML, JSON, EXI, RDF, and Entity Notation. Since the performances are getting better for these sensory platforms, the more semantics we will provide to the next node, the better exploitation, and usage of these data will be possible.



(a) Packet length extracted from [22]



(b) Number of Cycles [22]

In Figure 2.5a, the Exi format has the lowest packet length, followed by JSON and XML. In contrast, the number of cycles to process the same information (Figure 2.5b) is lower in JSON format compared to XML [22] and EXI. The last data format requires high processing power.

In conclusion, future application requires quite a standardization that could easily integrate the sensing devices into the system and make use of it. Echonet has attracted the Japanese market for home automation. Enabling Plug & Play properties for sensors, the IEEE 1451 standard is the most appropriate that offers such a possibility. This standard allows specifying configuration information in the memory of the sensor. Specifically, using TEDS, it is possible to identify the current sensor attached to the devices, its interface, and other relevant information the sensor detects. Later, all this knowledge can automatically be detected and received when the software interface is written. SenML utilizes constraint resources with minimal energy harvesting, low processing power, and when the packet length matters. To utilize the resources with limited bandwidth, it is SenML that can model objects as an appropriate model while this object contains an array of entries. Further, when the bandwidth is highly constrained, this protocol allows a model of the data using EXI format and sends minimal data over a mesh network. It also helps minimize energy consumption

when we know that most energy is spent during transmission. However, EXI requires more processing power to process the carrying information. Considering the popularity of JSON format, it is possible to model the data with 80-byte important observations. Amon is created more with the purpose of being human-readable, while Device Kit and DDL, even though young in the market, have the option to enter the industry of Wireless Sensor Networks. SensorML, on the other side, is well-known in industry [111], but its complexity and design nature limit the applicability in small, constraint devices [127].

2.7.4 Toward Semantic Interoperability for M2M Communication

The already mentioned standards are not widely accepted and thus are not enough to provoke semantic data interoperability because sensors and platforms are strongly dependent on the sensor manufacturer and industry. Data annotation and semantics are valuable metadata for today's real-time systems to overcome heterogeneity, M2M understanding and interpretation, and discovery of the resources and extraction of new knowledge from the output of multiple sensor measurements. While sensors consist of common characteristics across applications, metadata are those characteristics that make a difference and achieve common understanding for both: humans and machines.

Utilizing semantic presentation and the common standard ontologies would help to achieve interoperability at higher levels and understand the proper meaning of the data. For instance, machines could distinguish room and body temperature and apply different reasoning techniques. Therefore, through the literature review, sensor ontologies should include several concepts and relations between concepts. The general required attributes for sensors as proposed by the National Institute of Standards and Technology (NIST) [128] are summarized in Table 2.6. We are categorizing these concepts as follows: a) Sensors / Actuators: identity, manufacturer, configuration b) Physical features: location, power supply, energy, actual consumption energy, operating conditions, force, speed c) Observations: observation, accuracy, frequency, and d) Data: data types, latency, units of measurements, quality of data, time.

A shared ontology (or multiple shared ontologies) describing the above concepts and relations provokes semantic interoperability among the applications. This shared ontology allows exchanging information so that the meaning of the observed properties of the objects will be automatically processed and interpreted by the receiver. Through reasoning, new semantic facts could be achieved based on the observed data.

2.7.4.1 Technologies and Languages for Semantic Data Interoperability for Wireless Sensor Networks.

Semantic [6] is the study of understanding and interpreting the true meaning of a word or a sentence. Such words, to be understandable by computers, need a semantic representation using symbols or markup languages.

[6]Semantic, according to: http://www.thefreedictionary.com/semantic

Table 2.6: Sensor Required Attributes as proposed by NIST

Attributed	Comments
Physical	Power, weight, size
Operating environment	Conditions for device operation
Immediate data	Time, spatial data, latency, frame rate and other important data
Derived data	Results after computing raw data
Algorithms	Algorithms for producing outcomes (derived data)
Integration/fusion	Data aggregation and fusion combining data from multiple sensors
Capabilities	Functional applications from the data
Communication	Physical and logical protocols, interoperability
Processing	On-board processing power of sensors and sensor nodes
Calibration	Calibration algorithms
Provenance	Maintaining the records of raw data and derived data
Confidence	Level of confidence in raw data

We already know that the most intriguing and successful artifact is the current Web [129], but still, explicit resource exploitation needs semantic technologies. Hence, the semantic web allows humans and agents to cooperate with each other, reason over the available resources, and accomplish sophisticated tasks [130]. The core concept of this semantic representation is Ontology, which enables knowledge discovery of a shared resource. It can be used for knowledge representation [131], natural language processing and knowledge engineering [132], semantic representation, and building expert systems [131].

Most of the research attention is given to the true meaning of the receiving values from sensors. Several studies address the significance of annotating sensed data and the powerful utility that could be derived [133, 134, 135]. When the (semi)automatic annotation of sensed data is achieved, it will have a significant impact on the real-time monitoring system and enable real-time machine processing.

To make this happen, several ontological languages are created with the purpose of facilitating the annotation process, which will help to achieve semantic interoperability and make the data accessible on the web, as well as expose these as services to third parties. A set of core semantic technologies are already developed, and most of them are de facto standards [136]. Languages like Ontology Inference Layer (OIL)[129], DAML+OIL [137], RDF [138], Web Ontology Language (OWL)[139], JSON-LD [7], enable to describe sensor information with semantics attached to it. Standard ontologies are important for achieving semantic interoperability using the core semantic languages and technologies.

- **Ontology Inference Layer (OIL)** is presented as ontological infrastructure for the Semantic Web built upon Description Logic (to reason over services) and frame-based languages (semantic modeling) [129].
- **DAML+OIL** is a successor of OIL and combines features of DARPA agent markup

[7] http://json-ld.org/

language. It adds features to previous W3C standards (RDF, RDFS) by extending with ontological primitives of object-oriented and frame-based systems [137].

- **RDF** the standard for representing the data in the form of triples on the Web is RDF. Each triple in RDF carries a subject node, predicate, and object node [138]. To extend the structure of the Web, it uses URI to name the relationship between things. The supported data format is an XML serialization form for exchanging the data between applications [136]. RDF has limitations in providing the relationship between concepts. Thus, RDF Schema and OWL are proposed.

- **RDF Schema (RDFS)** extends RDF and adds features in describing the data properties [139]. In RDFS, we have the notion of classes and properties, similar to OOP languages like Java, with the difference in setting the properties as classes [139].

- **Web Ontology Language (OWL).** Representing the knowledge on the web is made through OWL [139]. The base of OWL is the DAML-OIL language which adds accessibility features to machines in the document's content and additionally increases the reasoning power [140].

These are the de-facto standards for integrating IoT into the Web of things.

2.7.4.2 Ontologies for Sensor Data.

The community has developed several ontologies for describing important information about sensor data. These include measurements, observations, features of interest, location, time, etc. In what follows, we present Ontologies (even though none of them is a standard) that could possibly provide interoperability at a global level of the system and become de facto standards.

Semantic Sensor Network (SSN) Ontology [141] describes sensors, observations, and related concepts. These concepts in SSN ontology are divided into four different perspectives: a) Sensors can semantically describe device capabilities of the sensed object, b) Observations represent the sensing values, c) Systems consisting of subsystems, and d) Feature and Property for representing phenomena properties.

An important aspect of this Ontology is the possibility to add external domain-specific ontologies and extend its capabilities (i.e. Time, Location, or domain-specific ontologies: i.e., Health ontologies). Further, we can have a better level of interoperability in SSN ontology since it aligns SSN ontology concepts with the DOLCE Ultra Lite (DUL [8]) upper ontology.

Sensor Web for Autonomous Mission Operations (SWAMO) [142] is a framework and ontology able to create interoperable sensor web products and services within the sensor web. It models components, systems, and processes as physical or logical representations. It consists of agents able to maintain operations as well as make decisions. It is compatible with SWE.

OntoSensor [143] presents another ontology for representing sensors, capabilities, and measurements. This ontology presents a broad knowledge base of sensors (humidity, GPS,

[8]DUL - is an upper ontology for presenting concepts understandable among all knowledge domains

Accelerometer, etc.) for query and inference. To express sensor properties, SensorML is used to derive its terms. The Ontology seems not updated and complex to implement [9].

Universal Plug and Play (UPnP) [144]. Presents an Ontology to enable Plug& Play capacities by describing devices and services they provide. The UPnP ontology presents instances of devices, services, actions, arguments, and state variables. The Ontology is not updated, and no new features are added (not actively maintained).

Foundation for Intelligent Physical Agents (FIPA) [145]. FIPA ontology describes devices and their characteristics aiming to exchange and create data interoperability between software agents. The main class of this ontology is Device which has descriptions and allows the description of hardware and software properties. Since its introduction, this ontology has had no updates or added features.

OpenIoT Ontology [146]. presents an extension of the SSN ontology with additional concepts relevant to cloud integration, utilizing unit metrics and points of interest (POI) at some level of granularity. It enables to collect the data virtually. It also offers tools for the development and deployment of IoT applications. There are extensions regarding client authentication, Client credentials, and other services-related concepts.

The most relevant ontologies offering semantics for describing features of sensing devices are reviewed. One of the ontologies from the SWE group was O & M. To distinguish with SSN ontology, the O & M describes the observations as events, while in SSN ontology, the observations are described as a process. SSN differentiates the domain-specific concepts and those devices-specific, while in O & M, everything is presented as a whole. The SWAMO ontology is very complex and hard to develop and integrate. This ontology helps expose components/processes as web products. The OntoSensor is not actively maintained and is hard to implement. OpenIoT ontology presents an extension of SSN attempting to capture important information about the sensor, its units, and domain-specific information. However, additional work needs to be addressed by OpenIoT in order to address sensor capabilities, performance, and usage conditions. The most relevant ontology to describe the sensed information semantically and the object being observed seems to be SSN ontology. This ontology, combined with other domain-specific ontologies, allows for semantic interoperability and prevents the misconception of the data coming from sensors. It means machines are able to process information in a meaningful manner. Another important feature of SSN is to use solely parts of it (i.e., measurements). It is very beneficial to transmit the correct information to other parties for further processing using low-cost technologies. However, transmitting the right information requires some higher intelligence in the nodes, and this cannot be achieved just by semantically annotating the sensed data. Using SSN ontology, we can connect other ontologies like location (indoor or outdoor location) ontology, which is among the research challenges. All this information could enhance the discovery process of the resources using exact locations and respond to user queries with higher precision. In other words, SSNs enhance the data and device discovery process by semantically describing available resources and observations and offers a great possibility to achieve semantic interoperability.

[9] `http://www.w3.org/2005/Incubator/ssn/wiki/Incubator_Report#OntoSensor`

Table 2.7: Sensor Ontology Features

Name		SSN	OntoSensor	SWAMO	UPnP	FIPA	CSiro	OpenIoT
Sensors	Physical characteristics	Y	Y	N	N	Y	Y	N
	Identity/ manufacturer	N	N	N	Y	Y	Y	N
	Configuration	Y	Y	N	Y	N	Y	Y
	Hierarchy	Y	Y	N	N	N	Y	Y
	Deployment	Y	N	N	Y	N	Y	Y
Physical features	Location	N	Y	Y	N	N	Y	Y
	Power supply	N	Y	N	N	N	Y	N
	Energy	N	N	N	N	N	N	Y
	Current consumption	N	N	N	N	N	N	N
	Operating conditions	Y	N	N	Y	N	Y	Y
Observations	Input/Output	Y	Y	N	N		N	Y
	Observation	Y	Y	Y	Y	N	N	Y
	Accuracy	Y	Y	N	N	N	Y	Y
	Frequency	Y	Y	N	N	N	Y	Y
	Response model	Y	Y	N	N	N	Y	Y
	Communication process	Y	N	N	N	N	Y	Y
Data	Latency	Y	N	N	N	N	Y	Y
	Units of measurements	N	Y	Y	N	N	Y	Y
	Data types	N	Y	Y	N	N	Y	Y
	Data quality	N	Y	Y	N	N	Y	Y
	Time	N	Y	Y	N	N	N	Y
Y = Supported, N = Not Supported								

2.7.5 Discussion

We have divided sensor standards into two categories: a) sensor data formats dedicated to simple sensor descriptions and b) standard ontologies for utilizing semantics when describing sensor data, capable of overcoming issues already mentioned in section 2.3 (syntactic semantic interoperability).

Certainly, in the first category, fall sensor standards that assist manufacturers with the standardized structured format to describe features of sensory devices and their capabilities. Due to restrictions in memory, bandwidth, and CPU, describing in detail each sensory device and enabling automatic discovery of the resources is not considered by several standards. Most of the time, manufacturers follow their standardized approach to describe information and present standards that describe observations and measurements.

In this category, the majority of the standards do not provide detailed complex features such as quality of the information, allowing to discover anomalies in the data that will later allow performing the cleansing procedure. With the increasing volume of data, data

Table 2.8: Number of Concepts, Axioms and Properties

Name	SSN	OntoSensor	SWAMO	UPnP	FIPA	CSiro	OpenIoT
Classes	52	289	No info	9	15	71	74
Axioms	599	1970	No info	310	203	503	383
Object Property	55	125	No info	11	14	61	27
Data Property	0	108	No info	23	18	11	15

inconsistency becomes an issue. Highly dynamic environments require temporal consistency, which needs small packet size and efficiency in transmission. It is known that most energy is spent during transmission, and sending large packet sizes requires a lot of energy. Utilizing XML as a data format creates a larger packet size, requires more processing power, and is less scalable compared to JSON. EXI format is a better alternative when constrained devices are present, while JSON seems a better alternative compared to XML. Future attempt needs to be addressed by each standard, to reduce packet size, add more semantics, and improve the accuracy of information. From the point of view of resource discovery, employing simple standards provides difficulties in automatic discovery, and answering sophisticated questions. It is difficult to enhance query answering using simple formats with no added semantics. Also, spatial information are important features for the domain of WSN since the sensor standards are limited to only latitude and longitude which most often not enough.

On the other hand, the second category tries to overcome issues of semantics, and heterogeneity, sophisticate query answering, enabling automatic discovery and inference of new knowledge from the output of several related measurements. Even though no standard is presented here to describe all the features and capabilities of sensory devices, some are becoming de facto standards. One widely adopted standard is the proposition of the W3C group with SSN, which follows upper ontology guidelines for modeling concepts, which will result in general semantic interoperability within systems. Other ontologies seem not updated, neither maintained.

Due to the constraints of sensory devices, these ontologies could help to annotate sensed data in more powerful devices. First, the data travels using a simple, non-semantic format with a small packet size, and later these data are semantically enhanced using techniques that enable automatic conversion. However, automatic semantic enrichment of sensor data that could be effectively connected with event processing to enhance the expressiveness of event processing demands a lot of effort [113]. What can be seen in the second category is the possibility of combining several ontologies, relating common concepts, and adding the features that are missing. Most of the Ontologies leave out the performance concepts, energy level, and capabilities which are exposed at some level of granularity. The SSN ontology plays a key role in what we saw from the reviewed Ontologies. It has inspired other projects, the most relevant one being the OpenIoT project. We can take this as a base ontology, use the features already present, and make use of concepts from domain-specific to enrich with semantics already missing in the base ontology.

An automatic semantic annotation process utilizing Ontologies could overcome hetero-

geneity and interoperability by possibly modeling all the concepts and attributes of WSN. Even though none of the presented Ontologies overcome heterogeneity/interoperability issues since they cover only parts of the required information model, there is a good indicator to be extended with additional features and properties to address such requirements thoroughly.

Inferencing new knowledge and transforming the raw data into sophisticated data could be achieved by utilizing a semantic representation of the observed raw data from sensors. New knowledge will be extrapolated by observing several related parameters and applying rules over the real-time values. For instance, in a driving car scenario, there is possible to predict, take the right directions and be in a specific location for the least time possible. Gathering values of car speed (velocity), location, and directions and applying specific rules could provide the best directions. Similarly, by measuring the distance between cars, rain level, or if the weather is foggy, it would recommend the best speed to drive the car and avoid possible accidents.

A significant role in processing and reasoning over the semantic data in WSNs applications is the modeling language. It requires to be lightweight, scalable enough due to a large amount of generated data, and easily processable considering the hard real-time constraints of Sensor Networks. Maarala and Su [147] evaluate the performance of different semantic ontological languages in processing in different scenarios (centralized systems, distributed, mobile). Short Entity Format (EN) outperforms in the experiments, having the lowest latency and minimal resource usage.

2.8 Reducing Data Transmission via Dual predictions

Many works exist today intending to increase the lifetime of WSNs. One category dedicates to routing strategies, while the other dedicates the research work to reduce data transmission. Our focus stands on the second category of improving the energy lifetime of WSNs by reducing data transmission. In literature, various techniques aim to reduce data transmission, starting with data aggregation techniques presented in [148], data compression[149], adaptive sampling [150], event-triggered and procedures based on dual prediction [151, 152]. Data compression algorithms aim to reduce data packets' size and transmit in decreased data size [153]. Data aggregation sends summary statistics in the gateway, mainly collected by similar sensor nodes. On the other hand, event-based triggers only when certain events occur. Adaptive sampling (AS) modifies the sampling rate when a critical event is detected. Another popular technique that reduces data transmission is via prediction techniques in the sensor node and the gateway. The dual prediction (DP) approach aims to reduce the number of transmissions using prediction models, while adaptive sampling reduces the number of collected data. Between the DP and AS, dual prediction is more suitable as it provides a better data suppression ratio [154] and thus helps reduce erroneous data and has fewer data at the gateway.

An approach proposed in [154] uses dual prediction and adaptive sampling to increase the energy efficiency of the WSN network. The aim is to reduce data transmission when there is a low correlation between variables, while the sink can generate the values based on previously collected samples. The algorithm reduces energy up to 57%. In [155], authors use

the line equations via n-dimensional vectors to predict both nodes for DP. Another technique, "error-aware data clustering (EDC)" for in-networking prediction, is the work proposed by the authors in [156]. The EDC model consists of three modules where users can choose a suitable module for their specific case. The main aim is the use of different techniques like histogram-based clustering (HDC), Recursive Outlier-Detection Smoothing (RODS), and V-RODS. The EDC approach decreases data redundancy and detects possible erroneous data in WSNs. Another technique aiming to increase energy in WSNs is the work proposed in [157]. The model combines a finite impulse response (FIR) filter and recursive least squares (RLS) filter. The applicability of least-means-square (LMS) for energy efficiency is the research work carried out in [158]. The nodes are supposed to transmit immediate value when the actual readings exceed a given threshold. Further, authors in [158] use Kelman filtering in the sensor node to reduce data transmission and remove redundant data. On the sink node, two algorithms (Sink Level Aggregation and Sink Level Grouping Algorithm) are used mainly for reducing spatial data redundancy. Even though the algorithm performs well in reducing spatial and temporal redundant data, the energy efficiency is increased by up to 50%.

All the abovementioned approaches address the ability to detect erroneous data while increasing the network lifetime based on the dual prediction mode. Although they reduce data transmission at a solid state, the data reliability is often compromised. Furthermore, the algorithms running on devices are complex, and devices with limited resources would be hard to deal with such algorithms. In this research, we will investigate the discrete Fourier approaches with the aim to build lightweight algorithms that can reduce data transmission, and simultaneously enhance data reliability at a satisfactory level.

2.8.1 Discrete Fourier Transform

Frequency analysis is an important method for interpreting sensed data in WSNs. Discrete Fourier Transform (DFT) is identified with frequency analysis, applicable in many science fields, where the most common one is in the field of digital signal processing [159]. Its real-world applicability is often associated with the development of fast algorithms in processing real values [160]. The algorithms are mainly developed using DFT [160] and DCT [161, 162]. The optimized approach to compute DFT is called Fast Fourier Transform (FFT) [160]. Using FFT, it is possible to process and interpret observed values and react to requests in real time.

Using the Fourier series, Zong-Chang [163] established an air temperature model to explain the temperature fluctuation throughout the year in terms of monthly and daily air temperature measured on an hourly basis. The author developed a temperature prediction model based on a carefully chosen Fourier coefficient in the sense of the least square error rule and validated its efficiency to Beichen and other districts of China. Yang [164] proposed a principal component analysis (widely known as PCA) based eigen-temperature model, where the keyword eigen-temperatures are the eigenvectors aimed to describe the temperature movement. The author tested the forecasting capability of the eigen-temperature model at Tongchuan, China, and claimed that the model exceeded the classical BP-ANN model for

forecasting. Solar energy is another form of clean energy, and it has numerous creative applications [165]. Seeing the applications of solar insolation in modeling phenomenon, Yang [166] developed a finite Fourier series-based prediction model incorporated with the least-square method.

To cater to the economic proceedings and other important factors of industries, Yang [167] proposed a discrete Fourier transform-based electric load forecasting model by considering the load movement as a time series data. The proposed model was tested at State Grid Corporation of China, and the model shows its potential over the other known classical models. Similarly, a discrete cosine transforms electric load prediction model [162] is also developed in the sense of the least-square technique. Furthermore, the author in his seminal work [168] investigated the electric loads by means of finite Fourier series in conjunction with the least-square method. Some recent efforts in the modeling and prediction of the water-level fluctuations by discrete cosine transform and finite Fourier series jointly with the least-square method are also discussed in the details, we recommend the serious reader to [161, 169].

2.8.2 Discrete Sine Transform (DST)

DST plays a significant role in many areas, particularly in the image and signal processing problems, initially introduced by Jain [170]. Later, various DST versions were developed by researchers such as Kekreet al. [171], Jain [170]. The evolving algorithms based on DSTs found a wide application in different fields in digital signal processing (DSP) (image processing, adaptive digital filtering) and interpolation [172]. The most important transform is the discrete Fourier transform (DFT) from the family of discrete transforms. It is mainly due to its effectiveness in many applications in diverse fields of science and technology. It also plays a crucial role in different DSP and image-processing apps. Moreover, DFT and inverse DFT (IDFT) have been viewed as SP's main technology in orthogonal frequency division multiplexing (OFDM) communication systems.

Similar to DST, the Discrete Cosine Transform (DCT) is utilized as a form to process and compress data in WSNs. Authors in [173] provide a DCT model that involves the coefficient transmission and signal recovery. Similar predictive models for sensor data based on DCT are research work of [162, 174].

2.8.3 Discrete Hartley Transoform (DHT)

Discrete Harley Transform is a great alternative solution to Discrete Fourier Transform (DFT) [175]. It is faster when processing real values[176]. DHT is also a useful alternative to compute other discrete transforms such as DCT [177] and DST[178]. Besides being an alternative tool for discrete transform, the DHT has a fairly wide use in various fields such as image compression [179], image watermark [180], fingerprint matching [181], image processing [182]. Further, DHT improves image prediction when combined with machine learning algorithms [183]. It must be noted that DHT is appropriate for use cases where there is a constraint sensor node, such as implementation in improving security [184]. It is considered a fast algorithm implementation without creating overhead in such constrained devices.

2.9 Existing Data Architectures for WSN

Wireless Sensor Networks provide raw data that machines cannot understand, process, and interpret. Much effort is spent in integrating WSN and providing service to end users, and less research effort in interpreting and enabling machine-to-machine understanding of sensed data context. The real-world sensed data are different (room temp., outside temp., vehicle temp., human temp., etc), and accessing, understanding, processing, and interpreting data from various sources is tremendously important. To address data interoperability among WSNs in what follows, we present semantic data architectures for WSNs.

Sense and Sens'ability: Semantic Data Modelling for Sensor Networks [185] is a semantic model for enabling heterogeneous sensor data to be annotated so as to enhance the processing and interpretation of massive amounts of heterogeneous sensor data. This architecture is based on standard formats and ontologies proposed by Sensor Web Enablement (SWE) group and SensorML data component models. Further, it extends its capabilities for unit measurements using NASA Sweet ontology.

A Data Annotation Architecture for Semantic Applications in Virtualized Wireless Sensor Networks [186] provides another semantic architecture for enabling Semantic interoperability among virtualized sensor nodes. This architecture deals with two types of sensor platforms; resource-constraint sensor platforms and devices without limitation. This architecture uses standards like SSN ontology and SenML as a data model for describing sensory information. This architecture addresses the enrichment of raw data only to nodes without constraints. However, nodes with limitations only provide a simple model, and semantic interoperability becomes an issue.

ES3N [187] proposes a semantic-based database approach tool that applies Semantic Web techniques for data management of heterogeneous sensor data, including collection, storage, and querying. It follows a multi-layered plan involving data collection, memory caching, data tagging, ontology representation, query processing, and user interaction. This work uses its ontology exported in OWL language for defining classes, concepts, and properties. Then, sensor raw data are tagged with meta-data and saved as RDF files. SPARQL query is used for querying specific data, and the response to end users is again in RDF format.

A Semantic-based Architecture for Sensor Data Fusion [188] proposes a three-layer architecture aiming to offer data aggregation, data annotation, data management, and querying functionalities. The features provided are accomplished by using Open Geospatial Consortium (OGC)[10] by extending Semantic Web techniques for processing raw data and enhancing with further semantics. This architecture can offer a scalable solution, but applying rules to the receiving data through a central unit seems to be a drawback that will indicate performance limitations.

[10]http://www.opengeospatial.org/

Semantic Web Architecture for Sensor Networks (SWASN) [189] utilizes Semantic Web technologies for facilitating the understanding, processing, and interpreting of sensor data. This work utilizes distinct parts of CommonSense [190], a multi-tier service-oriented architecture aiming to integrate sensor data received by heterogeneous devices. SWASN consists of four layers: SN data sources, Ontology layer, Semantic Web processing layer, and applications layer. It could be a good option to use different ontologies on a gateway; however, it does not provide any data modality, and no standard Ontology for describing sensor data is used. This project evaluates its architecture through the implementation of a fire emergency scenario.

Semantic Sensor Observation Service (SemSOS)[191] is another framework that provides enrichment of sensor data with semantics. This work extends the OGC Semantic Web Enablement (SWE) [11] specifications. It uses its core standards by adding data annotation to raw data with Ontological concepts and features presented by such standards. This framework enhances semantically the raw data with spatial and temporal concepts using domain Ontologies which then are used for reasoning. Even though this implementation looks very promising due to the standards they use, it is not clear enough how this framework will scale in situations when the number of sensors and the volume of the data size increase a lot because of the annotation process, which most often requires processing power and sensor nodes are devices with limitations.

2.9.1 Discussion

It is clear that Wireless Sensor Network is a complex domain and many research challenges exist. One of the challenges previously mentioned is syntactic and semantic interoperability which refers to the possibility of exchanging information, understanding, processing, and interpreting that information by machines. Considering the limitations of sensor nodes and a large number of nodes in the network, the existence of a variety of sensors that might use different formats results in heterogenous data and thus requires a flexible solution that needs to be tackled by architectures that offer WSN solutions.

These architectures should consider: a) scalable solutions due to the increasing volume of data coming from sensor networks that, if not designed properly, would reduce the overall system's performance. Thus, architectures need to consider as much as possible to avoid a single point of doing things. For instance, annotating data from a central computer/server or providing centralized data storage solutions is not highly scalable b) a data representation model, which is a significant issue to be tackled by architectures. In a sensor network with constraint nodes, it is highly important for representing collected data in a lightweight format. In those nodes, using a simple data model is crucial. For instance, using the binary format is a good option for a limited sensor platform but will result in less interoperable data. However, other formats can enable the enhancement of raw data with semantics, such as RDF and JSON-LD, that are presented in this report. c) Semantics are necessary for gathering raw data. With the introduction of Ontologies, we can describe and give

[11] http://www.opengeospatial.org/projects/groups/sensorwebdwg

meaning to raw data. d) Ontologies are another important aspect that needs consideration by architectures. Utilizing main concepts and attributes in an abstract manner offered by Ontologies would result in semantic interoperability on a more global level. It is essential to make use of standard Ontologies so that everyone speaks the same "language". e) application interface and its data representation format is another parameter that needs consideration. End users could develop different applications; thus, relying on a standard interface and the use common data format is very important.

In the following, we will evaluate the reviewed data architectures.

Table 2.9: Characteristics of semantic architectures

Name	**Scalability**	**Ontology**	**Data Format**	**Application Interface and Data Format**	**Semantics**	**Approach**
Sense and Sens'ability	Medium	SWE, SensorML, SWEET	RDF/XML	Not considering	sensor data, location	Service-oriented
Data Annotation Architecture for Semantic Applications in Virtualized Wireless Sensor Networks	Medium-to-high	SSN	SenML for constraint nodes, RDF for advanced nodes	Coap	sensor data	sensor virtualized-based
ES3N	Low	Own implementation	RDF	No info	Data, queries	Database centered
A Semantic-based Architecture for Sensor Data Fusion [188]	Medium	Sensors to high-level events mapping	RDF and XML	Mapping rules, sensor data	High-level events detection	rule-based transformation
SWASN	Medium	Hybrid, sensors, location, domain	RDF	HTTP	Service oriented	rule-based transformation
SemSOS	Medium	O&M, OWL-Time	OWL instances	Service oriented	O&M data, queries	Service-Oriented

Table 2.10: Characteristics of semantic architectures

Name	**Scalability**	**Ontology**	**Data Format**	**Protocol Interface**	**Semantics**	**Approach**
Sense and Sens'ability	Medium	SWE, SensorML, SWEET	RDF/XML	HTTP	sensor data, location	Service-oriented
Data Annotation Architecture for Semantic Applications in Virtualized Wireless Sensor Networks	Medium-to-high	SSN	SenML for constraint nodes, RDF for advanced nodes	Coap	sensor data	sensor virtualized-based
ES3N	Low	Own implementation	RDF	No info	Data, queries	Database centered
A Semantic-based Architecture for Sensor Data Fusion [188]	Medium	Sensors to high-level events mapping	RDF and XML	HTTP	High-level events detection	rule-based transformation
SWASN	Medium	Hybrid, sensors, location, domain	RDF	HTTP	Service oriented	rule-based transformation
SemSOS	Medium	O&M, OWL-Time	OWL instances	HTTP	O&M data, queries	Service-Oriented

Most of the architectures presented in the table above have addressed semantic interoperability by employing semantic web standards and ontologies presented previously by other parties (like O&M by OGC). Regarding scalability, the majority of the architectures seem not to scale very well due to the centralized approach they follow. First, providing a single unit for converting raw data with semantics in a large-scale network would lower the performance of the system overall. Imagine having thousands of nodes, and we add semantics through a single unit. The architecture presented by [185] enhances the data with semantics in nodes without limitations. It offers better scalability than others by converting raw data into semantics in each gateway. However, constraint nodes are not enhanced semantically.

Concerning the Ontologies, most of them follow standard models, or they extend the existing ontologies with other capabilities. Most of the architectures presented here follow SWE standards like O & M. However, SSN is more desirable compared to O & M as we have reviewed in section 2.7. Even though they are a good point to provide meta-data to raw data, providing domain-specific ontologies that will help to increase the reasoning and interpretation of the data seems difficult to extend in these frameworks. For instance, location is a crucial aspect that needs to be tackled by almost every modern data architecture.

Efficient processing and transmission from one node to another are crucial in a sensor

network. Thus, in order to address the energy efficiency and limited bandwidth, we cannot employee directly models with rich semantic formats and exchange unchanged data between the nodes. We need a format that is simple enough to be delivered without obstacles in this constrained environment. In this approach [185], authors have considered limitations, where the collected data are sent to end users. Providing a simple model in constraint nodes and later enhancing it with further semantics seems promising. Finally, none of the architectures have considered the use of lightweight formats like JSON or JSON-LD[12] for employing semantics to the data.

Following the above, the architecture proposed by Khan et al. [186] addresses some of the requirements we have identified in the research challenges. They offer data solutions for constrained devices and capable devices. They consider delivery of information using a constrained protocol for sensor networks that is called CoAP [13] protocol. However, employing a simple model in all relevant components results in poor data interoperable. Other solutions are not addressing such constraints, and employing SensorML from the beginning does not seem the right solution. Also, a single point of doing things would result in an inefficient solution. Further, considering the system's performance, we already reviewed different data formats, and none of the architectures have considered lightweight formats. Finally, employing domain-specific Ontologies in a different layer to enhance knowledge of the information seems that further research is needed.

2.10 Emerging Issues and Research Questions

The literature review enabled a comprehensive study of the phenomenon under investigation. Hence, elaborated technologies like Wireless Sensor Networks, their features, issues, and challenges that need to be tackled, existing middlewares in the market, and data standards, ontologies, and semantic middleware; all this information gives us a thorough understanding and insight for adopting the right network architecture for enabling the development of future highly dynamic WSN applications and then propose a data architecture for the adopted network architecture. Altogether, they allow us to address the research issues presented in the first section.

As result, several research questions emerge:

1. How can the proposed data architecture overcome the limitations that existing data architectures have?

2. In what context does the proposed data architecture improve the performance of the existing adopted architecture?

3. How can the proposed data architecture improve the discovery of the resources?

4. What should be the characteristics of an efficient data architecture when adopting a sensor-based architecture?

[12]http://json-ld.org/
[13]http://coap.technology/

5. What are the specific criteria to evaluate the data architecture to overcome the performance limitations?

6. How can the proposed architecture improve energy efficiency in WSNs?

7. How can prediction models can improve energy efficiency in sensor nodes and by what extent can the improvement be quantified?

8. What are the specific criteria to evaluate the prediction models for energy efficiency?

2.11 Research Design and Methodology

The research design refers to the steps we are following during our research to integrate different study components coherently and logically, allowing us to address the research question efficiently. Major components of the research project, like measures, treatments/programs, and methods, are well-adjusted, so they address the primary research questions [192]. The research design and methodology are elaborated on in the preceding section.

2.11.1 Research Design and Methodology

"This work aims to identify and adopt the right network architecture that will enable well-designed solutions in extremely dynamic environments, especially in situations where well-defined and well-designed infrastructures do not exist or are not preferable. Then it will propose a data architecture that will enable the efficient communication of information in this network, considering all the challenges imposed by sensor networks stated above. A significant contribution lies in improving energy efficiency in the data collection phase while reducing transmission. The applicability of the proposed models will be validated with a real-world scenario using available data sets."

Let us describe the methodology by breaking down the aim into smaller pieces so we better understand the whole process.

1. *"...identify and adopt the right network architecture ..."*
   First, identifying the appropriate network architecture for a highly dynamic environment requires a thorough study of the nature of sensor networks from the literature review and the features they possess to build future smart applications. That's why we address the features of Wireless Sensor Networks starting from section 2.1.1. However, the most interesting part that allows us to identify and adopt the most promising architecture according to the problem domain is by reviewing the paradigms addressing the scalability, distribution, and ability to work in a highly dynamic environment. We considered the architectures that address such needs (section 2.5.3) and the evaluation of properties, challenges, and issues addressed by them. Such information gives us insights to classify and decide on the appropriate architecture and, as a result, adopt it.

2. *"...propose a data architecture ..."* Introducing a data framework that precisely fulfills the specific requirements of a Wireless Sensor Network, suits the adopted architecture, and addresses the research challenges like syntactic and semantic interoperability. First of all, the data architecture requires an understanding of the requirements and properties of sensor nodes. Second, it needs to research the existing data standards and technologies that allow us to enable semantic interoperability. Third, it requires an understanding of the components of an existing architecture. We addressed the requirements and properties issues of sensor nodes and wireless sensor networks via a literature review and the case studies other researchers have done (this can be found in Section 2.2). In section 2.6, we analyzed the current trends, data standards, technologies, and existing frameworks allowing us to address the second challenge. Understanding the agent-oriented approaches and details from the middleware allows us to propose a data framework for enhancing machine-to-machine communication by addressing the needs of WSNs.

3. *"...A significant contribution lies in improving energy efficiency in the data collection phase while reducing transmission ..."*
   A central role in the architecture is to have energy-efficient models that allow us to increase network lifetime via the prediction algorithms. Thus, the key contribution will be to propose new models that can calculate coefficients and start predicting values on the sink nodes, and all these will result in fewer data transmissions between the nodes.

   Following the prediction models, we will evaluate the performance of the proposed approaches regarding energy efficiency and accuracy.

   Evaluation The required performance measurements include the following:
   1. Energy savings: We need to measure how much energy can be saved while reducing transmission and maintaining data integrity at a desired level.
   2. Data Transmissions. We can show how much data transmissions can be reduced using the proposed prediction approaches.
   3. Processing Time. What is the processing time to generate coefficients, and whether such processing time could impact the energy efficiency of sensor nodes?
   4. Data Accuracy. These include mathematical metrics to evaluate data accuracy and compare with existing approaches. Via the following metrics, we can evaluate whether the model can be a right fit: Root-mean-square-error, mean-absolute-error, mean-square-error, and mean-square-absolute-percentage-error.

   Evaluation Criteria:

   (a) ***Energy consumption:*** Several techniques/models exist today for measuring the energy consumption of the sensor nodes in the Sensor Networks. These include simulation tools like Tossim [193], Atemu [194], Contiki [195], theoretical estimations, mathematical modeling languages like PetriNet [196] and in situ measurements. Theoretical estimations rely on a mathematical formula to calculate the energy of sensor nodes. Simulators are often device specific or not mature enough to execute the code we provide on sensor nodes. Thus, for the purpose of

this study, we rely on theoretical estimations for evaluating the performance of proposed approaches with regard to energy efficiency.

Performing direct measurements that rely on physical sensor nodes provides the best accurate results of energy estimation [197]. These measurements are often performed in real-time, which can be seen in [198, 199, 200]. Direct measurements require a specific device (ex., oscilloscope[14]) that measures energy in the sensor nodes.

In our case, we will rely on theoretical foundations to evaluate the energy performance, while incorporating real data from direct sensor measurements. The real data measurements will need to be periodically monitored to ensure the most accurate statistics are included.

4. Compare performance metrics from point nr. 3. with a similar approaches using the same metrics identified from point nr. 3.

Simply, we have summarized our methodology in figure below.

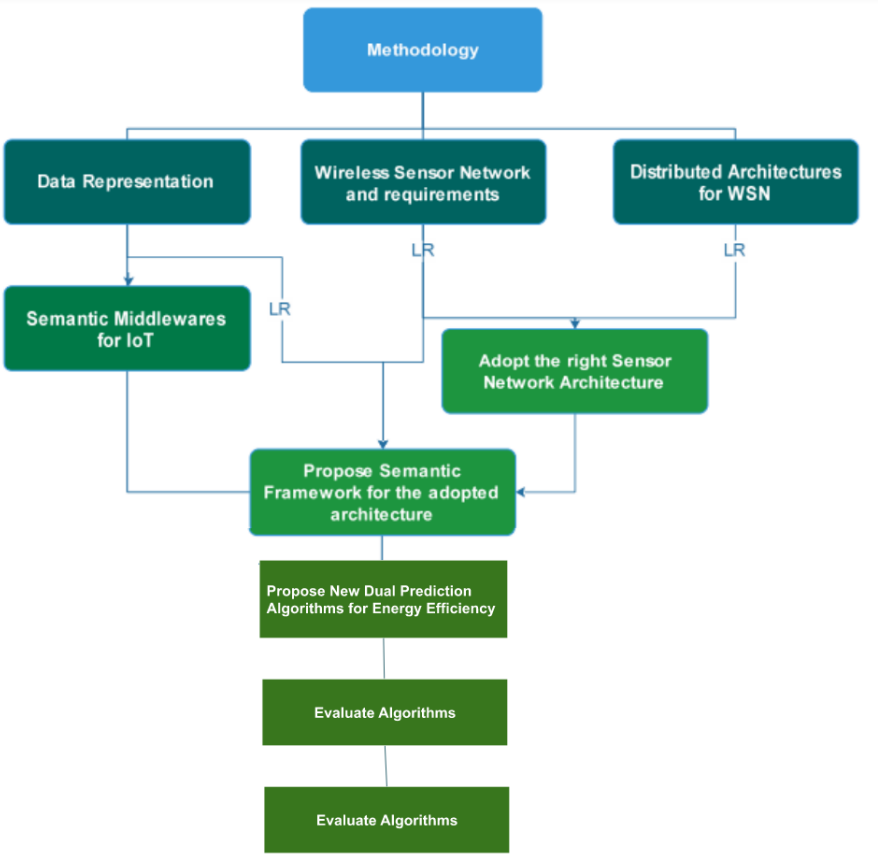


Figure 2.6: The methodology presented visually

[14]a device that displays oscillation on the screen.

2.11.2 Discussion

This research work aims to evaluate and adopt the network architecture that works in a highly dynamic environment, then propose a data architecture that will enable effective communication of information in sensor-based networks considering all challenges that are imposed there. Primarily, the intention is to compare the existing architectures and analyze the dynamic requirements of the already implemented sensor-based networks. The literature review presented in the previous chapter will help to identify the appropriate network architecture. By this, the aim is to propose a new data framework that will fulfill the requirements of a sensor-based network and overcome the limitations of the already existing frameworks.

The proposed data architecture will ensure to address the research challenges imposed by the wireless sensor network (limited bandwidth, memory, and computational power) through data modeling using existing standards reviewed in section 2.7. Through the existing data standards that will be part of the proposed data architecture, machines can efficiently process the vast amount of data coming from sensor networks and overcome the limitations of the already existing frameworks. Similar architectures are presented in the literature review and discussed thoroughly in section 2.9. These architectures try to create data interoperability in their corresponding architectures by integrating semantic standards. However, they do not address the limitations of Wireless Sensor Networks by employing directly rich semantic formats nearby nodes. We believe that by proposing a data architecture that employs a data model that fits the needs of constraint sensor nodes and later employing semantic standards where no single unit of conversion exists and considering domain-specific Ontologies, we are tackling some crucial challenges of Wireless Sensor Networks data architectures.

We are talking about the vast amount of data generated by sensor networks. One important aspect to consider in data architecture is data redundancy and accuracy. We need to adopt the right ontological language that does scale efficiently. Directly we cannot apply ontologies due to constraints imposed by SN. Thus, applying the correct technique for annotating the receiving data affects the response time.

Finally, new prediction algorithms can improve energy efficiency in sensor nodes. This research work will evaluate energy consumption, processing and the impact on energy efficiency, and the accuracy of the proposed models. We can increase or decrease energy consumption depending on the data/packet we transmit. The larger the packet we transmit in a sensor network, the less energy efficient the node is. Thus, energy consumption is a requirement to be considered when designing the data architecture. The core part of the architecture is to include algorithms that highly impact energy efficiency via coefficient transmissions and use the coefficients to predict future values. Thus, the fewer data packets we transmit, the more energy efficient the network becomes.

Chapter 3

Adoption and Enhancement of Sensor Networks Architecture. The proposition of new Data Architecture

In this chapter, the focus is to adopt the right network architecture based on the dynamic requirement presented in the literature review and elaborated in more detail in this chapter, extend the adopted network architecture with additional features by proposing a regional network, and finally propose a new data architecture that can tackle additional data requirements, specifically interoperability and energy efficiency.

3.1 Adopting the right Network Architecture

In Chapter 2, WSN middlewares paradigms were discussed, pros and cons were provided, and the most suitable architectural paradigm for the specified criteria was identified. Following such a paradigm, middleware approaches were identified, and a preliminary comparison was given. Despite the similarity that can be seen from that specific comparative study which provides a high-level overview of the qualities of each approach, there are differences if a detailed comparison is considered.

In agent-oriented approaches, it is evident that the efforts to provide intelligence to the nodes by supported agents increase the network longevity and adapt to the environment's dynamics without external interference. The network dynamism is achieved through the support of intelligent algorithms that can make the decision locally and algorithms that reduce energy through data reduction or estimation models. Several middleware approaches are designed to work for specific hardware platforms, such as those working for Sun SPOT nodes from Sun Microsystems. In addition to this, these approaches provide a communication model using tuple spaces. Despite advantages, there is no convergence to the semantics where machines could apply real-world knowledge, provide linking to analogous concepts, and as a result, increase reasoning capabilities [201].

Section 2.3 - 2.6 is viewed as a decomposition to fulfill the primary thesis objective given in Section 1.3; *"adopt successful network architecture that is robust, enable mobile ad hoc*

network solutions, and afford an extremely high number of sensor nodes and concurrent users in real-time so to form a stable architecture for further enhancements."

In what follows, a set of criteria and qualities of a desired middleware are presented in more detail, and a detailed comparison is performed under specific criteria which would result in the adoption of the most "righteous" architectural middleware.

3.1.1 Ultra Scalability

In an era where the number of connected "things" has drastically increased, it becomes necessary for middleware to accept new network connections without losing efficacy and performance. This need for scalability is required in all components of the middleware, from efficient communication to service delivery. Any form which requires manual configuration, installation, and management is not feasible. Multiple devices need to connect on a daily basis, and the middleware should be capable of dynamically injecting the service functionalities, service discovery, and composition, as well as enabling efficient communication with other entities in the network and the Internet. On the other hand, middleware algorithms must be able to perform efficiently to process a large amount of generated data; the persistence layer must be capable of affording the preservation of data, while query answering requires an immediate response with accurate results. Moreover, scalable middleware is also essential for security purposes and the management of billions of connected things.

Several questions should be considered when providing scalable systems in IoT and WSN in particular.

- What happens when the frequency of data increases by a factor of a hundred?
- What is the time to process the query, and what kind of queries can be supported?

The only way to keep the trend is to scale quickly with a set of rich behaviors and intelligent algorithms with self* properties and not burden developers with implementation tasks whenever a new device is added. If not possible, the middleware should make it as trivial as possible for the developer to extend the behavior of middleware concepts easily and scale with new "things."

Among agent-oriented approaches, some middlewares are dedicated to working for a specific platform such as Agilla [93], MAPS [97], MASPOT [98] etc. To collect information from other nodes, other than the approaches that are able to run, is almost infeasible, or this requires building the code from scratch and running on that specific hardware.

As the information generated by sensory devices are different data models, a middleware could provide an interface that accept the information streams and convert them to a common standardized format that is accepted among all entities in the network. Then, the interface should allow to easy configuring of the resources in the local network.

3.1.2 Robustness and Adaptivity

One critical factor of the middleware using Wireless Sensor Networks is to react to disruptions of broken links and node failures. Sensor nodes may also fail as a result of battery harvesting, hardware faults, and similar environmental parameters that may affect the communication between the nodes (high temperature, low temperature, rain etc.). All of these

may affect the accuracy of the receiving values, which could potentially require new settings for the nodes or new network configuration in case of node failure. These unintended features are very desirable when performed automatically by the middleware. No matter the failures and the environmental changes that may affect the physical layer, the architectural middleware approach should recover and adapt to the new condition without affecting the overall performance.

In the upper layer, as the number of end users grows, the performance of the middleware should remain as in the case of small scale and fulfill all the requirements without delaying the results.

3.1.3 Autonomy and Self-organization

To extend the utilization of the IoT at a global scale, it is of critical importance for the middleware to be able to perform tasks without external input, manage the components and devices in an integrated manner, optimize the use of the resource, and expand automatically when there is a need and continually increase the real world knowledge for reasoning and guidance. For instance, Agilla [93] uses the local knowledge of the nodes to make decisions without user input. More recent middleware such as SAMSON [101] and EDBO [202] exploit more advanced algorithms in decision-making; to optimize network lifetime and adapt to network dynamics.

IBM self-management properties of an autonomous system [203] comprise self-healing, self-configuration, self-optimization, and self-protection. In case a node enters the network, it automatically set-up the connections and starts communicating with other nodes. If nodes generate false values, the system may detect and stop sending erroneous data. This could be an indicator to self-optimize the network by preserving energy usage by not delivering erroneous data.

Another property that is highly important and very related to the autonomy of the sensor network is self-organization: a collection of entities/agents that interact and coordinate their actions in order to achieve the global goal efficiently [51]. The interactions of entities in the middleware should happen with the mandatory/needed ones until the common goal is fulfilled. From the agents perspective, the entities/agents know with whom and when to interact for a specified goal. Therefore, the agents interactions can help the whole system to remain stable in such dynamic environment like WSN, and also help sensor nodes to reduce energy consumption when not all sensor nodes are involved in communication. Having "intelligent" communication among agents can help the WSN network to extend its lifetime. Finally, in an autonomous IoT, the middleware could provide the possibility to reason over the data using real-world knowledge and make decisions partly drawing upon machine learning (ML) and optimization algorithms. Still, the possibility to make informed choices when important actions needs to be taken should be left to users[1].

[1] http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=EP/N014243/2

Table 3.1: Detailed agent-oriented middleware approach comparison

Name	Dynamics	Autonomy	Robust and Avail.	Scalability	Self-org.	Energy	Mem. and Proc.	Heterogeneity	Semantic Interoperability
Agilla	✓	✓	✓	✓	✓	***X***	✓	✓✓	***X***
Impala	✓	***X***	***X***	✓✓	***X***	✓	***X***	✓	***X***
ActorNet	✓✓	***X***	NI	✓	NI	✓	✓	***X***	✓
UbiRoad	✓✓✓	✓✓	✓	✓	✓	NI	NI	***X***	✓✓
MAPS	✓✓	NI	NI	✓	NI	***X***	✓	***X***	***X***
MASPOT	✓✓	✓	✓	✓	***X***	***X***	✓	***X***	***X***
TinyMAPS	✓	NI	✓	✓✓	NI	✓	✓	***X***	***X***
UBIWARE	✓✓	✓✓	✓	✓	✓	NI	NI	✓✓	✓✓✓
EDBO	✓✓	✓✓✓	✓✓✓	✓✓✓	✓✓	NI	NI	✓✓	***X***
SAMSON	✓	✓✓✓	✓✓	✓✓	✓	✓	NI	✓	***X***
DIAT	✓✓	NI	NI	✓✓	***X***	***X***	***X***	✓	***X***
EAGILLA	✓✓	✓	✓	✓✓	NI	NI	NI	✓	***X***
Sensomax	✓✓	✓✓	NI	✓	***X***	***X***	✓	***X***	***X***
Bisnet	✓	✓	✓	✓✓	✓✓	✓✓✓	✓	***X***	***X***
FIoT	✓✓	✓	NI	✓	✓	***X***	***X***	✓	***X***
ACOSO	✓✓	NI	NI	✓✓✓	***X***	***X***	***X***	✓	***X***
Al-Sakran	✓✓	***X***	***X***	✓	***X***	***X***	***X***	✓	***X***

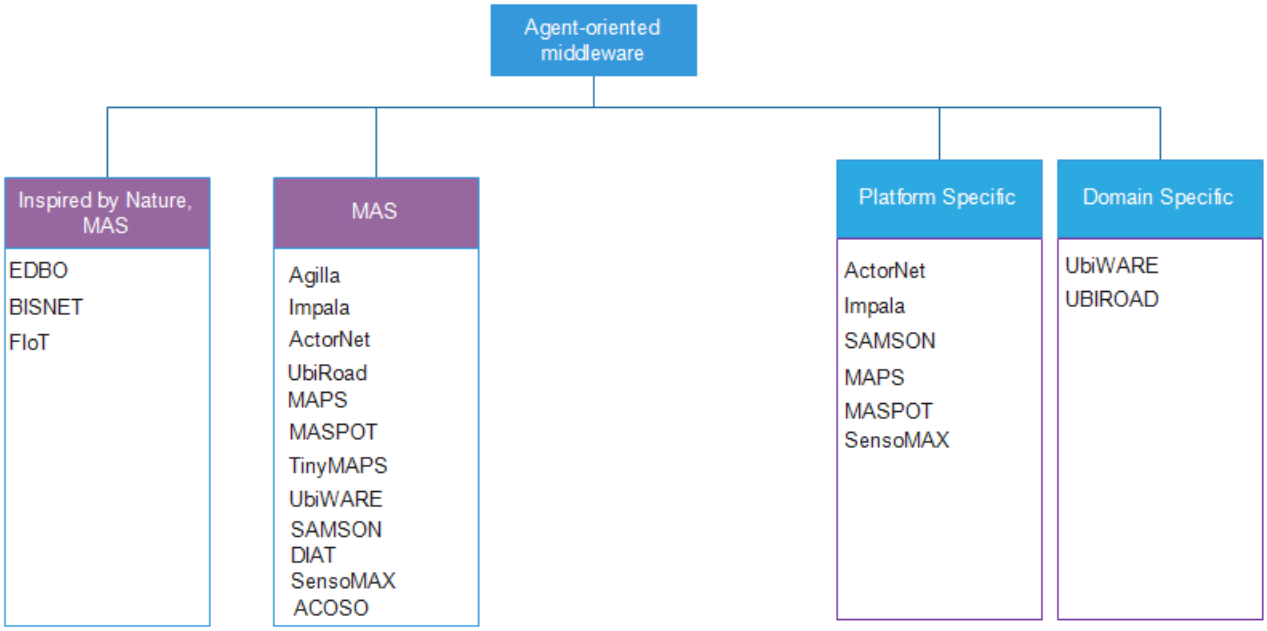
Legend: NI: No Information, X: Not supported ✓: Low Support ✓✓: Moderate Support ✓✓✓: Strong Support

3.1.4 Energy, Memory, and Processing limitations

It is already discussed in the last two paragraphs of section 2.3 (not only) the limitations in energy, memory, and processing power. Thus, we will also derive the topics and compare the middlewares according to these criteria.

There is no universal agreement on what Architecture Description Languages (ADLs) should represent, particularly regarding the architecture's behavior. Representations currently in use are relatively challenging to parse and are not supported by commercial tools. Most ADLs tend to be vertically optimized toward a particular kind of analysis.

Figure 3.1: Agent-oriented middlewares



3.1.5 Discussion

This chapter discussed in detail a set of requirements for architectural middleware using a sensor network. This was modeled using the research literature within Chapter 2. The need to support a highly scalable dynamic sensor network was noted. The heterogeneity of sources, including the generation of information, presents one of the criteria; this is because of the need for open generic middleware that would easily extend the network as per user preferences of sensor nodes. In a related manner, the middleware must provide self* properties in order to tackle the requirements of sensor networks and near-future applications.

Following such requirements, we categorize the agent-oriented approaches into middleware architectures built upon bio-inspired concepts and agent-oriented approaches as can be seen in Figure 3.1. Natural processes inspire the bio-inspired solution to build intelligent middlewares that enable self* properties to emerge. In such cases, agents via local collaborations could emerge toward the intended goal at a global level. From an unstructured network such as WSN, this could be a new viable structure created via agent interactions. Additionally, the created structure is a new way to create an even larger structure, totally different from the first one, which could be extended further, and so on. Moreover, all these processes applied to software agents are inspired by biological processes such as bee colonies in Bisnet or several heuristic algorithms in EDBO.

On the other hand, agent-oriented middleware is developed using agent-oriented programming, which is represented by object entities with additional characteristics and behavior. Agents communicate with each other and possess a certain degree of awareness which is defined by a set of constraints. The majority of middlewares presented in this thesis have

implemented partly sets of self* properties with a high potential to be improved further.

Bio-inspired algorithmic solutions are becoming a fundamental solution for designing complex dynamic systems. Such approaches are used for service composition [204] and have been successfully implemented in several agent-oriented middlewares (BISNET, EDBO, FIoT). They could provide an autonomous solution in the area of Wireless Sensor Networks addressing the requirements already presented.

Regarding diversity, most of these approaches need to tackle heterogeneity which is fundamental for the expansion of future IoT applications. Most agent-middlewares are tailored for a specific node or specific operating system, and extra work should be devoted to a different target to handle diversity. There is a demand to offer plug & play solutions using different sensor nodes and, thus, different sensor networks. For this thesis, middlewares tailored to a specific family of sensor platforms are not considered potential candidate.

Considering the scalability parameter, the bulk of solutions are able to offer a certain degree of scalability. Solutions such as agilla, impala, MAPS, and a few others are able to perform well locally while performing at a network level could be questionable due to the overhead that is created. Also, several manual configurations are required to be performed before initiating the monitoring process, which leads to leads to less scalable systems in terms of integrating new devices in the network.

Based on the facts and comparison table presented with comparison criteria for this thesis, the most relevant middleware are considered SAMSON, Bisnet, and EDBO. Recently, Bio-inspired solutions have been the preferred alternative compared to static specified solutions, and thus, Bisnet and EDBO stand on the tight list as the preferred architectural middleware approach. Comparing the last two, EDBO wins by a few points, as shown in Table 3.1. Thus, according to the criteria specified, the most suitable architecture is EDBO.

3.2 Regional Sensor Network Architecture - proposing an extension of the adopted Sensor Network Architecture

The proposed architecture is a refinement of the IoT architecture proposed in [202], which is a bio-inspired middleware optimized for cyber-physical systems. The newly proposed refinement is the outcome of an attempt to adopt the architecture proposed in [202] in the health monitoring scenario. It tackles all the requirements and challenges imposed by WSNs (section 2.3), and it will be called from now on "eXtreme Sense Network" (XSN). XSN aims to serve as the primary infrastructure that will enable any authorized consumer to perceive the required sensor data as if connected to the central nervous system of an organism. It facilitates a plug-and-play solution to add a new sensor to the network at any time with minimal effort. Then, service will be automatically provided, and discovery will be enabled in a fully decentralized manner.

3.2.1 XSN Overall Architecture

XSN comprises the "Global Network (GN)" and a number of "Regional Networks (RN)" (Fig. 3.2). The GN is an overlay network of Biobots as proposed in [202], and the proposed refinement that satisfies all mentioned WSN challenges is designed in the RN. The RN consists mainly of two different kinds of nodes; *XSN Platforms* and *XSN Nodes*. The role of *XSN Platforms* is to facilitate a platform that hosts primitive sensors and enables their effective communication to the rest of the regional layer. *XSN Nodes* facilitate the connectivity of all *XSN Platforms* in the regional layer and interfaces this layer to the global layer when required. Connectivity with the GN is not required, and the RN can work autonomously.

The global layer is a realization [202] of a research prototype network that achieves several self* properties using a bio-inspired solution named EDBO [205]. Thus, the global layer is composed of biobots (logical nodes) that are realized in a biospace (middleware). Biobots can communicate with *XSN Nodes* if in range; hence a connection between two layers is created.

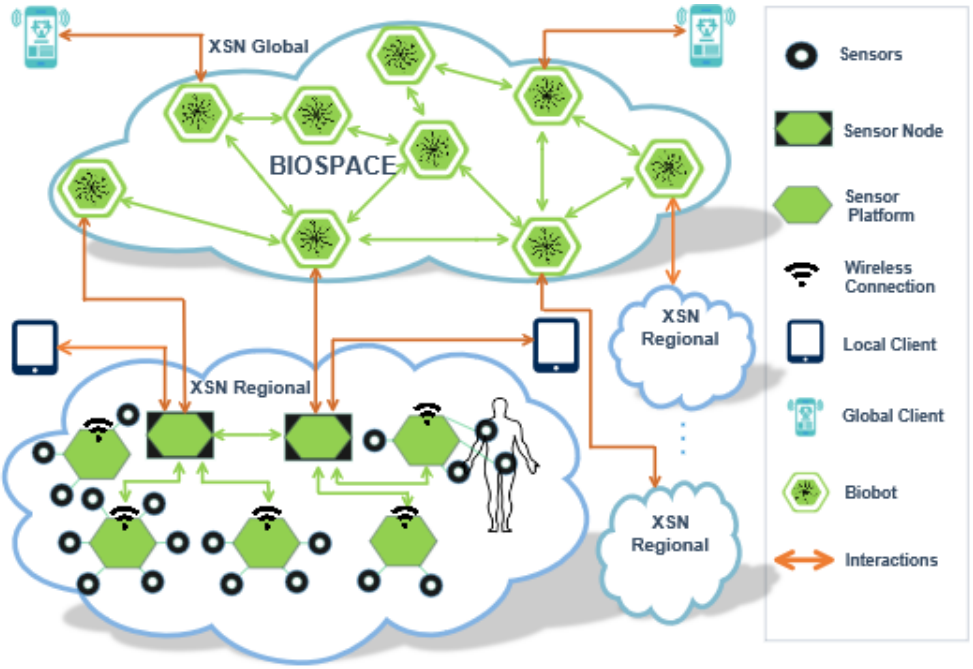


Figure 3.2: Overall system (XSN) architecture

3.2.2 Regional Network

The new concept introduced in this paper is the "Regional Network", which constitutes of each local deployment that allows direct access to sensor information. The system components of RN are:

- *XSN Platform*: This is an embedded system with integrated sensors used to monitor the required environment, in our case, vital signs of humans and other useful measurements

of their surrounding environment. It employs an energy-efficient short-range wireless network interface that enables broadcasting sensor information to *XSN Nodes*. Sensors are connected to the XSN Platforms and periodically collect sensed data that need to be sent to the XNS Nodes.
In this health monitoring scenario, the Libelium WaspMote build on ATmega 1281 micro-controller was chosen as *XSN platform*, which is an open-source platform with low energy consumption, battery-powered, reasonable technical and user documentation, and a growing community of users. It is an improvement over the Arduino microcontroller with a built-in accelerometer, thermometer, and sleep mode configuration for better power management. A plethora of sensors is available for sensing the required information via sensor boards that are either dedicated to their product lines or prototyping sensor boards to integrate any sensor. WaspMote also integrates XBee wireless interface for wireless communication based on IEEE 802.15.4 standard. The XBee shield is appointed directly on the WaspMote; meanwhile, the Connection Bridge allows to use of the same shield on Raspberry Pi (*XSN Node* in our case) to ensure compatibility. The Pi offers GPIO[2] and device peripheral compatibility for Arduino via an optional board.

- *XSN Node*: This represents a small device equipped with a microcontroller and wireless network interface to receive data broadcasted by the *XSN Platforms* and then forward to the nearby nodes by realizing mobile ad hoc network (MANET) routing. This *XSN Node* ensures connectivity among multiple nodes in the same RN. Its position can be stationary or carried by a tracking device. The processing capabilities are enough to create network services accessible by clients. In the context of the RN, it plays the role of the sink in the formed WSN. They include an interface component that plays the role of the gateway router for the RN and allows *XSN Nodes* to communicate with *Biobots*, which are the nodes of the "Global Network."
  To prototype an XSN node, we used Raspberry Pi, a mini-computer capable of running an operating system (lightweight Raspbian OS was selected). To create communication compatibility between the XSN platform and the XSN node, the Xbee connection bridge was used. In the XSN Node, TP-Link TL-WN725N Wireless Interface is selected to connect to the WiFi network due to low power consumption and high transmission speed. The mesh network and ad-hoc network connectivity are achieved by using BATMAN and IPv6 protocol stack in each node. These will allow limitless network addresses. Thus, an XSN Node can discover other XSN Nodes within the communication range using BATMAN ad-hoc networking. Then, the local client can access the sensor services provided by XSN nodes using HTTP requests, and the responses it receives are in a simple JSON format. Unlike the local client, to access the services by a global client, BioBot provides the adapter implementation for XSN Node services.
- *Local Client*: This represents an optional end-user with the option to access the sensed data from the regional network. This user may query data directly from the *XSN Node* since it is part of the regional network but can also access the distributed service

[2]GPIO (general-purpose input/output) are input/output pins on an integrated circuit whose behavior can be controlled at run time.

provided by biobots in the global network.

3.2.3 Global Network

The bio-inspired agent-based distributed system is part of the IoT architecture that realizes the "Global Network". The GN middleware enables the discovery of service for the sensor data and allows seamless access from remote locations over an unstructured distributed artificial network of XSN middleware nodes called *Biobots*.

- *Biospace*: This is the platform that constitutes the basis for creating agents that can serve sensor data.
- *Biobots*: These agents access sensor nodes via RESTful service requests to access and provide information collected by sensor platforms. They can communicate in a distributed manner with each other for service discovery purposes.
- *End User Systems*: Represents applications that access sensor services by communicating with the discovered Biobots. The end-user systems are installed on user devices (such as tablets or smartphones) that remotely access sensor data or remote servers that collect and process sensor information.

3.2.4 Discussion and Conclusion

In this chapter, we presented a refined network architecture called "Regional Network" that, together with the "Global Network" provides a complete architecture called eXtreme Sense Network (XSN). The XSN architecture presents abstractions on different levels; i) the GN is inspired by emergent natural phenomena and enables the integration of physical systems to the Web, and ii) the RN, which aims to provide an abstraction for the physical layer of sensor networks. Together, they can solve critical issues; scalability, heterogeneity, robustness, self-adaptability, self-optimization, and resource discovery. The XSN architecture enhances network flexibility by allowing sensor devices to easily connect to the network in an ad hoc manner via several wireless interfaces that also contribute to service discovery. In contrast to other middleware approaches, this applies the idea of a bio-inspired model by introducing several bio-inspired algorithmic solutions for decision-making, birth, and death of biobots, resource optimization, and discovery. Using the XSN architecture, a WSN application can use an initially unstructured network that will adapt and change its structure autonomously according to the changing demands and load in the network.

The proposed XSN architecture achieves scalability by allowing clusters of *XSN Nodes* in the RN to be connected with global nodes called BioBots, creating clusters in the GN for service discovery and processing. Each *XSN platform* is connected with *XSN nodes* via short-range wireless communication, whereas nodes create multi-hop mobile ad hoc networks in the RN. The presence of various wireless interfaces in a node using the BATMAN[3] protocol provides a discovery mechanism and allows to integrate multiple *XSN nodes* which ensures scalability while empowering heterogeneity. As the number of *XSN nodes* increases, more

[3]https://www.open-mesh.org/projects/open-mesh/wiki/BATMANConcept

dynamic links are available for data transmission, and robustness is achieved. Further, the *XSN node* that serves as a gateway delivers the data to a BioBot. As the number of nodes increases in the RN, the number of Biobots increases as well due to emergent properties of the Biobot; that is, by increasing the energy level through the service discovery mechanism, replication occurs, and more logical connections are made with other Biobots. As a result, the concentration of BioBots close to the RN where the load exists increases. In addition to this, the communication of two *XSN nodes* in different RNs is handled via BioBots that can communicate with each other and exchange the necessary information to make local decisions concerning the services they offer.

At the middleware level, the communication between devices and end-users is performed via BioBots, which provides an adapter layer that hides the heterogeneity of different data formats and device features. The current implementation accepts XML and JSON format in the form of rest services offered by an *XSN node*. This can easily be extended with additional data formats.

XSN architecture accomplishes autonomy by providing several self* properties to the system. Self-organization is achieved when discovering the services from the connected "Regional Networks". The more successful the response ratio for the end-user queries, the stronger the relationship among the Biobots. Otherwise, the inability to provide such service weakens the relationship between Biobots, and as a result death of an agent occurs, which makes other BioBots change their structure. Thus, self-adaptivity and self-organization are achieved through load balancing via energy maximization.

The RN layer offers the ability to work independently without relying on the GN layer, comprising of components that allow *XSN Platforms*, which are resource-limited, without any pre-configuration, to deliver the data to *XSN Nodes* (node with multiple interfaces) that offer the required services. In the RN, the security issues are addressed by configuring the XBee with a 128-bit Advanced Encryption Standard (AES) key to ensure confidentiality and integrity. The security mechanism is implemented from *XSN platform* to *XSN node* while other layers of architecture need further security considerations.

Having such a complete XSN network architecture could find numerous applications, including but not limited to applications in an open field (e.g., a camp) without a well-established network infrastructure aiming to control individuals' physiological and health status as well as use for operational purposes. The whole network architecture could offer a cost-effective solution that supports such dynamism. Other potential applications include agriculture, traffic, and transportation, where each device could be treated as an XSN node, and the biospace could create a highly intelligent control system.

A prototype implementation of the proposed architecture has been used in a health monitoring scenario enabling remote monitoring of vital signs and providing proof of concept. In the next section, we aim to add features and functionalities supporting data interoperability that will enhance the automatic discovery process and enable machine-to-machine communication and shared understanding for better local or global decision-making. The limitations in XSN platforms forbid the use of semantic technologies directly. Thus, several data representation standards are investigated for this purpose. Based on each node's limitations and requirements, we can use different standards at different levels of the network.

3.3 New Semantic Data Architecture for Sensor Network

Previous sections of this chapter describe a desiderata middleware for an unstructured, dynamic sensor network; we adopted the right architectural middleware approach based on the functional and architectural requirements (subject to this thesis) of the IoT and WSN in particular. This section considers the adopted sensor network architecture and discusses the design of a new data framework for the adopted middleware, which aims to further extend with additional requirements already identified in chapter 2 under data requirements. The framework is termed ESIADS; the name stems from the problem domain under discussion. It stands for Enabling Semantic Interoperability in Artificial Distributed Systems in the domain of Wireless Sensor Networks.

ESIADS design considers the resource limitations of the sensor node, provides semantic interoperability by utilizing de facto standard semantic web technologies in sensor networks and semantic web languages and adapts the framework to the adopted architecture.

The new data architecture presents the second objective of this thesis which is:
Objective 2: " propose a data architecture for the adopted architecture in order to enable syntactic and semantic data interoperability considering the limitations of Wireless Sensor Nodes (bandwidth, memory, CPU, and energy)".

Many proposals address specific IoT requirements, but only a few have considered the data requirements. Some previously discussed data architecture proposals have addressed specific needs, such as offering machine interoperability by employing semantic formats and using Ontologies. These data architectures mainly offer centralized solutions, and there is still room for improvements in treating IoT data requirements such as qualitative data and machine interoperable using lightweight data formats by considering constrained sensor nodes and handling the scalable volume of IoT-generated data.

We are proposing a data framework for the proposed EDBO architecture so we are able to address the research problems presented in the literature review in Section 2.6.2. In what follows, we will present the details of such a framework.

Although machine-interpretable data (semantics) are employed and recognized by several architectures, more work is necessary on adopting semantic technologies for agent-oriented architectures. Agents could overcome the constraints of sensor nodes and employ semantic web technologies in their architecture. All these are due to the power of autonomy they can maintain and the possibility to set the processing of incoming data outside the sensor layer. Considering that WSN is a hot research area, data issues are still present in many middlewares. Additional attention is necessary to model a layer of semantic data that could highlight some challenges. In Figure 3.3, we have presented a new semantic data architecture that can tackle some of the data-related challenges.

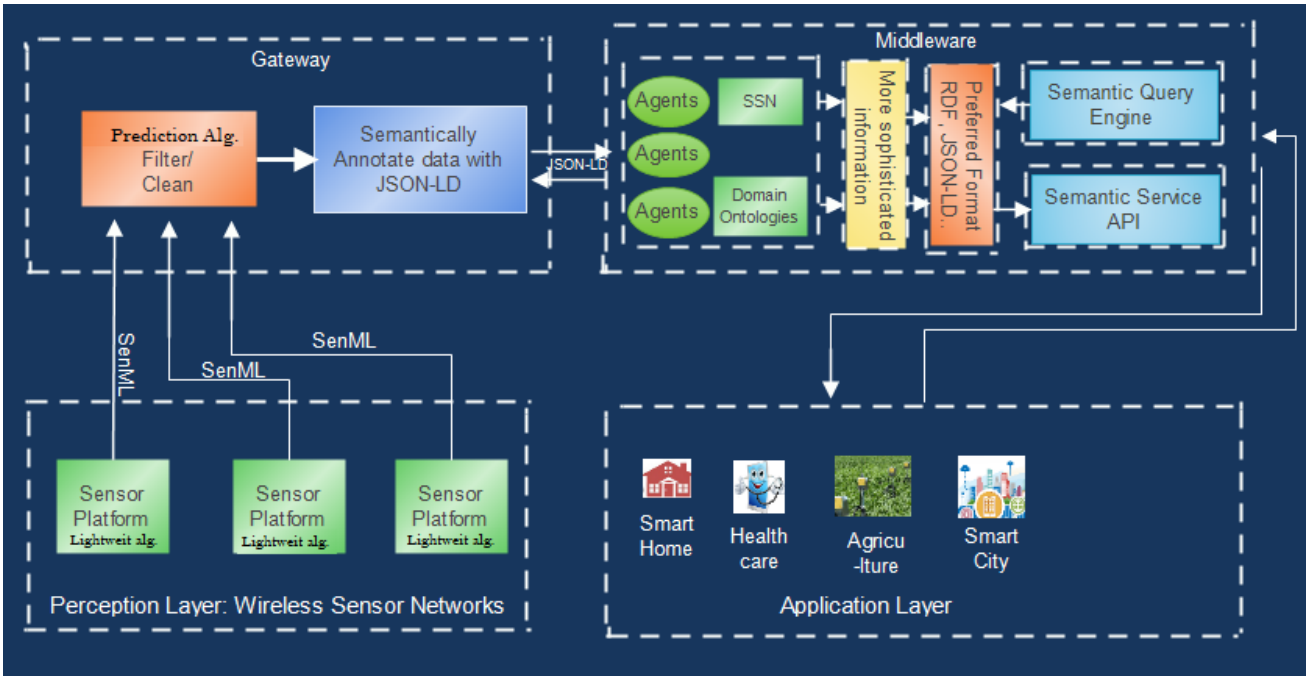


Figure 3.3: Architecture Proposal

3.3.1 Architecture Layers

In what follows, we describe the new data architecture layers.

Layer I: The wireless Sensor Network layer consists of wireless sensor platforms forming a mesh network using low-power radio frequency protocols such as ZigBee, Z-Wave, etc. These platforms are constrained devices with limited capabilities. Thus, it is difficult to address the issues of providing semantics directly to the nodes or producing qualitative data. The first layer employs simple formats that are lightweight and can easily travel the information from node to node with less energy possible. Basically, in these wireless sensor platforms, we cannot preprocess the information to enable quality information to be sent throughout the other nodes until it arrives at the Sink node. However, employing a simple format with simple semantics could easily enable this information to be preprocessed, filtered, and cleaned in nodes with processing power.

Limited Bandwidth and Energy Efficiency - The maximum bandwidth for transmitting the information using the employed protocols in WSN networks is 255kb. The available payload for a packet traveling from one node to another using ZigBee protocol [206] is 116 bytes. Next, the appropriate data format to describe the sensor description is chosen to be SenML, specifically designed for constrained devices. In SenML, we can represent the sensor information in different formats, while JSON is the preferred format for easy processing and better energy utilization than XML. Thus, keeping and transmitting information in a small packet efficiently makes use of bandwidth and results in less energy spent.

Why SenML? SenML is a lightweight format that allows data modeling in different formats. SenML is created to be used with constrained devices. Among the standards already discussed in section 2.7, SenML provides a means to model the data in a format appropriate to the needs of the platform. If the platform is very constrained, one can model the data using the CBOR format or EXI, which cannot be accurate for other standards.

Suppose we have two sensors measuring different properties (i.e., temperature sensor and glucose). In SenML, such a description looks like in Appendix A, where the packet size is 107 bytes, which is the maximum size supported by low-rate wireless technologies (IEEE 802.15.4) that are 116 bytes [206]. The same measurements are described by the SensorML standard (Appendix B). There are 1100 bytes needed for transmitting the same information. This number is high and exceeds the max payload for low-rate wireless technologies. Most of the time, many packets penetrate the same route, and thus delivery delay is expected. The lower the packet size, the faster the packet is transmitted. As a result, SenML seems an adequate protocol for transmitting the packets over constraint communication mediums. Also, it supports different formats (JSON, XML, Exi).

In the sink node, the transmitted information can be transformed into the semantic format and annotate the raw data with meta-data since there are no limitations on the processing power. The sink node can be any node with high capabilities. We can do processing without any limitations and add additional knowledge to the receiving data.

It is important to note that, on the first layer, we employ lightweight algorithms (perception layer) when it comes to the data collection phase. There will be running algorithms on the XSN platform and XSN node. The algorithm that runs on the XSN platform will generate coefficients of a function based on the input values. Once the coefficients are generated, they will be transmitted to the XSN node via the data format explained above. These coefficients will help the XSN node generate the real values via prediction during a scheduled period of time and predict future values. In this case, the XSN platform can stay in sleeping mode while the XSN node predicts future values at a given period.

Layer II: Second Layer would consist of nodes with processing power and no limitation boundaries. As a result, these nodes can process the information locally before sending the pieces of information to the middleware. Also, this will allow semantically annotating the data with further semantics using Ontologies. Although there is no standardization for Sensor Ontologies, we employ SSN ontology that seems promising and quite standardized for the semantic representation of these data. Thus, in this Layer, there are algorithms that will automatically transform the SenML to SSN, specifically using the JSON-LD data format. The second Layer would also help agents to make decisions locally based on the global needs of the middleware. Having several local networks and appointing an agent or some agents in each network would enable dynamic cooperation and achieve global intelligence at a higher level. It will also help to optimize the use of the resources at the local level.

Layer III: In the last layer, there is a possibility for applying real-time processing techniques of the information coming from each local network or applying semantic rules to increase the knowledge to a higher level that would enable sophisticated information to end up to end-users. A critical step towards it would be discovering the employed resources in the network. At the top of the third layer, a Semantic query engine would improve the extrac-

tion of information from the lower layers by enhancing the knowledge and resource discovery process. This query will enable the agents to be activated and collaboratively respond to such requests and even more by employing semantics agents that will possibly answer sophisticated queries. Since agents collaboratively solve issues, there would be possible to enhance the performance when hundreds or thousands of nodes are accessed within the network and accessed by millions of users daily. It is because nearby agents would have the knowledge of nearby agents, deliver the request to the appropriate one, undertake such issues, and deliver the "best response" if possible. The architecture will improve data characteristics as described in the following subsections.

3.3.2 Energy Efficiency

Most of the data architectures previously presented assume sensor platforms with enough capabilities and directly employ semantics in these platforms. We have noticed that the architecture presented in [186] separates sensors with limited capabilities and those without constraints. Sensors with limited capabilities employ a flexible, lightweight format such as SenML, but they deliver the services to end users without enhancing further with semantics. Since the mesh network allows the information to be traveled from node to node, the provision of a lightweight format for transmitting information increases the chances for energy efficiency. The fewer bytes transmitted between the nodes, the more energy is saved. Thus, the proposed framework improves the energy of the sensor network from two perspectives: it makes use of lightweight and uses prediction algorithms. Employing prediction models will help greatly increase the network's lifetime, as we will see in the following chapters. The prediction models can reduce energy usage during transmission and send less data on the network. Sending just the coefficients one time and then using such coefficients to predict future values is a great way to improve energy efficiency.

3.3.3 Data Quality

In Sink Node, employing prediction algorithms will allow us to have quality data and remove erroneous ones. The dual prediction algorithms allow removing erroneous data when the data are interpolated and even extrapolated. The details on qualitative data will be part of the following chapters since, in our case, quality depends on the data predictions by the algorithms themselves.

3.3.4 Velocity and Data Scalability

- Agents could reduce data generation if these are not necessary to be delivered to other layers. Although applying lightweight formats such as JSON and JSON-LD for semantic representation of the data would reduce the amount of data that can be stored either in semantic databases or as a file stored in the specified directory. Instead of saving every sensor value from the XSN platform, the coefficient will reduce much space. Since the coefficients

can predict a number of future values, including the values within the range, the only point is to process the queries properly to extract the right data.

3.3.5 Enabling Interoperability: Syntactic and Semantic

The receiving information with SenML have few semantics. However, to facilitate resource discovery and create semantic interoperability, Ontologies are introduced. The well-known format that allows semantics to be included and makes the data publicly available is JSON-LD[4]. Following standardized format and ontologies improves data heterogeneity as well. The followed standards avoid conflicts about heterogeneous data and its meaning.

A wrapper library performs the transformation of SenML to JSON-LD. This wrapper enriches the data with further semantics. The SSN is the appropriate Ontology for semantically annotating the sensor data because of the strengths presented in the literature review (section 2.). It offers rich semantics for sensor observations and follows an upper ontology like Dolce Ultra Lite to enable higher-level semantic interoperability.

3.4 Conclusions

The work presents the research analysis and evaluation of existing architectures that address the dynamic requirements of WSNs with high potential to produce sophisticated applications. As a first initial step, we based our research on adopting the right network architecture that has already addressed specific issues and has a higher potential than others for future enhancements. As a result, we adopted EDBO as network architecture for highly dynamic WSN characteristics. Then, we extended the middleware with the regional layer that shows the applicability of middleware using a Wireless Sensor Network in a domain of interest.

Further, we have proposed a data framework for the adopted architecture that tackles interoperability issues, energy efficiency, and data quality. Specifically, we tackled the problem based on the requirements of WSNs. That is why we designed the data architecture in such a way to fit, first of all, with the adopted architecture and, secondly, handle the data requirements of dynamic systems.

After finishing all the steps discussed in the methodology and addressing the research questions, a possible future direction would be advancing the knowledge of the existing architecture. Some possible questions that could be addressed based on the data architecture proposed are as follows:

- How to improve data knowledge using domain-specific ontologies?
- How can ontologies enhance the response to user queries?
- Are we able to provide more intelligence closer to the nodes using Ontologies to reduce bandwidth?

[4]http://json-ld.org/

- How to extend the lifetime of the sensor network using intelligent algorithms?
- Can neural networks help us improve energy efficiency?

In order to address further energy efficiency, as the main contribution of the thesis is about, the subsequent chapters follow dual prediction models for the increasing lifespan of XSN platforms.

Chapter 4

Data Transmission Reduction and Data Prediction for Energy Efficient Wireless Sensor Network applications

This chapter proposes an algorithm that optimizes energy usage in Wireless Sensor Networks when transmitting information between the nodes. Since Sensor Network makes it possible to monitor physical properties and their surrounding environment in real-time, massive amounts of data are generated daily and transmitted over the network. A lot of processing power, memory, and a high-speed network are required. Further, transmitting a large amount of data consumes much energy in sensory devices and limits the network lifetime. The less the data size in data transmission, the more energy-efficient the node is. Sending 100 bits of data from node to node consumes 5 μJ [117]. Imaging sending time series data where the event can repeat the same value several times.

A novel data prediction approach is proposed for data transmission reduction from the sensor node to the base station while not breaking the data integrity. This algorithm is appropriate for sensory devices with energy constraints, memory, and computing resources. The intended proposed approach is based on the Newton interpolation algorithm combined with the arithmetic means to reduce the data transmission by sensory device and allow the same data to be generated whenever necessary on the base station. This algorithm runs on the sensor node and the gateway. Instead of transmitting data packets periodically to the gateway, we propose to send only coefficients of Newton interpolation polynomials. By having the coefficients, we can generate the same data at any time using the same algorithm that runs on the gateway. Newton's interpolation is also helpful in assessing the missing values between the data range.

The proposed approach based on Newton's divided difference interpolation polynomial allows us to achieve better energy efficiency by predicting values. It provides an excellent method to obtain reasonably accurate data when interpolating the data within the range and quite accurate data when we predict out of the range. Although, predicting data out of range requires a more often coefficient re-evaluation. Data results from the real sensed temperature show that the algorithm is quite accurate and significantly reduces data transmission over

the network, improving energy efficiency and increasing the network lifetime.

The chapter structure is as follows: we describe the design and problem formulation followed by the prediction model, where we describe the maths of the Newton interpolation polynomial and its corresponding algorithms. Then, the next section describes the results based on the real data set of the temperature collected within a day. The metrics include the effect of Newton degree variation, the number of necessary transmissions based on the math formula, and energy consumption. Further, we check the accuracy of the data (mean square error, mean absolute error, mean absolute percentage error). Finally, the discussion concludes the chapter.

4.1 Network Design and Problem Description

This research considers a regional network consisting of sensor platforms (XSN platforms), embedded sensory devices with limited resources, and sensor nodes (XSN nodes), devices with no restrictions. The XSN platform is responsible for observing the environmental values or values of the monitoring object. Then the platform is connected with powerful devices called sensor nodes (XSN nodes). XSN nodes might connect with other nodes and serve as a gateway. With the connection of multiple nodes, we form a local network.

Our approach consists of the dual prediction model where the same algorithm runs in XSN platforms and XSN nodes. XSN platform collects a few data points and creates a prediction model using the Newton interpolating algorithm combined with the means, and the same runs in the XSN node. We reduce the data transmission from XSN platforms to XSN nodes while a given threshold is not exceeded. Also, we only transmit the coefficients of the function. The provided algorithm can be adapted in every application, whether critical or non-critical. For example, in a critical application, a minimal deviation is allowed; however, this algorithm can impact the energy efficiency of the XSN platform. On the contrary, many applications, i.e., temperature, humidity, etc., might produce similar values for extended periods, and employing such an approach will significantly increase WSN lifetime.

4.1.1 Problem Description

Let represent WSN as a set of XSN platforms N_i where $i = 0, 1, ..n$, represented by a unique ID, and E that represent a set of connections. XSN platform measures the values over time and transmits them to the XSN node. Let T_1 represent the beginning time when the XSN platform N_i measure a value V_1 and transmit it to the end node. This way, we can present measurements at each time with their corresponding values as data points (T_i, V_i), $i = 0, ...n$ and easily find polynomials passing these data points as a system of linear equations. As soon as the coefficients are known from the equation, they can be delivered to the end nodes. The XSN node, at any time, can reconstruct the equation based on the receiving coefficients.

The XSN platform aims to collect a few data points and construct the interpolating polynomial function corresponding to the data points' degree. The higher the rate of changing values, the more complex the prediction of the following values becomes. The slower the changing values one after the other, the better the prediction. Although, we reconstruct

the polynomial as soon as we catch changes that do not correspond to the equation line. Another benefit of using such an equation is that we can construct the values at specified points (say, a user requires temperature at a specific time).

4.2 Data Prediction

As WSN applications are increasing fast, sensor devices' volume of generated data that need transmission from node to the endpoint is exceptionally high. Sensors observe different kinds of data, starting from environmental conditions (temperature, humidity, air pressure), monitoring object movements (patients, traffic, enemy), patients' conditions (temperature, heart rate, blood pressure), etc. All these will significantly impact memory and, specifically, battery power consumption for the XSN platforms. This section presents an approach to improve energy efficiency based on the Newton interpolation polynomial.

4.2.1 Newton Interpolation Polynomial Model

Let t_{N_0} be the observation in time N_0; meanwhile, y_{N_0} is the temperature corresponding to that time. Then, we have, $t_1 \leq t_2 \leq ... \leq t_{N_0}$, while y_i does not necessary have to be ordered. We form the sorted pairs $(t_1, y_1), , ..., (t_{N_0}, y_{N_0})$ and we can present in the coordinate system. Let us consider the problem: How to find function $y = f(t)$, which passes through some of these points and is closer to the other points. Then through this polynomial, we can make the prediction outside this area. In the beginning, we are dealing with the solution to the first problem.
If a function of time is fitted by a polynomial of time, the error between them is

$$f(t) - \sum_{j=1}^{M} A_j e_j(t) \tag{4.1}$$

where $e_j(t) = \prod_{i=0}^{j-1}(t - t_i)$ and $e_0(t) = 1$ while $f(t)$ is the function, t is time, M is the highest order of the polynomial, and A_M is a coefficient of each term.
Let us revisit the quadratic polynomial interpolant formula

$$f_2(t) = A_0 + A_1(t - t_0) + A_2(t - t_0)(t - t_1)$$

where

$$\begin{aligned} A_0 &= f(t_0) \\ A_1 &= \frac{f(t_1) - f(t_0)}{t_1 - t_0} \\ A_2 &= \frac{\frac{f(t_2)-f(t_1)}{t_2-t_1} - \frac{f(t_1)-f(t_0)}{t_1-t_0}}{t_2 - t_0} \end{aligned}$$

Note that A_0, A_1, and A_2 are finite divided differences. A_0, A_1, and A_2 are the first, second, and third finite divided differences, respectively. We denote the first divided difference by

$$f[t_0] = f(t_0)$$

the second divided difference by

$$f[t_1, t_0] = \frac{f(t_1) - f(t_0)}{t - t_0}$$

and

$$f[t_2, t_1, t_0] = \frac{f[t_2, t_1] - f[t_1, t_0]}{t_2 - t_0} = \frac{\frac{f(t_2)-f(t_1)}{t_2-t_1} - \frac{f(t_1)-f(t_0)}{t-t_0}}{t_2 - t_0}$$

where $f[t_0]$, $f[t_1, t_0]$, and $f[t_2, t_1, t_0]$ are called bracketed functions of their variables enclosed in square brackets. Rewriting,

$$f_2(t) = f[t_0] + f[t_1, t_0](t - t_0) + f[t_2, t_1, t_0](t - t_0)(t - t_1)$$

Table 4.1: The calculation of divided differences can be expressed in the following table, called the divided-difference scheme:

t_0	$f(t_0)$			
		$f[t_0, t_1]$		
t_1	$f(t_1)$		$f[t_0, t_1, t_2]$	
		$f[t_1, t_2]$		$f[t_0, t_1, t_2, t_3]$
t_2	$f(t_2)$		$f[t_1, t_2, t_3]$	
		$f[t_2, t_3]$		$f[t_1, t_2, t_3, t_4]$
t_3	$f(t_3)$		$f[t_2, t_3, t_4]$	
		$f[t_3, t_4]$		
t_4	$f(t_4)$			

This leads us to write the general form of Newton's divided difference polynomial for $N_0 + 1$ data points, $(t_0, y_0), (t_1, y_1), \cdots, (t_{N_0}, y_{N_0})$, as

$$f_{N_0}(t) = A_0 + A_1(t - t_0) + A_2(t - t_0)(t - t_1) + \cdots + A_2(t - t_0)(t - t_1)...(t_{N_0} - t_{N_0-1})$$

$$
\begin{aligned}
&A_0 = f[t_0] \\
&A_1 = f[t_1, t_0] \\
&A_2 = f[t_2, t_1, t_0] \\
&\vdots \\
&A_{N_0-1} = f[t_{n-1}, ..., t_1, t_0] \\
&A_{N_0} = f[t_{N_0}, t_{N_0-1}, ..., t_1, t_0]
\end{aligned}
$$

where the definition of the m^{th} divided difference is

$$A_{N_0} = f[t_{N_0}, t_{N_0-1}, ..., t_0] = \frac{f[t_{N_0}, t_{N_0-1}, ..., t_1] - f[t_{N_0-1}, ..., t_0]}{t_{N_0} - t_0}.$$

More about the interpolating polynomials see [207].
The procedural description is given below :

$$
\begin{aligned}
&(1)\ (A_0, A_1, ..., A_{N_0}) \leftarrow (f_0, f_1, ..., f_{N_0}) \\
&(2)\ \textbf{for } k = 1 \textbf{ to } N_0 \\
&(3)\quad \textbf{for } j = N_0 \textbf{ downto } k \\
&(4)\qquad A_j \leftarrow \frac{A_{j-1} - A_j}{t_{j-k} - t_j}
\end{aligned}
$$

4.2.2 Algorithms

Finding coefficients of the interpolating polynomial (Newton Divided-Difference Table): Given k distinct interpolating points (the time when the measurement is done) $x_0, x_1, ...x_k$, and the sensor values at these points, $y_k = f(x_k)$. Then, the coefficients of the interpolating function are: $c_l = f[x_l]$ for $l = 0, 1, ..., n-1$.
INPUT: x, y
Output: Coefficient set C_i
1. SET n = length(y)
2. SET $C_i j = 0$
3. **for** $i = 1$ to n do
4. $C_i 0 = y_i$
5. **end for**
6. **for** $j = 1$ to n do
7. **for** i = 0 to n-j do
8. $C_i j = (c_{i+1,j-1} - C_{i,j-1})/x_{i+1} - x_i$
9. **end for**
10. **end for**
11. **return** C_i

Algorithm to evaluate the new data point. This stage aims to predict new values using newton interpolation using coefficients produced by algorithm 1.
INPUT: C - coefficients, X data points, X_{NEW} - evaluation point
Output: P - predicted value
1. SET n = length(y)
2. SET n = length(X) - 1
3. SET p = C[n]
4. **for** $k = 1$ to $n + 1$ do
5. $p = C[n - k] + (X_{NEW} - X[n - k]) * p$
6. **end for**
7. **return** p

Since the Newton interpolation at some points can diverge far from the values within the range, the aim is to put a regulator between the predicted value and the mean of the last two data points. The role of this regulator is not to construct new coefficients when there are no big changes in values. Thus, we have:

$$MEAN = (Y[n - 1] + Y[n - 2])/2$$

Then, if $|MEAN - P| > 1$ we set $P = MEAN$. Further, in the XSN platform, we compare the real value with the predicted value, whether they are within a given threshold. The threshold value depends on the application area and can be changed accordingly. Those applications that are very critical must have a low threshold value. Then, the algorithm follows: if $|Current\_value - P| > THRESHOLD$, we construct new coefficients; otherwise, we do not need to send any value to the XSN node since the predicted one is valid and within the allowed range.

4.3 Experimental Results

The experimental results are tested with actual temperature values gathered within 12 hours from the XSN platform, namely waspmote[1], which is an embedded device with the integrated temperature sensor. The data are sent to the XSN node (raspberry PI), saving the data set in a file. We run the algorithm against this data set to check the technique's effectiveness. The algorithm works as follows: Few latest values are always collected in the sensor platform, and based on these values, we built the coefficients of the Newton interpolation. The coefficients are transmitted to the XSN node. XSN platform does not transmit any coefficients until the absolute value between the measured and the predicted one does not exceed a given threshold. Once it exceeds, new coefficients from the latest values are generated and sent again to the XSN node. Since the XSN node has coefficients, it can generate values anytime. The algorithm predicts values within and outside the range. Thus, it exceeds the limitations of the proposed algorithm given in [208], which compares values within the range. Furthermore, our algorithm is compared with the technique proposed in [208].

[1]https://www.cooking-hacks.com/documentation/tutorials/waspmote.html

4.3.1 Data Prediction

The main difference between the algorithm running on the XSN platform and the XSN node is that the generation of coefficients runs on the platform level. The following metrics are as follows:

Table 4.2: Parameters and values

Prameter	Value
Year	21/10/01
Observed Condition	temperature
Collected Readings	506
Slot interval	1 minute

4.3.1.1 Effect of Netwon's Degree Variation and Threshold

Each sensor platform needs to few raw data points to find the coefficients of a polynomial. The performance and accuracy of data depend on Newton's degree d that we perform in our simulation. An appropriate polynomial degree can be selected depending on the application requirements and sensor resources. The greater the polynomial degree, the more computation is needed, but more accurate data can recover at the XSN node. Furthermore, a given threshold is helpful for the algorithm to reconstruct coefficients in case the predicted value exceeds the given threshold.

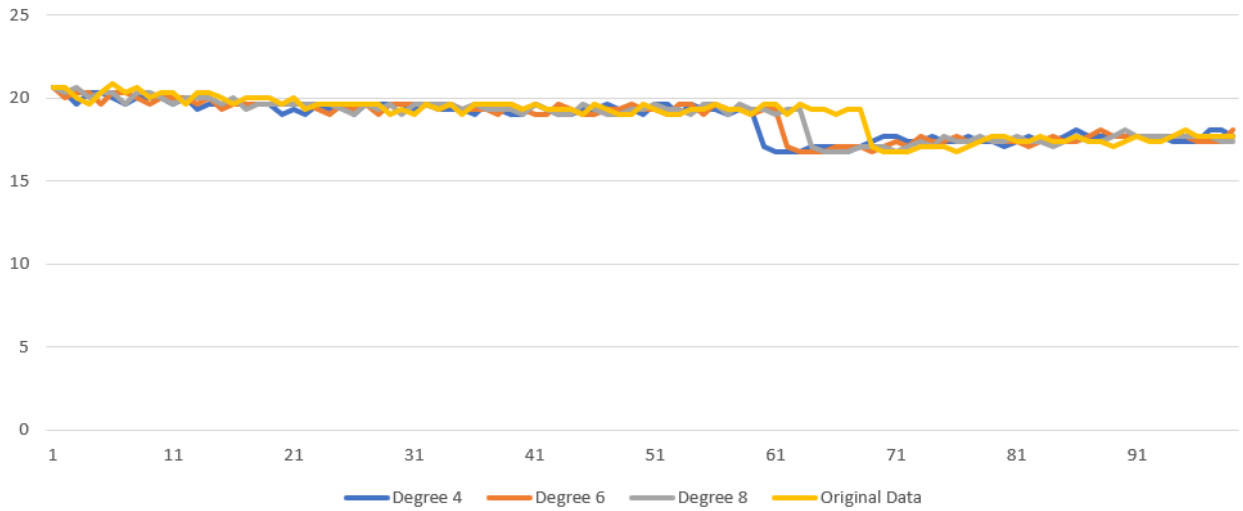


Figure 4.1: Newtons Degree Variation: $d = 4$, $d = 6$, $d = 8$ and threshold 0.1

In Figure 4.4, we predicted values within the range for a fixed size of 100 slots. As it is shown in the figure, the original data and the predicted values are not far away, specifically at

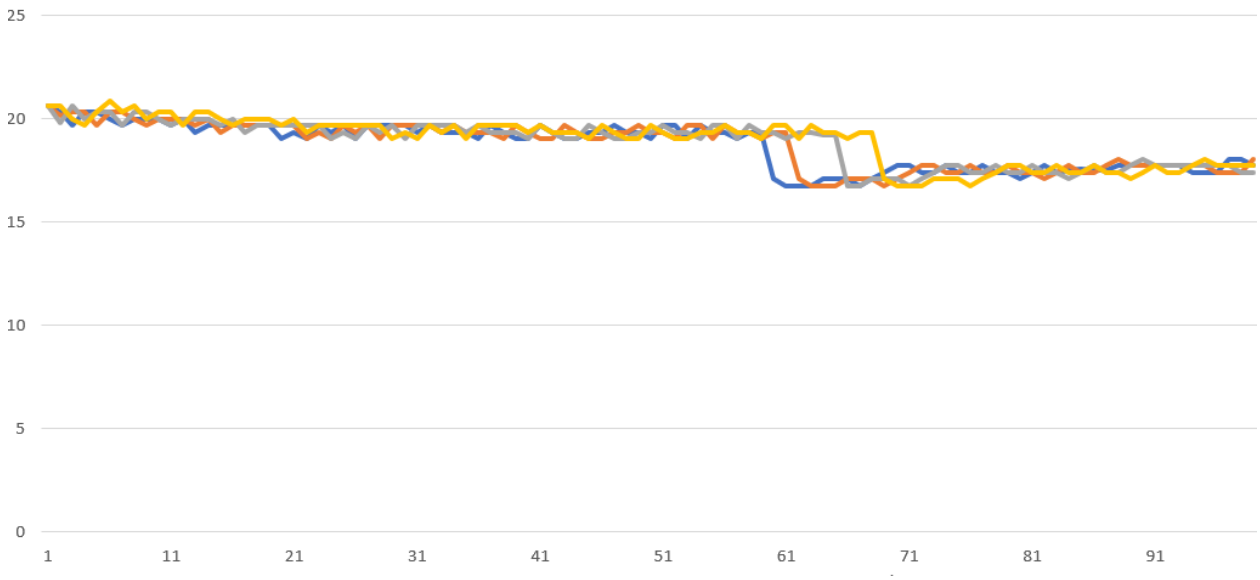


Figure 4.2: Newtons Degree Variation: $d = 4$, $d = 6$, $d = 8$ and threshold 0.25

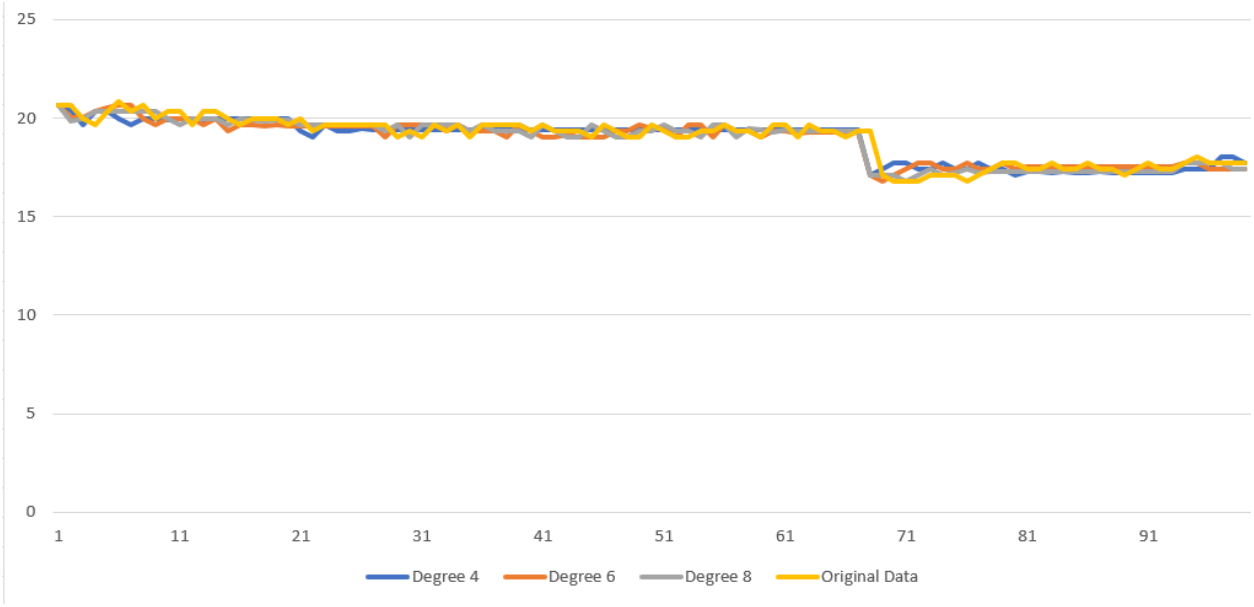


Figure 4.3: Newtons Degree Variation: $d = 4$, $d = 6$, $d = 8$ and threshold 0.5

degree 8 of the polynomial, where values match a lot. As a result, the algorithm reduces the transmission similar to [208] in the case of fixed periodic size and fixed degree. Furthermore, the algorithm exceeds the limitations of the technique in [208], for having a fixed period size. Our approach reconstructs new coefficients and re-evaluates the polynomial dynamically when there is no match between predicted and observed values in a given threshold.

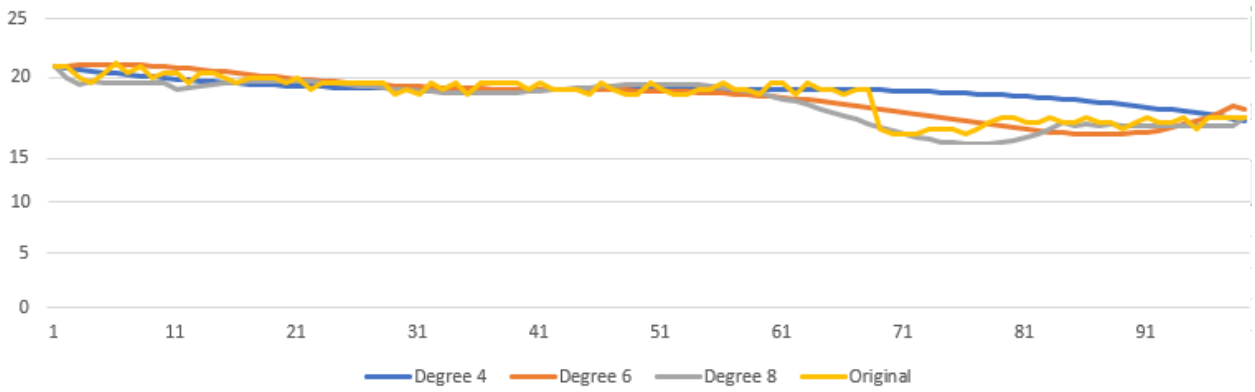


Figure 4.4: Predicting values within range in time slot of 100 values

4.3.1.2 Effect of Condition Variation

The results for the proposed approach reveal that if the current observation is quite different from the following observation, the algorithm requires more often to re-evaluate the coefficients, and the predicted value is slightly more pronounced than the original one. On the other hand, when the values vary slightly, the Newton data are much closer to real observed values, and there is no need to reconstruct the coefficients of Newton polynomials. As can be seen from Figure 4.5, having variations in values one after the other makes the real value and predicted ones have much difference compared to other cases.

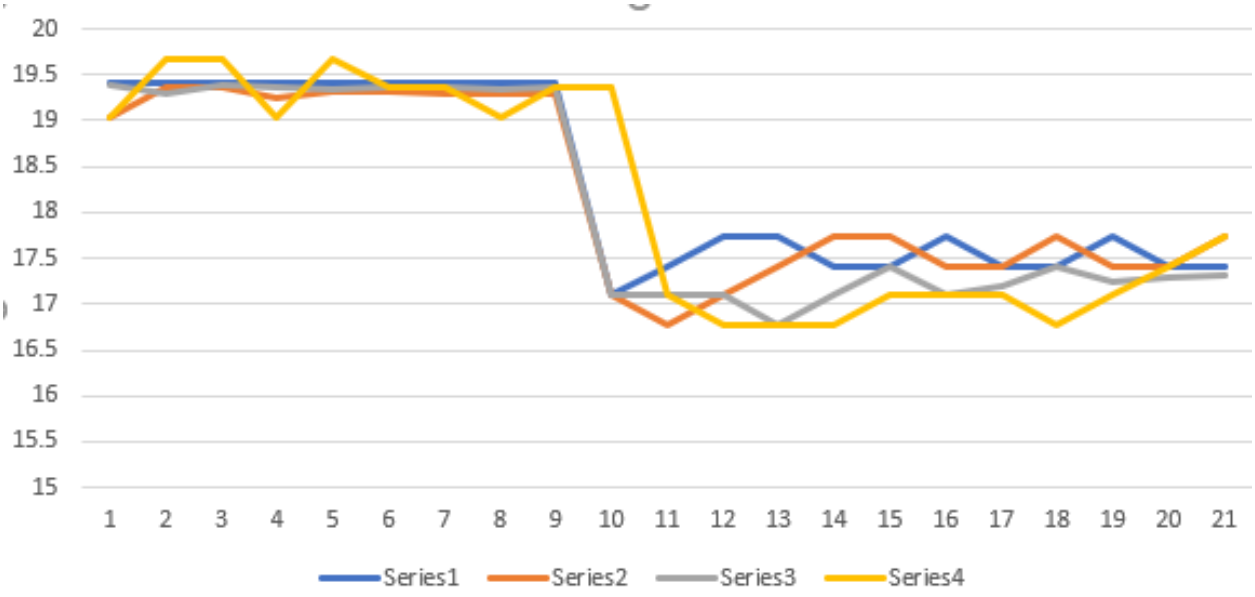


Figure 4.5: Effects of Condition Variation

4.3.1.3 Data Transmission and Energy consumption

This section analyzes the number of coefficients transmitted from each sensor in the XSN platform to the XSN node. The simulations show the algorithm can reduce transmissions between 92% - 94% when the threshold is set to 0.5, up to 45% when the threshold is set to 0.25, and up to 34% when the threshold is 0.1. The statistics are from the dynamic generation of coefficients when predicted and measured values exceed the threshold. Suppose we predict values within the given range in a fixed time slot. In that case, our technique behaves similarly to work proposed in [208] when it comes to periodic data transmission ratio for a fixed time slot.

We will use the following Equations presented in [152] to calculate the data reduction percentage.

$$T_{TR} = T_{CR} - N_{TR} \tag{4.2}$$

$$D_{PR} = \left| \left(\frac{T_{TR}}{T_{CR}} * 100 \right) - 100 \right| \tag{4.3}$$

where T_{TR} represent total transmitted readings, T_{CR} - represent the data set, N_{TR} - non transmitted readings. From Table 4.7, we see that the algorithm can save energy on data transmission from 34% to 94%. Regarding the periodic transmission, this all depends on the time slot. Suppose we transmit every half an hour; then the number of coefficient transmissions for 24h will be 48.

Table 4.3: Dynamic Data Transmission out of 350 data set

Threshold	0.5			0.25			0.1		
Newton degree	d4	d6	d8	d4	d6	d8	d4	d6	d8
Nr. of transmissions	24	28	20	192	216	220	231	231	231
Reduced Transm. in %	93.1	92	94.3	45.1	38.3	37.1	34	34	34
Periodic transmissions	depends on fixed time period								

In order to evaluate the energy consumption of the proposed model, we will use the energy model in [209]. This model depends highly on data transmission, and other factors like sensing and processing are not considered. The energy consumed by the XSN platform with N sensors is given using the equation

$$Energy = N * (d + 1) * E_c * 64 \tag{4.4}$$

where d represents the Newton degree (i.e., for degree 4, we have d+1 values), E_c is electronic circuity (mostly $50nJ/bit$)

Hence, our technique performs very well in improving the energy lifetime of XSN platforms/nodes when considering that most of the energy is spent during data transmission. Specifically, this is emphasized when having a border threshold of 0.5. Then, the more energy is spent, the more often we generate new coefficients that need to be delivered to the end node. Also, the energy lifetime decreases with the increase of Newton's degree. If we

Table 4.4: Energy Consumption

Threshold	0.5			0.25			0.1		
Newton degree	d4	d6	d8	d4	d6	d8	d4	d6	d8
Energy cons. in mJ	0.38	0.63	0.51	3.07	4.84	5.38	3.69	5.17	5.91
Energy saving in %	94	90	92	54	28	20	45	23	13

compare our technique with the technique proposed [208], even though it dynamically generates coefficients performs better when we provide a threshold of 0.5. The authors in [208] can achieve 86% - 93% for Lagrange degree 6, while we can achieve 90%-94% by dynamically generating coefficients. However, our technique does not perform better when reducing the threshold to 0.1. This performance decrease is due to dynamic generation and frequent data transmission. However, having a concise period time slot, our technique might achieve the same results as the research results provided [208].

4.3.1.4 Data Accuracy

Data accuracy refers to the similarity between the predicted values and observed values. To evaluate the proposed approach's data accuracy, we will consider the Mean Square Error (MSE) and Mean Absolute Error (MAE). Both are useful for checking the quality (accuracy) of an estimator. The formula for MSE and MAE are as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(AV - PV)^2$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|AV - PV|$$

where AV represents observed values from the data set and PV represents predicted values. Then, the Tables below show different Newton degrees' overall performance with different thresholds. All the tables show that the algorithm performs relatively well regarding accuracy. The higher the degree of Newton's polynomial, the more accurate the prediction. On the other hand, the data quality degrades when we lower the threshold value. Thus, if we consider data accuracy, it is beneficial to set the threshold higher to have better accuracy for data sets that do not have more considerable changes between time slots.

A common metric to check in percentage the accuracy of a model is Mean Absolute Percentage Error (MAPE) which is given below:

$$MAPE = \frac{100\%}{n} * \sum_{i=0}^{n}\left(\frac{|actual – forecast|}{|actual|}\right)$$

According to Tables 4.8, we can easily see that our approach achieves highly accurate data with error percentages lower than 2%. That means we achieve an accuracy of about 98%. Note that the algorithm did not consider any erroneous value or loss, which may affect the data accuracy.

Table 4.5: MSE and MSA for given threshold 0.5 in the proposed approach

Threshold	0.5		
Newton degree	d4	d6	d8
MSE	0.112818144	0.103024362	0.081035668
MAE	0.253929079	0.223464962	0.22946131

Table 4.6: MSE and MSA for given threshold 0.25 in the proposed approach

Threshold	0.25		
Newton degree	d4	d6	d8
MSE	0.298122177	0.259323889	0.207310902
MAE	0.341373224	0.312772419	0.307370295

Table 4.7: MSE and MSA for given threshold 0.1 in the proposed approach

Threshold	0.1		
Newton degree	d4	d6	d8
MSE	0.220642	0.208681613	0.207784315
MAE	0.313059853	0.296422767	0.310286971

Table 4.8: Mean Absolute Percentage Error

Threshold	0.5			0.25			0.1		
Newton degree	d4	d6	d8	d4	d6	d8	d4	d6	d8
Error (%)	1.39	1.22	1.22	1.85	1.69	1.67	1.70	1.61	1.62

4.3.1.5 Processing Time

Due to energy, memory, or processing limitations, complexity is another important metric to be considered in XSN nodes and XSN platforms. Processing is a metric that can delay data transmission to the end node. In our algorithm, the complexity of processing the coefficients depends on the polynomial degree, which means the number of readings to evaluate the function. The complexity of our algorithm is $\frac{n*(n-1)}{2}$ that is $O(n^2)$, where $n = degree$. The complexity is similar to the Lagrange polynomial proposed in [208]. One of the advantages of using Newton's polynomial is adding more data points without re-constructing the whole problem. Regarding the processing time, we need $O(d+1)$ in order to evaluate the function. Of course, the complexity increases with the increase of Newton's degree. This complexity is due to more computations involved in reading and processing data to find the coefficients. Even though processing impacts energy, the approach consists of simple processing that will

not impact that much.

4.4 Conclusion and Future Work

The number of sensory devices is increasing daily, and IoT is becoming mainstream for future applications. More data will be generated as more sensory devices are used in everyday applications. It is much needed to advance current strategies either in data management or specifically in saving energy from battery-powered sensory devices. Thus, in the current work, we have proposed a dual data prediction algorithm that runs on two different nodes, namely the XSN platform and XSN Node, to save energy by reducing data transmission between such nodes. The first node will generate coefficients, while the second node can predict the values based on the generated coefficients. Through simulations of the temperature data set, we demonstrate that the technique exceeds some limitations of having a fixed time slot for other research. It can dynamically predict values in-between and outside the range. Also, the technique shows that it can improve energy consumption and thus extend network lifetime and have better energy saving on some specifically given threshold. Another critical point is that this technique can produce highly accurate data.

However, for future work, some improvements can be made. First, we need to check how the algorithm can behave when we consider erroneous data and improve in that direction, so the predicted values do not reduce data quality. Another approach is to advance the algorithm more, so we can set it in sleep mode on the sensor platform, which can save the XSN platform's energy even more. In this direction, we can combine our algorithm with the Fourier series to predict future values without reconstructing coefficients. As a result, we can reduce data transmission even more without breaking data accuracy.

Chapter 5

Energy Efficient Data Reduction and Prediction Based on DST-I and DST-II Least-squares Extended model

Energy efficiency in WSNs is a fundamental problem for many research studies. Wireless Sensor Networks often consists of many small, reasonable sensors with restricted power, processing, and memory resources. The sensory devices can measure and gather information from the monitoring environment and transmit the raw sensory data to the base station. Since battery-powered sensory devices often operate in harsh environments in battery-enabled mode, they must secure power as much as possible to hold their connections in WSN networks. Therefore, saving energy in sensory devices demands efficient prediction techniques to reduce data transmission between the nodes. It will help to avoid waste of energy during transmission.

This chapter proposes two novel forecasting models, Discrete Sine Transform (DST-I and DST-II) Least-Squares Extended (LSE) model, which are Fourier-related transforms using a purely real matrix. In the proposed models, DST-I and DST-II are used to generate coefficients of the function based on a finite number of discrete data points collected from sensor readings of a sensor node. The same algorithms can be used on the sink node to recover the data and predict future values. By using the proposed techniques, we decrease the number of transmissions from the sensor node to the sink node, which can significantly improve energy efficiency and extend the sensor network lifetime. The proposed models are evaluated using real sensor data collected from Chickago Beach district from three locations. MSE, RMSE, MAE, and MAPE are the metrics for performance evaluation. The evaluation shows highly accurate data and energy improvements. The performance results clearly show that the proposed models outperform similar techniques regarding prediction accuracy, such as DNN + Rough sets and DCT. Furthermore, the results of the two proposed techniques (DST-I and DST-II) show that DST-II performs better than DST-I in terms of accuracy.

The chapter is structured as follows: Section 5.1 is the problem description of DST-I, followed by the next section describing the combination of DST-I with the Least Square Extended (LSE) model used for coefficient findings. Then, the next two sections describe

DST-II and the connection of DST-II with LSE. Then we describe the steps of the algorithmic procedure, followed by algorithms. Section 5.7 elaborates the results of the algorithms performed on the real sensor data set. Finally, a discussion completes the chapter.

5.1 Problem Description of the DST- I based Forecast Modelling

In maths, the discrete sine transform (DST-I) belongs to Fourier transforms similar to the DFT using a purely real matrix. It is equivalent to the imaginary parts of a DFT of approximately twice the length, operating on real data with odd symmetry, where the input/output data in some variants is shifted by half a sample. DST-I is a critical technique or method to convert a signal into an elementary frequency component. It is commonly used in image compression techniques similar to JPEG compression, which converts each pixel intensity of an image into its corresponding frequency value. This chapter aims to study DST's efficacy in predicting WSN data, so we can use the built coefficients to avoid unnecessary data transmissions between the nodes. It is imperative to mention that there has been no work in using DST on WSNs for data reduction and prediction. Hence, there is no work on using DST in forecasting water temperature.

For a sequence (discrete time-series) of length T, its 1-D (one-dimension) DST-I definition is,

$$\mathcal{H}(s) = \sum_{t=0}^{T-1} \mathcal{G}(t) \sin\left((t+1)\theta_s \pi\right), \quad s = 0, 1, ..., T-1,$$

where $\mathcal{G}$ represents the sequence (or called discrete time-series) of length T $(\mathcal{G}(t), t = 0, 1, ..., T-1)$, $\mathcal{H}$ is the DST-I of $\mathcal{G}$ $(\mathcal{H}(s), s = 1, 2, ..., T)$ and $\theta_s = \frac{(s+1)\pi}{T+1}$. The DST-I of the T-point input is given by

$$\mathcal{G}(t) = \sum_{s=0}^{R-1} \sqrt{\frac{2}{R+1}} \mathcal{H}(s) \sin\left(\frac{\pi(s+1)(t+1)}{R+1}\right), \quad s = 0, 1, ..., T-1.$$

5.2 The Least-Squares Solution for the DST-I Coefficients-Seeking

In this section, we try to solve the problem of determining the DST-I coefficient for prediction modeling by employing the LS (least-squares) estimates. That is, on the basis of the finite observations of water temperature for an hour $\mathcal{G}(t)(t = 1, 2 \ldots T_0)(T_0 < T)$ to determine the called LSO (least-squares optimum) $R-$term DST-I coefficients: $\hat{\mathcal{H}}(s)(s = 1, 2 \ldots R)$, a called the DST-I-LS-extended (DST-I-based least-squares-extended) prediction model is proposed as follows,

$$\hat{\mathcal{G}}(t) = \sum_{s=1}^{R} \hat{\mathcal{H}}(s) \sin\left(\frac{s\pi t}{T+1}\right) \sqrt{\frac{2}{R+1}}; \tag{5.1}$$

where, the DST-I-LS-extended model defined by equation (5.1)) will be determined by the model's outputs $\hat{\mathcal{G}}(t)$ which fit the previous observations $\mathcal{G}(t)(t = 1, 2 \ldots T_0)(T_0 < T)$ in the LS (least-squares) sense. Note that with consideration of the effective solution for getting the R-term DST-I coefficients $\hat{\mathcal{H}}(s)$ $(s = 1, 2, \ldots, R)$ on the basis of observation of the finite water temperature for an hour $T_0(T_0 < T)$, we usually put $R \leq T_0 < T$. Denote:

$$\begin{aligned} \mathcal{E}^2 &= \sum_{t=1}^{T_0} \left(\mathcal{G}(t) - \hat{\mathcal{G}}(t) \right)^2 \\ &= \sum_{t=1}^{T_0} \left[\mathcal{G}(t) - \sum_{s=1}^{R} \hat{\mathcal{H}}(s) \sqrt{\frac{2}{R+1}} \sin\left(\frac{s\pi t}{T+1} \right) \right]^2 . \end{aligned} \tag{5.2}$$

The called LSO (least-squares-optimum) solution for the DST-I-LS-extended prediction model defined by the equation (5.1) indicates that the sum of squares of the errors $\mathcal{E}^2$ defined by the equation (5.2) will be the least amount possible. To minimize $\mathcal{E}^2$ (Eq.(5.2)) with respect to $\hat{\mathcal{H}}(s)$, $\left(\frac{\partial \mathcal{E}^2}{\partial \hat{\mathcal{H}}(s)} = 0,\ s = 1, 2, ..., R \right)$ it follows,

$$\sum_{t=1}^{T_0} \sin\left(\frac{s\pi t}{T+1} \right) \sqrt{\frac{2}{R+1}} \left[\mathcal{G}(t) - \sum_{s=1}^{R} \hat{\mathcal{H}}(s) \sqrt{\frac{2}{R+1}} \sin\left(\frac{s\pi t}{T+1} \right) \right] = 0 \tag{5.3}$$

from here we have

$$\begin{aligned} \sum_{t=1}^{T_0} \sin\left(\frac{s\pi t}{T+1} \right) \sqrt{\frac{2}{R+1}} \mathcal{G}(t) &= \sum_{t=1}^{T_0} \sin\left(\frac{s\pi t}{T+1} \right) \sqrt{\frac{2}{R+1}} \\ &\cdot \sum_{s=1}^{R} \hat{\mathcal{H}}(s) \sqrt{\frac{2}{R+1}} \sin\left(\frac{s\pi t}{T+1} \right) . \end{aligned} \tag{5.4}$$

Let $a(r, t) = \sin\left(\frac{r\pi t}{T+1} \right) \sqrt{\frac{2}{R+1}}$, it results in,

$$\sum_{t=1}^{T_0} a(r, t) \mathcal{G}(t) = \sum_{t=1}^{T_0} a(r, t) \sum_{s=1}^{R} \hat{\mathcal{H}}(s) a(s, t), \quad \text{for } r = 1, 2, ..., R. \tag{5.5}$$

For brevity of mathematical expression, further definitions are introduced as,

$$A_{R \times T_0} = \begin{bmatrix} a(1,1) & a(1,2) & \cdots & a(1,T_0) \\ a(2,1) & a(2,2) & \cdots & a(2,T_0) \\ \vdots & \vdots & \ddots & \vdots \\ a(R,1) & a(R,2) & \cdots & a(R,T_0) \end{bmatrix}$$

and

$$\mathcal{G}_{T_0\times 1}=\begin{bmatrix} G(1)\\ G(2)\\ \vdots\\ G(T_0)\end{bmatrix}\quad \hat{\mathcal{H}}_{R\times 1}=\begin{bmatrix} \hat{\mathcal{H}}(1)\\ \hat{\mathcal{H}}(2)\\ \vdots\\ \hat{\mathcal{H}}(R)\end{bmatrix}$$

From here and from equation (6.7) it follows

$$A_{R\times T_0}\cdot\mathcal{G}_{T_0\times 1}=A_{T\times T_0}\left(\hat{\mathcal{H}}'_{R\times 1}A_{R\times T_0}\right)'=A_{R\times T_0}\cdot A'_{T_0\times R}\cdot\hat{\mathcal{H}}_{R\times 1}\tag{5.6}$$

Note that for $r=1,2\ldots R$, we have

$$\sum_{t=1}^{T_0}a(r,t)\sum_{s=1}^{R}\hat{\mathcal{H}}(s)a(s,t)=A_{R\times T_0}\left(\hat{\mathcal{H}}'_{R\times 1}A_{R\times T_0}\right)'\tag{5.7}$$

From the equation (5.6), on the basis of the prior finite T_0 observations of water temperature for hour $\mathcal{G}(t)(t=1,2,...,T_0)(T_0<T)$, the called $R-$term LSO (least-squares-optimum) DST-I coeficinets $\hat{\mathcal{H}}(s)(s=1,2,...,R)$ for building the DST-I-LS-extended prediction model defined by (5.1) are determined by,

$$\hat{\mathcal{H}}_{R\times 1}=\left(A_{R\times T_0}A'_{T_0\times R}\right)^{-1}A_{R\times T_0}\mathcal{G}_{T_0\times 1}\tag{5.8}$$

Finally, the DST-I-LS-extended prediction model defined by relation (5.1) can now be established with the LSO (least-squares-optimum) DST-I coefficients $\hat{\mathcal{H}}_{R\times 1}$ defined by (5.8) available. Then we can employ the DST-I-LS-extended prediction model defined by the equation (5.8) to predict the future fluctuation of the water temperature for an hour at its next point of time: $\hat{\mathcal{G}}(t)(t=T_0+1)$ or ones at its subsequent points in time: $\hat{\mathcal{G}}(t)(t=T_0+1,\ldots,T)$.

5.3 Problem Description of the DST-II- Based Forecast Modeling

For a discrete-time signal (also called a sequence) with T components, its 1-D (one-dimension) DST-II definition is,

$$Q(s)=\vartheta(s)\sum_{t=1}^{T}\mathcal{P}(t)\sin\left(\frac{(2t-1)\theta_s}{2}\right),\quad s=1,2,...,T,\tag{5.9}$$

where $\mathcal{P}$ represents the sequence (or called discrete time-series) of length T $(\mathcal{P}(t),t=1,2....,T)$, Q is the DST-I of $\mathcal{P}$ $(\mathcal{P}(s),s=1,2,...,T)$ and

$$\vartheta(s)=\sqrt{\frac{2}{T}}\psi_s$$

$$\psi_s := \begin{cases} \sqrt{\frac{1}{2}}, & \text{if } s = T, \\ 1, & \text{if } s = 1, 2, ..., T-1. \end{cases}$$

and

$$\theta_s = \frac{s\pi}{T}.$$

Its corresponding inverse sine transform called the inverse DST-II (IDST-II) is defined by,

$$\mathcal{P}(t) = \sum_{s=1}^{T} Q(s)w(s)\sin\left(\frac{(2t-1)s\pi}{2T}\right), \quad s = 1, 2, ..., T, \tag{5.10}$$

where

$$\omega(s) = \sqrt{\frac{2}{T}}\psi_s$$

$$\psi_s := \begin{cases} \sqrt{\frac{1}{T}}, & \text{if } s = T, \\ \sqrt{\frac{2}{T}}, & \text{if } s = 1, 2, ..., T-1. \end{cases}$$

5.4 The Least-Squares Solution for the DST-II Coefficients-Seeking

From the previous section, we see that the definitions of DST-II and IDST-II (relations (5.9)-(5.10)) show that the discrete time-series $\mathcal{P}(t)(t = 1, 2, \ldots, T)$ and the DST-II transform coefficients $\mathcal{Q}(s)(s = 1, 2, \ldots, T)$ are the same size of T-length. Given the limited number of T_0 $(< T)$ daily water temperature data over a given period of time: $\mathcal{P}(t)(t = 1, 2, \ldots, T_0)$, is needed how to extend the presented model of the DST-II defined by relation (5.10) to the prediction to prediction the daily water temperature movement at its next time points: $\mathcal{P}(t)(t = T_0 + 1, ..., T)$. How to calculate the Discrete Sine Transform coefficients $\mathcal{Q}(t)$ (relation (5.9)-(5.10)) based on the limited hourly observations is the principal problem. However, since $T_0 < T$, we can not directly calculate the Discrete Sine Transform coefficients $\mathcal{Q}(s)$ by using relation (5.9). The least-squares approach can solve the problem. Namely, on the basis of the finite observations: $\mathcal{P}(t)(t = 1, 2, \ldots, T_0)(T_0 < T)$, we are to get its R-term Discrete Sine Transform coefficients $\hat{\mathcal{Q}}(s)(s = 1, 2, \ldots, R)$ for building the extended model defined by relation (5.11), which yields $\hat{\mathcal{P}}(t)$ most fitting the given $\mathcal{P}(t)(t = 1, 2, \ldots, T_0)(T_0 < T)$ in the least-squares sense,

$$\hat{\mathcal{P}}(t) = \sum_{s=1}^{R} \hat{Q}(s)w(s)\sin\left(\frac{s\pi(2t-1)}{2T}\right). \tag{5.11}$$

where

$$\omega(s) := \begin{cases} \sqrt{\frac{1}{R}}, & \text{if } s = R, \\ \sqrt{\frac{2}{R}}, & \text{if } s = 1, 2, ..., R-1. \end{cases}$$

It must be borne in mind that to effectively obtain the R-term coefficients $\hat{\mathcal{Q}}(s)(s = 1, 2, \ldots, R)$ based on the limited $T_0 < T$ observations, typically, we set $R \leq T_0 < T$. Denote $\mathcal{E}^2$ as sum of squares of the errors (or called the deviations) $(\epsilon_t, t = 1, 2 \ldots T_0)$,

$$\begin{aligned} \mathcal{E}^2 &= \sum_{t=1}^{T_0} e_t^2 = \sum_{t=1}^{T_0} \left(\mathcal{P}(t) - \hat{\mathcal{P}}(t) \right)^2 \\ &= \sum_{t=1}^{T_0} \left[\mathcal{P}(t) - \sum_{s=1}^{R} \omega(s) \hat{Q}(s) \sin\left(\frac{s\pi(2t-1)}{2T} \right) \right]^2 ; s = 1, 2, .., R. \end{aligned} \tag{5.12}$$

Now we are to find its optimal solution, which subjects to that sum $\mathcal{E}^2$ of squared deviations or called residuals e_t $t = 1, \cdots, T_0$ (relation (5.12)) will be the least amount possible. To minimize $\mathcal{E}^2$ (relation (5.12)) with respect to $\hat{Q}(k)$, $\left(\frac{\partial \mathcal{E}^2}{\partial \hat{Q}(s)} = 0,\ s = 1, 2, ..., R \right)$ it follows,

$$\sum_{t=1}^{T_0} \omega(s) \hat{Q}(s) \sin\left(\frac{s\pi(2t-1)}{2T} \right) \left[\mathcal{P}(t) - \sum_{s=1}^{R} \hat{Q}(s) \omega(s) \sin\left(\frac{s\pi(2t-1)}{2T} \right) \right] = 0$$

from here we have

$$\begin{aligned} \sum_{t=1}^{T_0} \omega(s) \sin\left(\frac{s\pi(2t-1)}{2T} \right) \mathcal{P}(t) &= \sum_{t=1}^{T_0} \omega(s) \sin\left(\frac{s\pi(2t-1)}{2T} \right) \\ &\times \sum_{s=1}^{R} \hat{Q}(s) \omega(k) \sin\left(\frac{s\pi(2t-1)}{2T} \right). \end{aligned}$$

Let $l(r, t) = \omega(r) \sin\left(\frac{r\pi(2t-1)}{2T} \right)$, it results in,

$$\sum_{t=1}^{T_0} l(r, t) \mathcal{P}(t) = \sum_{t=1}^{T_0} l(r, t) \sum_{s=1}^{R} \hat{Q}(s) \omega(s) \sin\left(\frac{k\pi(2t-1)}{2T} \right), \quad \text{for } r = 1, 2, ..., R. \tag{5.13}$$

For brevity of mathematical expression, further definitions are introduced as,

$$L_{R \times T_0} = \begin{bmatrix} l(1,1) & l(1,2) & \cdots & l(1,T_0) \\ l(2,1) & l(2,2) & \cdots & l(2,T_0) \\ \vdots & \vdots & \ddots & \vdots \\ l(R,1) & l(R,2) & \cdots & l(R,T_0) \end{bmatrix}$$

and

$$\mathcal{P}_{T_0\times 1}=\begin{bmatrix}\mathcal{P}(1)\\ \mathcal{P}(2)\\ \vdots\\ \mathcal{P}(T_0)\end{bmatrix}\quad \hat{Q}_{R\times 1}=\begin{bmatrix}\hat{\mathcal{Q}}(1)\\ \hat{\mathcal{Q}}(2)\\ \vdots\\ \hat{\mathcal{Q}}(R)\end{bmatrix}$$

From here and from equation (5.13) it follows

$$L_{R\times T_0}\cdot\mathcal{P}_{T_0\times 1}=L_{R\times T_0}\cdot L'_{T_0\times R}\cdot\hat{Q}_{R\times 1} \tag{5.14}$$

Note that for $r=1,2,\ldots,R$, we have

$$\sum_{t=1}^{T_0}l(r,t)\sum_{s=1}^{R}\hat{Q}(s)l(r,t)=L_{R\times T_0}\left(\hat{\mathcal{P}}'_{R\times 1}L_{R\times T_0}\right)' \tag{5.15}$$

From the equation of relation (5.14), on the basis of the prior finite T_0 for observations water temperature for hour $\mathcal{P}(t)(t=1,2,...,T_0)(T_0<T)$, the called $R-$term LSO (least-squares-optimum) DST-II coeficinets $\hat{Q}(t)(t=1,2,...,R)$ for building the DST-II-LS-extended prediction model defined by relation (5.11) are determined by,

$$\hat{Q}_{R\times 1}=\left(L_{R\times T_0}L'_{T_0\times R}\right)^{-1}L_{R\times T_0}\mathcal{P}_{T_0\times 1} \tag{5.16}$$

Finally, the DST-II-LS-extended prediction model defined by relation (5.11) can now be established with the LSO (least-squares-optimum) DST-II coefficients $\hat{Q}_{R\times 1}$ defined by relation (5.16) available. Then we can employ the DST-II-LS-extended prediction model defined by relation (5.11) to predict the future fluctuation of the water temperature for an hour at its next point of time: $\hat{\mathcal{P}}(t)(t=T_0+1)$ or ones at its subsequent points in time: $\hat{\mathcal{P}}(t)(t=T_0+1,\ldots,T)$.

5.5 Procedure for application of DST-I-LS (DST-II-LS)-Extended Forecast Model

To elaborate, the algorithmic procedure of the proposed DST-I-LS (DST-II-LS)-extended forecast model (Eq.(5.1) and (5.11)), a procedure of analysis is composed of the following steps:

(1) To evaluate annual water temperature by month, we assume its discrete-time period T, as T = 12 for it has 12 months in each day, and similarly, we put T = 24 for daily water temperature by an hour, that is, 24 h of each day. Note that we can also set T = 48 for half-hourly air temperature forecasting as there are 48 half-hours in one day, and so on.

(2) On the basis of previous daily water temperature for hour observations: $\hat{\mathcal{G}}(t)(t=$

$1, 2 \ldots T_0)(T_0 < T)$, $(\hat{\mathcal{P}}(t)(t = 1, 2 \ldots T_0)(T_0 < T)$, we are to implement the DST-I-LS(DST-II-LS)-extended forecast modeling (Eq. (5.1)) to predict its future fluctuation at its next point of time $\hat{\mathcal{P}}(t)(t = T_0 + 1)(\hat{\mathcal{G}}(t)(t = T_0 + 1))$ or ones at its subsequent points in time: $\hat{\mathcal{P}}(t)(t = T_0 + 1, \ldots, T)(\hat{\mathcal{G}}(t)(t = T_0 + 1, \ldots, T))$.

(3) To determine the so-called M-term LSO (least-squares-optimum) DST-I (DST-II) coefficients: $\hat{\mathcal{H}}(s)(s = 1, 2, \ldots, R)$ $(\hat{\mathcal{Q}}(t)(t = 1, 2, \ldots, R))$ for forecast modeling (Eq.(5.1)) based on the previous for daily water temperature by hour observations: $\hat{\mathcal{G}}(t)(t = 1, 2, ..., T_0), (\hat{\mathcal{P}}(t), (t = 1, 2, ..., T_0))$ by

$$\hat{\mathcal{H}}_{R\times 1} = \left(B_{R\times T_0} B'_{T_0\times M}\right)^{-1} B_{R\times T_0} x_{T_0\times 1}; (\hat{Q}_{R\times 1} = \left(L_{R\times T_0} L'_{T_0\times R}\right)^{-1} L_{R\times T_0} \mathcal{P}_{T_0\times 1});$$

(4) With the DST-I-LS (DST-II)- coefficients $\hat{\mathcal{H}}(s)(s = 1, 2, \ldots, R)$; $\hat{\mathcal{Q}}(s)(s = 1, 2, \ldots, R)$ available, future daily water temperature hourly at its next point time $\hat{\mathcal{G}(t)}(t = T_0 + i), i = 1, 2, ..., T - T_0$, $(\hat{\mathcal{P}(t)}(t = T_0 + i), i = 1, 2, ..., T - T_0)$ or ones at its subsequent points in time $(t = T_0 + 1, \ldots, T)$ can be predicted by the DST-I(DST-II)-LS-extended model:

$$\hat{\mathcal{G}}(t) = \sum_{s=1}^{R} \hat{\mathcal{H}}(s) \sqrt{\frac{2}{R+1}} \sin\left(\frac{s\pi t}{T+1}\right), \left(\hat{\mathcal{P}}(t) = \sum_{s=1}^{R} \hat{Q}(s) w(s) \sin\left(\frac{s\pi(2t-1)}{2T}\right)\right).$$

5.6 Algorithms

Finding coefficients of the DST are as following:

Given the limited number of the sensor values T_0 $(< T)$ of daily water temperature data over a given period of time, described using the model $\hat{\mathcal{G}}(t)(t = 1, 2, \ldots, T_0)$, $(\hat{\mathcal{P}}(t)(t = 1, 2, \ldots, T_0))$, it is required to extend the presented model of the DST-I (DST-II) defined by relation (5.1, respectively 5.10) to predict daily water temperature movement at future points: $\mathcal{G}(t)(t = T_0 + 1, ..., T)$, $(\hat{\mathcal{P}}(t)(t = T_0 + 1, ..., T))$.

5.6.1 DST-I

INPUT: x, R, T, T_0
Output: Coefficient set C_i
1. SET B = $\{\}_{RXT_0}$;
2. **for** n = 1 to T_0 do
3. **for** $m = 1$ to R do
4. $B_{m,n} = sqrt(2/(R+1)) * sin((m * \pi * n)/(T+1))$;
end for
5. **end for**
6. SET $C_i = inv(B * B') * B * x$;
11. **return** C_i

Algorithm to evaluate the new data point. This stage aims to predict new values using coefficients from DST produced by algorithm 1.
INPUT: C - coefficients, R, T, T_0
Output: S - predicted value
1. SET S = $\{\}_{T_0+1X1}$;
2. **for** n = 1 to $T_0 + 1$ do
3. **for** $k = 1$ to R do
4. $S_n = S_n + sqrt(2/(R+1)) * sin((k * \pi * n)/(T+1)) * C_k$;
5. **end for**
6. **end for**
7. **return** S

5.6.2 DST-II

INPUT: x, R, T, T_0
Output: Coefficient set C_i
1. SET B = $\{\}_{RXT_0}$;
2. **for** n = 1 to T_0 do
3. **for** $m = 1$ to R do
4. $B_{m,n} = sqrt(2/(R+1)) * sin((m * \pi * n)/(T+1))$;
end for
5. **end for**
6. **for** n = 1 to T_0 do
7. B(R,n)=sqrt(1/R)*sin((R*pi*(2*n-1))/(2*T));
8. SET $C_i = inv(B * B') * B * x$;
9. **end for**
10. **return** C_i

Algorithm to evaluate the new data point. This stage aims to predict new values using coefficients from DST-II produced by algorithm above.
INPUT: C - coefficients, R, T, T_0
Output: S - predicted value
1. SET S = $\{\}_{T_0+1X1}$;
2. **for** n = 1 to $T_0 + 1$ do
3. **for** $k = 1$ to R do
4. $S_n = S_n + sqrt(2/(R+1)) * sin((k * \pi * n)/(T+1)) * C_k$;
5. **end for**
6. **for** $n = 1$ to $T_0 + 1$ do
7. $S(i) = S(i) + P(R) * sqrt(1/R) * sin((R * pi * (2 * i - 1))/(2 * T))$;
8. **end for**
9. **return** S

5.7 Experimental Results

To validate the proposed approach under real use case WSN deployment conditions, we use the actual data set from automated sensors of the Chicago Park District Beach Weather Stations [210]. Unlike the dataset used in the previous chapter, we aim to evaluate the proposed approach from the same sensors in different locations. The sensors dataset is from locations in the Chicago district that observe air temperature, water temperature, humidity, wind, etc., on an hourly basis. All the sensors measure values on an hourly basis. We evaluate our approach using water temperature from sensors located at Calumet, Montrose, and Osterman for this study. The code is written in Matlab, and the experiments are finalized using Octave. We do compare our results with the research work based on Deep Neural Network proposed in [211], and discrete cosine transform-least squares-extended (DCT-LSE) [212].

5.7.1 Evaluation Metrics

The metrics used in the previous chapter are also valid here. Thus, we will use the following:

- Means Square Error (MSE): an indicator to estimate the average error between the predicted and the actual value (equation provided in section 4.3.1.4).
- Means Absolute Error (MAE): an indicator to estimate the average absolute error between the predicted and the actual value (equation provided in section 4.3.1.4).
- Means Absolute Percent Error (MAPE): an indicator to estimate the average absolute percentage error between the predicted and the actual value (equation provided in section 4.3.1.4).
- Root Mean Squared Error (RMSE): an indicator of how close the predictions and observations are. It shows the standard deviation of the difference between the true value and the predicted one. The formula is as follows:

$$RMSE = \sqrt{\frac{\sum_{j=1}^{n}(x_j - x'_j)^2}{n}}$$

5.7.2 Dataset preparation

The observations are the real-time water data from three different places as discussed in section 5.7. There are a few data points where some features were missing, and thus we avoided those in our experiment. Data cleaning is an important step to initiate the evaluation of our approach and start forecasting future values.

5.7.3 DST-I

5.7.3.1 Data Prediction

The DST-based least-squares approach is used to predict values every hour at three regions ("Calumet", "Osterman", "MontRose" in Chicago District [210]) during 70-hour periods in 2014 (i.e., from 5/7/2014 12:00:00 AM to 5/7/2014 11:00:00 PM). The algorithm works as follows: we use 23 values to build the coefficients of the algorithm, and then the model predicts the future values. Every predicted value is added to the model, while the first value existing in the queue will be removed. The concept is similar to queue with FIFO data structure type.

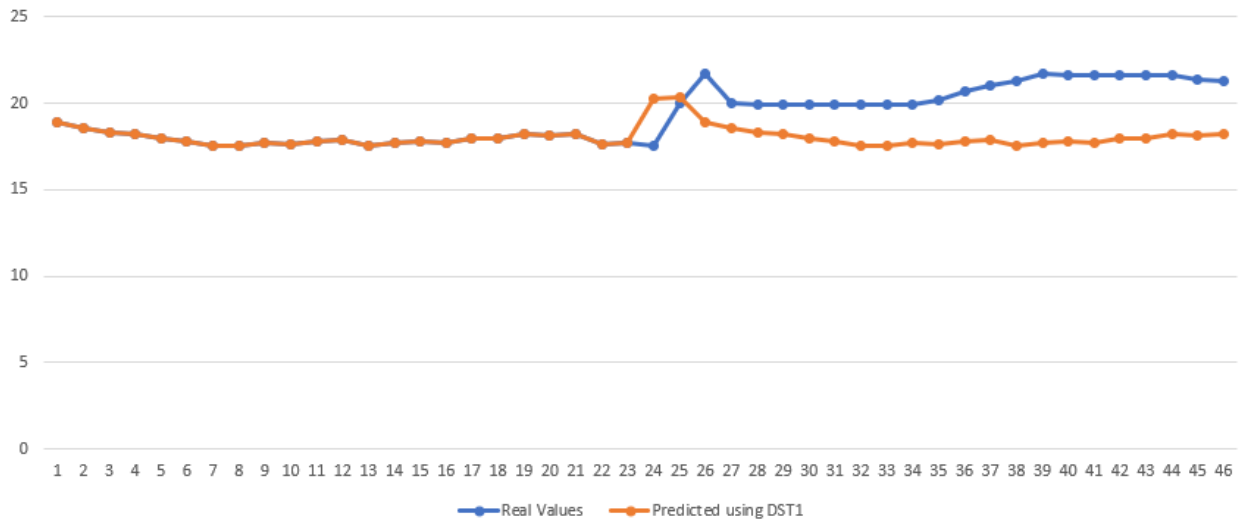


Figure 5.1: Predictions using DST-1 for 46 hours forecast of Osterman. Settings are as follows: $T_0 = T - 1, T = 24, R = 23$

As shown in Figures 5.1 - 5.3, the predicted values within the range are the same as the actual values. That means the algorithm is the perfect fit when predicting within the range. Predicting future values shows a minimal difference between predictions and actual values. As a result, the algorithm can reduce data transmission in the case of a fixed periodic date frame producing the same data at the sink node. For instance, the approach is appropriate for cases where there is not much of a need to send values instantly but instead to have a periodic timeframe where the coefficients are built and transmitted. Based on the coefficients, the sink node can produce exact values anytime. However, the future values are always within the range. They do not diverge like Newton Interpolating polynomials, where the coefficients constantly need re-evaluation because of divergence.

5.7.3.2 Data Accuracy

Using the metrics provided in section 5.7, we compare the proposed approach from three locations. The primary focus is predicting water level parameters from sensor nodes in the Chickago district. After applying the algorithm on the data set mentioned in section 5.7.3.1,

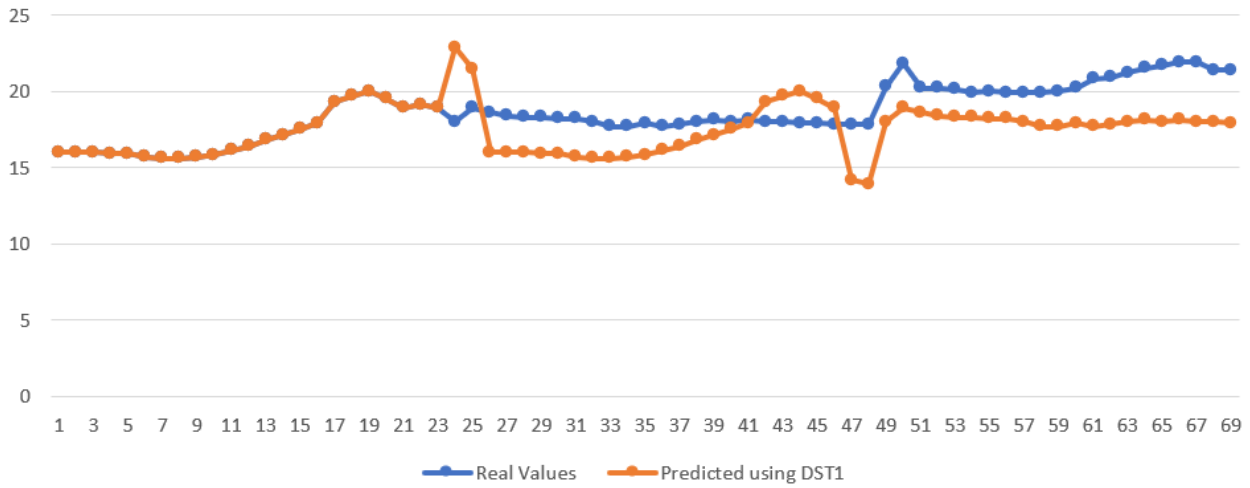


Figure 5.2: Predictions using DST-1 for 69 hours forecast of MontRose. Settings are as follows: $T_0 = T - 1, T = 24, R = 23$

the results are summarized in Table 5.1. The results represent the accuracy of hourly water temperature predictions and how well the model fits the current data.

Table 5.1: Metrics of DST-I on different regions for R=23, $T_0 = T - 1 = 23$

DST-I - Regions	Calumet	MontRose	Osterman
MSE	4.81	4.22	4.14
MAE	1.55	1.55	1.36
MAPE	7.27	7.99	6.56
RMSE	2.19	2.05	2.034

If we analyze the MAPE metrics, the model shows that the average difference between the residuals is 6.5% - 8%. That means the model is around 92% accurate. It is beneficial for applications that allow such a difference in error values. Additionally, this model is quite suitable for such applications if used to predict values within a range.

Table 5.2: Metrics for forecasting data of three algorithms

Aproach	Proposed DST-I	DNN + Rough sets[211]	DCT (average) [212]
MSE	4.14	35.68	8.94
MAE	1.36	5.16	NI
RMSE	2.034	5.85	2.99

The proposed DST-I approach has the lowest RMSE compared with the two other models from Table 5.2, and from the first perspective, it outperforms the other two approaches.

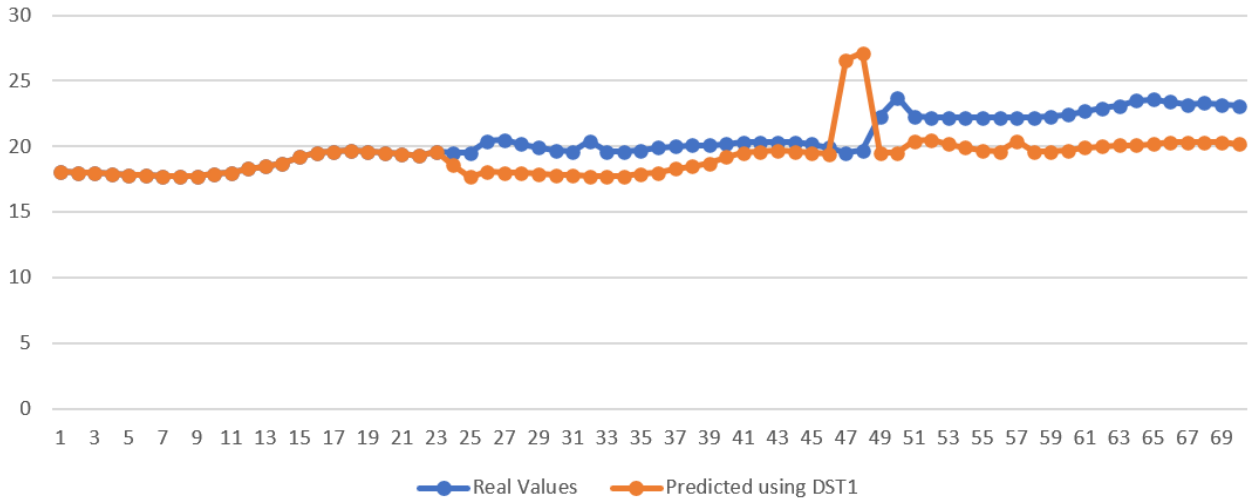


Figure 5.3: Predictions using DST-1 for 69 hours forecast of Calumet. Settings are as follows: $T_0 = T - 1, T = 24, R = 23$

However, it must be noted that the research work on [211] predicts the battery level from the same data set as in this research work (our approach is tested with water temp.), while the [212] use the data set to predict hourly electricity load movement (for 72 hours) at PJM regions.

5.7.3.3 Data Transmission and Energy Efficiency

This section discusses and analyses the energy savings of the DST-I and DST-II approaches in the WSN. There are 24 values necessary to generate the coefficients (both algorithms require this). Once the coefficients are generated, they are transmitted to the sink node, where we can generate and forecast future values—using the same formula from the previous chapter (eq. 4.2 and 4.3) for calculating the data reduction percentage and considering the RMSE as provided previously with an average of 2.2, we only need one transmission consisting of 24 coefficients. It reduces approximately 98% of data transmission considering the data set used for evaluation. However, considering equation 4.4, on how much energy we can actually save on the XSN sensor platform, we have:

Table 5.3: Energy Consumption and Saving

Algorithm	DST-I & DST-II
Energy cons. in mJ from algorithm	76.8
Energy saving in %	66 %

As Table 5.3 shows, the number of transmissions is very low considering the prediction of the next two days (hourly water temperature). However, the energy savings are only around 66% without considering the processing calculations. By extending the hourly predictions

using the same coefficients, the energy savings becomes higher (i.e., forecasting 100 future hourly values will increase to 76% the energy efficiency). In such cases, it is necessary to take care of data integrity because the values may diverge to a level that is not allowed. Once the coefficients are delivered to the sink node, the sensor platform is set to sleep mode.

5.7.3.4 Data Processing

Processing is another important factor that may affect the energy of XSN nodes. It also directly affects the information delay crucial for real WSN applications. In the proposed methods, the complexity stands in finding the coefficients of the DST-I and DST-II. The coefficients of the models proposed in this research require calculating a pure real matrix that depends on the value of R and T_0. That means the algorithm, in our case, requires 23x23 when $R = 23$ and $T_0 = 23$. That means the algorithm has a complexity of $O(RXT)$ that is approximate $O(n^2)$. Further, to finalize the coefficient calculations, the inverse of a given matrix must be resolved. That involves another $O(n^2)$ calculations. Finally we have $O(n^2)+O(n^2) = 2O(n^2) \approx O(n^2)$. The best part of the algorithm is that once the coefficients are calculated, there is no need to re-evaluate because it is considered appropriate for the use case where it is applied. The calculations provided above might have a low impact in the long term on the total energy spent.

5.7.4 DST-II

5.7.4.1 Data Predictions

The same data set as in section 5.7.3.1 is applied for evaluating the performance of DST-2. Also, under the same settings as DST-I, we evaluate the algorithm.

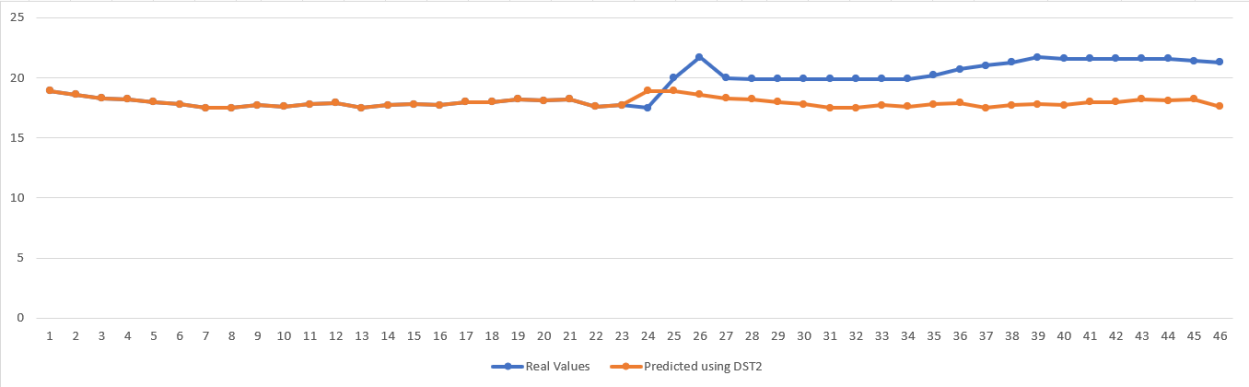


Figure 5.4: Predictions using DST-2 for 46 hours forecast of Osterman. Settings are as follows: $T_0 = T - 1, T = 24, R = 23$

The results from Figs.5.4-5.6 show that the predictions results fit quite well to the simulated data set . There is a perfect match within the given range, and slight difference outside the range.

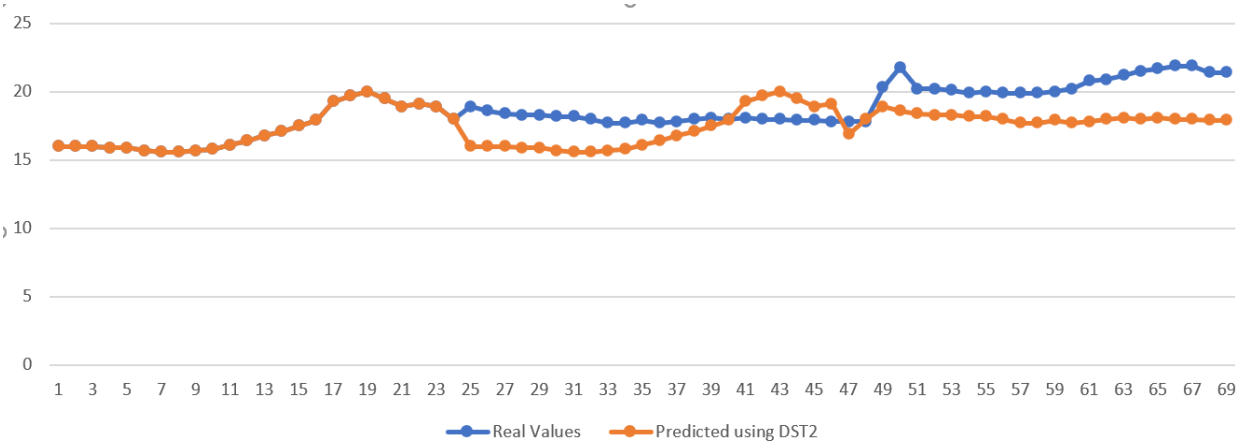


Figure 5.5: Predictions using DST-2 for 69 hours forecast of MontRose. Settings are as follows: $T_0 = T - 1, T = 24, R = 23$

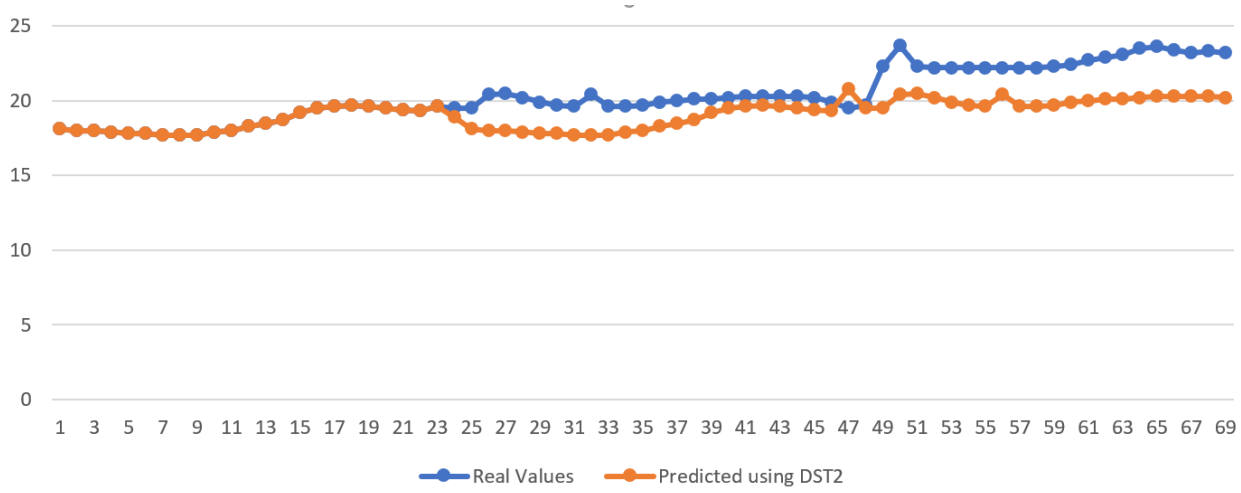


Figure 5.6: Predictions using DST-2 for 69 hours forecast of Calumet. Settings are as follows: $T_0 = T - 1, T = 24, R = 23$

5.7.4.2 Data Accuracy

Following the section 5.7.3.2, the same metrics for evaluating DST-II are provided. The results of this evaluation are summarized in Table 5.4.

From Table 5.5 we can easily see that DST-II performs better than DST-I, and thus better than the other two algorithms.

5.8 Discussion

Energy efficiency is crucial in WSN networks. The more data is transmitted within the sensor network, the more energy is spent, and battery-powered sensory devices stop functioning.

Table 5.4: Metrics of DST-I on different regions for R=23, $T_0 = T - 1 = 23$

DST-II - Regions	Calumet	MontRose	Osterman
MSE	3.14	3.47	4.15
MAE	1.32	1.37	1.37
MAPE	6.12	7.01	6.61
RMSE	1.77	1.86	2.038

Table 5.5: Metrics for forecasting data of three algorithms (Average from different regions)

Aproach	DST-II	DST-I	DNN + Rough sets[211]	DCT [212]
MSE	3.58	4.14	35.68	8.94
MAE	1.35	1.36	5.16	NI
RMSE	1.88	2.034	5.85	2.99

Fourier transform methods are widely used in several domains, such as signal processing, image processing, and interpolation. DST is a Fourier analysis method expressed as a sum of sinusoids having various amplitudes and frequencies. Although they have high applicability, they cannot be used to forecast directly. However, combining DST with least-square shows a great match as a prediction model applicable to the WSN domain to reduce data transmissions via the prediction. They are just a few coefficients that are transmitted from one node to another. The contributions of this study include
. 1. They are two prediction algorithms proposed in this study: DST-I and DST-II, both extended with least-squares error that are evaluated using the real sensor data sets from different locations of the Chicago Beach district.
2. The experiments and metrics show the proposed approaches' efficacy in predicting sensor data expressed as a time series. Comparing the metrics of both models show that DST-II slightly outperforms DST-I. Comparison of both models can be seen in Figures 5.8-5.9 and Table 5.5. Table 5.5 also shows better performance compared to the other two approaches.
3. The proposed models provide an advantage in reducing the data transmissions between the nodes. Both approaches show that using the previous 23 values to build a model allows us to forecast the next 48 values without breaking data integrity. Thus, we can easily say they are good approaches to save energy in sensor nodes by sending 24 coefficients and predicting future values with around 90% accuracy.

The proposed algorithms improve energy efficiency and reduce data transmission by a considerable percentage. However, the algorithms might be heavier while calculating the function's coefficients. It can further improve the accuracy by considering threshold parameters, which is part of future work. These parameters will automatically automate the XSN nodes to set on sleep mode and, whenever necessary, to set real value as input to the algorithm. Furthermore, the need to evaluate the energy efficiency performance in the real device will be considered.

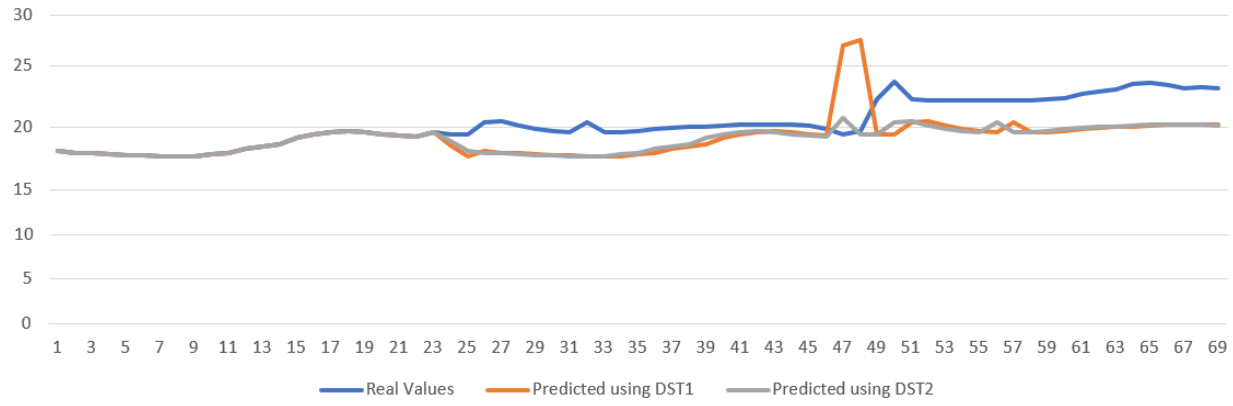


Figure 5.7: Predictions using DST-2 for 69 hours forecast of Calumet. Settings are as follows: $\mathrm{T}_0 = T - 1, T = 24, R = 23$

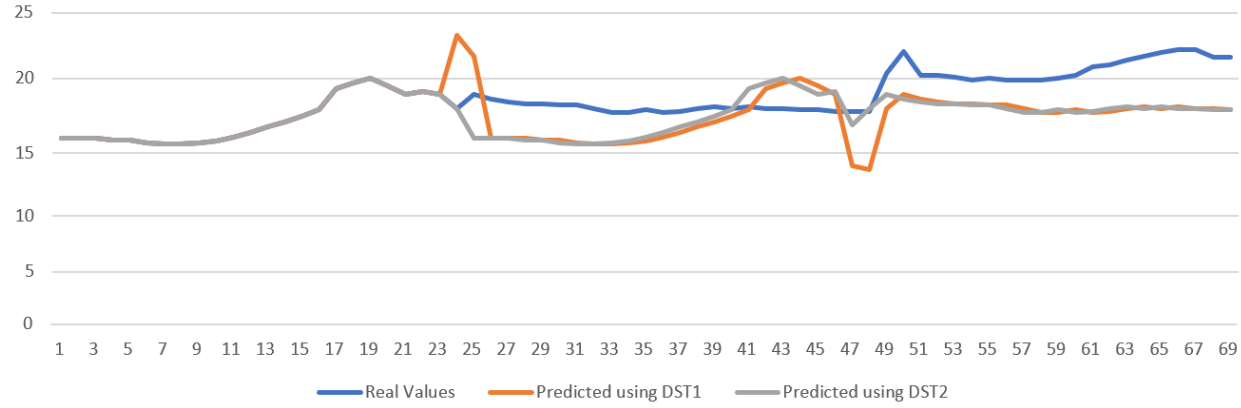


Figure 5.8: Predictions using DST-2 for 69 hours forecast of MontRose. Settings are as follows: $\mathrm{T}_0 = T - 1, T = 24, R = 23$

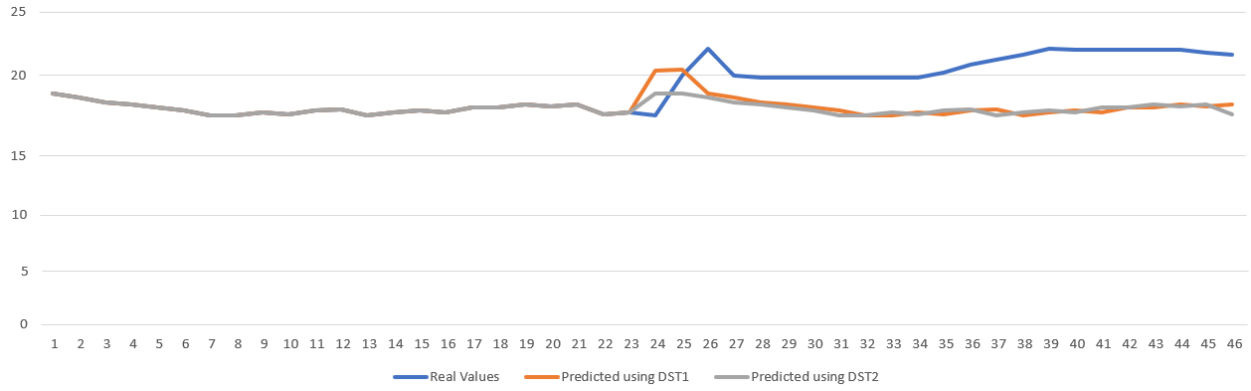


Figure 5.9: Predictions using DST-2 for 69 hours forecast of Osterman. Settings are as follows: $\mathrm{T}_0 = T - 1, T = 24, R = 23$

Chapter 6

Discrete Hartley Transform Based Forecast Modeling for Energy Efficient Wireless Sensor Network

In the previous chapter, we demonstrated the efficiency of Fourier-related transform (DST-I and DST-II combined with Least Square) as a dual data prediction reduction approach to improving energy efficiency in WSN networks. Although the model is highly accurate in predicting the data, it involves quite a number of coefficients to build the model. The purpose of this chapter is to introduce a new approach from the Fourier-related transform called Discrete Hartley Transform (DHT). The key idea is to improve efficiency with better data prediction accuracy, performance, or both. The accuracy and performance represent the main concern in the sensor nodes where constraints in memory and processing are apparent. Thus, efficient prediction aims to reduce data transmission, resulting in less energy usage from sensor nodes.

Discrete Hartley Transform (DHT) can help in this regard to transform real input from the gathered sensor information and produce real output on the sink node. DHT is a more efficient computational alternative to Discrete Fourier Transforms where the data are purely real, and the inverse and direct transformation is identical. The literature review revealed the DHT applicability in many domains like image processing, watermark image authentication, encryption, etc. Although, using DHT in the domain of WSN is limited nor existing.

As a result, this chapter introduces a new dual prediction approach based on Discrete Hartley Transform extended with least-squares. The next section 6.1 describes the problem formulation in forecasting data, then followed by section 6.2 with the new approach of the DHT-Least Square forecasting model. The rest of the sections provide the algorithm and the results in forecasting sensor data from the real data set.

6.1 Problem Description of the Hartley Transform Based Forecast Modeling

A discrete Hartley transform (DHT) is a Fourier-related transform of discrete, periodic data similar to the discrete Fourier transform (DFT), with analogous applications in signal processing and related fields. Its main distinction from the DFT is that it transforms real inputs into real outputs, with no intrinsic involvement of complex numbers.
The discrete Hartley transform (DHT) is given by

$$\mathcal{B}(s) = \sum_{t=1}^{T} \mathcal{C}(t) cas\left(\frac{2\pi(t-1)(s-1)}{T}\right), \quad s = 1, 2, ..., T, \tag{6.1}$$

where $\mathcal{C}$ represents the sequence (or called discrete time-series) of length T ($\mathcal{C}(t), t = 1, 2...., T$), $\mathcal{B}$ is the DHT of $\mathcal{C}$ ($\mathcal{C}(s), s = 1, 2, ..., T$). The combination $\cos z + \sin z = \sqrt{2}\cos\left(z - \frac{\pi}{4}\right)$ is sometimes denotes $cas(z)$ and should not confused with $cisz = e^{iz} = \cos(z) + i\sin(z)$ or $e^{-iz} = cis(-z)$. Its corresponding inverse sine transform, called the inverse DHT (IDHT) is defined by,

$$\mathcal{C}(t) = \frac{1}{T}\sum_{s=1}^{T} \mathcal{B}(s) cas\left(\frac{2\pi(t-1)(s-1)}{T}\right), \quad s = 1, 2, ..., T. \tag{6.2}$$

The definitions of Discret Hartley Transform (Eqs.(6.1)-(6.2)) show that the DTS (discrete time-series) $\mathcal{C}(t)(t = 1, 2, \ldots, T)$ and the Discret Hartley transform coefficients $\mathcal{C}(s)(s = 1, 2, \ldots, T)$ are the same size of T-length. When given $T_0(< T)$ limited for observations of water temperature for an hour: $\mathcal{C}(t)(t = 1, 2 \ldots T_0)$, we now consider how to extend the Discrete Hartley model (relation (6.2)) to forecast its future movement at its next time point ($\mathcal{C}(t)(t = T_0 + 1)$) or at its succeeding points in time: $\mathcal{C}(t)(t = T_0 + 1 \ldots T)$. However, because of $T_0 < T$, the transform coefficients $\mathcal{B}(s)$ cannot be straightly computed by using the original definition of relation (6.1). Then the calculation of the Discrete Hartley transforms coefficients $\mathcal{B}(s)$ on the basis of the previous limited for observations water temperature for an hour $\mathcal{C}(t)(t = 1, 2, \ldots, T_0)(T_0 < T)$ becomes the key problem for the forecast modeling. The following section presents one LSO (least-squares-optimum)-based solution for seeking the Discrete Hartley transform coefficient.

6.2 Dicrete Hartley Transforms Model Extended in the Least-squares for Forecasting

Given a limited number $T_0(< T)$ of the observations water temperature for an hour: $\mathcal{C}(t)(t = 1, 2, \ldots, T)$, we now focus on how to extend the Discrete Hartley Transforms model (relation 6.3) to predict fluctuations of water temperature for an hour at its subsequent time points in the future: $\mathcal{C}(t)(t = N_0 + 1, \ldots, N)$. The major problem (Yang, 2018) is to estimate the Discrete Hartley Transforms coefficient $\mathcal{B}(s)$ in relation (6.2). We now try to solve the

problem in the least-squares sense. That is, to observations water temperature for hour: $\mathcal{C}(t)(t = 1, 2, ..., T_0)(T_0 < T)$,we try to get its Discrete Hartley Transform coefficient $\mathcal{B}(s)$ for building the Discrete Hartley model with employing R-term harmonics, which yields $\hat{\mathcal{C}}(t)(t = 1, 2, \ldots, T)$ most fitting the given $\mathcal{C}(t)(t = 1, 2, \ldots, T)$ in the least-square sense,

$$\hat{\mathcal{C}}(t) = \frac{1}{T}\sum_{s=1}^{R}\hat{\mathcal{B}}(s)cas\left(\frac{2\pi(s-1)(t-1)}{T}\right) \tag{6.3}$$

Denote $\hat{\mathcal{E}}^2$ as the sum of squares of the errors (or called the deviations) $(\hat{\epsilon}_t, t = 1, 2, \ldots, T_0)$. Now we are to find its optimal solution, which subjects to that sum $\hat{\mathcal{E}}^2$ of squared deviations or called residuals $(\hat{\epsilon}_t, t = 1, 2 \ldots T_0)$ (relation (6.4)) is one minimum, usually put $R \leq T_0 < T$,

$$\begin{aligned}\hat{\mathcal{E}}^2 &= \sum_{t=1}^{T_0}\hat{\epsilon}_t^2 = \sum_{t=1}^{T_0}\left(\mathcal{C}(t) - \hat{\mathcal{C}}(t)\right)^2 \\ &= \sum_{t=1}^{T_0}\left[\mathcal{C}(t) - \frac{1}{T}\sum_{s=1}^{R}\hat{\mathcal{B}}(s)cas\left(\frac{2\pi(s-1)(t-1)}{T}\right)\right]^2\end{aligned} \tag{6.4}$$

The called LSO (least-squares-optimum) solution for the DHT-LS-extended forecast model (relation (6.3)) indicates that the sum of squares of the errors $\hat{\mathcal{E}}^2$ (relation (6.4)) will be the least amount possible. To minimize $\hat{\mathcal{E}}^2$ (relation (6.4)) with respect to $\hat{\mathcal{B}}(s)$, $\left(\frac{\partial\hat{\mathcal{E}}^2}{\partial\hat{\mathcal{B}}(s)},\ s = 1, 2, ..., R\right)$ it follows,

$$\sum_{t=1}^{T_0}\frac{1}{T}cas\left(\frac{2\pi(s-1)(t-1)}{T}\right)\left[\mathcal{C}(t) - \sum_{s=1}^{R}\frac{1}{T}\hat{\mathcal{B}}(s)cas\left(\frac{2\pi(s-1)(t-1)}{T}\right)\right] = 0 \tag{6.5}$$

from here we have

$$\begin{aligned}\sum_{t=1}^{T_0}\frac{1}{T}\hat{\mathcal{B}}(s)cas\left(\frac{2\pi(s-1)(t-1)}{T}\right) &= \sum_{t=1}^{T_0}\frac{1}{T}cas\left(\frac{2\pi(s-1)(t-1)}{T}\right) \\ &\quad\cdot\sum_{s=1}^{R}\frac{1}{T}\hat{\mathcal{B}}(s)cas\left(\frac{2\pi(s-1)(t-1)}{T}\right).\end{aligned} \tag{6.6}$$

Let $g(r,t) = \frac{1}{T}cas\left(\frac{2\pi(r-1)(t-1)}{T}\right)$, it results in,

$$\sum_{t=1}^{T_0}g(r,t)\mathcal{C}(t) = \sum_{t=1}^{T_0}g(r,t)\sum_{s=1}^{T}\hat{\mathcal{B}}(s)g(s,t),\ \text{ for } r = 1, 2, ..., R. \tag{6.7}$$

For the simplicity of the above expression, further definitions are introduced as

$$W_{R\times T_0} = \begin{bmatrix} g(1,1) & g(1,2) & \cdots & g(1,T_0) \\ g(2,1) & g(2,2) & \cdots & g(2,T_0) \\ \vdots & \vdots & \ddots & \vdots \\ g(R,1) & g(R,2) & \cdots & g(R,T_0) \end{bmatrix}$$

and

$$\mathcal{Z}_{T_0\times 1}=\begin{bmatrix}\mathcal{Z}(1)\\ \mathcal{Z}(2)\\ \vdots\\ \mathcal{Z}(T_0)\end{bmatrix}\quad \hat{\mathcal{B}}_{R\times 1}=\begin{bmatrix}\hat{\mathcal{B}}(1)\\ \hat{\mathcal{B}}(2)\\ \vdots\\ \hat{\mathcal{B}}(R)\end{bmatrix}$$

From here and from equation (6.7) it follows

$$W_{R\times T_0}\cdot\mathcal{Z}_{T_0\times 1}=W_{R\times T_0}\left(\hat{\mathcal{B}}^T_{R\times 1}W_{R\times T_0}\right)'=W_{R\times T_0}\cdot W'_{T_0\times R}\cdot\hat{\mathcal{B}}_{R\times 1}\tag{6.8}$$

Note that for $r=1,2\ldots R$, we have

$$\sum_{t=1}^{T_0}g(r,t)\sum_{s=1}^{T}\hat{\mathcal{B}}(s)g(s,t)=W_{R\times T_0}\left(\hat{\mathcal{Z}}'_{R\times 1}W_{R\times T_0}\right)'\tag{6.9}$$

From the relation (6.8), on the basis of the prior finite T_0 observations of water temperature for hour $\mathcal{C}(t)(t=1,2,...,T_0)(T_0<T)$, the called $R-$term LSO (least-squares-optimum) DHT coefficients $\hat{\mathcal{B}}(t)(t=1,2,...,R)$ for building the DHT-LS-extended forecast model (relation.(6.3)) are determined by,

$$\hat{\mathcal{B}}_{R\times 1}=\left(W_{R\times T_0}W'_{T_0\times R}\right)^{-1}W_{R\times T_0}\mathcal{Z}_{T_0\times 1}\tag{6.10}$$

Finally, the Discrete Hartley Transform-LS-extended prediction model (relation (6.3)) can now be established with the LSO (least-squares-optimum) DHT coefficient $\hat{\mathcal{B}}_{R\times 1}$ (relation (6.10)) available. Then we can employ the Discrete Hartley Transform-LS-extended prediction model (relation (6.3)) to predict the future fluctuation of the water temperature for an hour at its next point of time: $\hat{\mathcal{C}}(t)(t=T_0+1)$ or ones at its subsequent points in time: $\hat{\mathcal{C}}(t)(t=T_0+1,\ldots,T)$.

6.3 Algorithms

Finding coefficients of the DHT are as follows:

Given the limited number of the sensor values T_0 $(<T)$ of daily water temperature data over a given period of time, described using the model $\hat{\mathcal{C}}(t)(t=1,2,\ldots,T_0)$, it is required to extend the presented model of the DHT defined by relation (6.3) to predict daily water temperature movement at future points: $\hat{\mathcal{C}}(t=T_0+1,...,T)$.

6.3.1 CAS Algorithm

INPUT: x,R,T,T_0
Output: Coefficient set C_i

1. SET B = $\{\}_{RXT_0}$;
2. **for** n = 1 to T_0 do
3. **for** $m = 1$ to R do
4. $B(m,n) = (1/R) * (sin(((m-1) * 2 * pi * (n-1))/T) + cos(((m-1) * 2 * pi * (n-1))/T))$;

end for
5. **end for**
6. SET $C_i = inv(B * B') * B * x$;
11. **return** C_i

Algorithm to evaluate the new data point. This stage aims to predict new values using coefficients from CAS produced by algorithm 1.
INPUT: C - coefficients, R, T, T_0
Output: S - predicted value
1. SET S = $\{\}_{T_0+1X1}$;
2. **for** n = 1 to $T_0 + 1$ do
3. **for** $k = 1$ to R do
4. $S(n) = S(n) + (1/R) * (sin(((k-1) * 2 * pi * (n-1))/T) + cos(((k-1) * 2 * pi * (n-1))/T)) * P(k)$;
5. **end for**
6. **end for**
7. **return** S

6.4 Experimental Results

6.4.1 Data Prediction and Data Accuracy of CAS Approach

The DHT-based CAS approach uses the same data as the previous chapter from the Chicago District [210]) during 70-hour periods in 2014 (i.e., from 5/7/2014 12:00:00 AM to 5/7/2014 11:00:00 PM). There are different settings for the DHT-based CAS approach: 23 values serve as input to build the coefficients, then the coefficients are transmitted to the sink node necessary to predict future values. In a similar manner as previous approaches, DST-I and DST-II, every time a prediction is made, it is added to the model, and the first real value from the queue is removed.

Table 6.1: Metrics of DHT for Calumet region with R=23, $T_0 = T - 1 = 23$

DHT	R=12	R=8	R=6	R=4	R=2
MSE	0.86	0.58	0.51	0.48	0.52
MAE	0.61	0.54	0.51	0.5	0.51
MAPE	3.05	2.66	2.54	2.47	2.52
RMSE	0.93	0.76	0.71	0.69	0.72

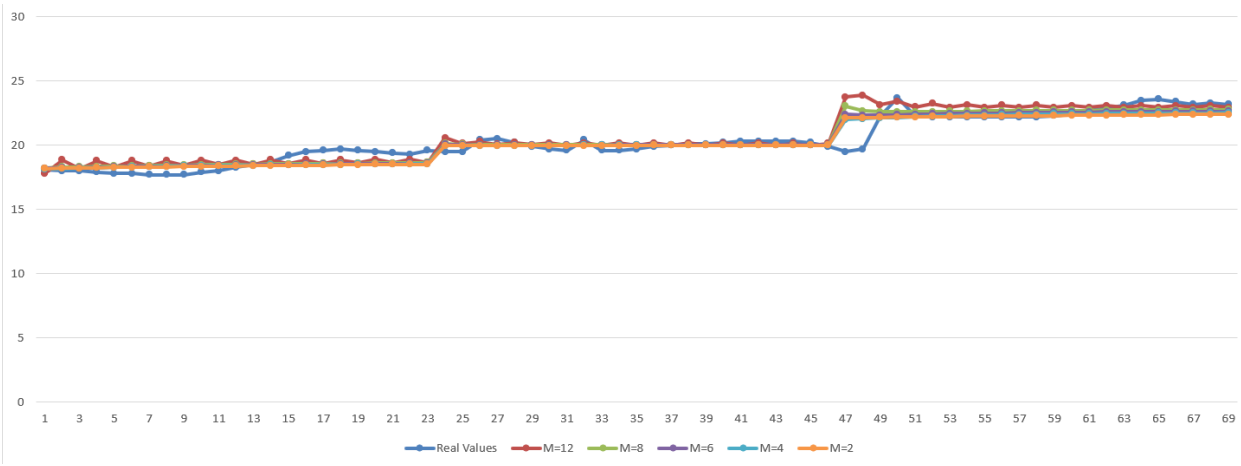


Figure 6.1: Predictions using DHT for 70 hours forecast of Calumet. Settings are as follows: $T_0 = T - 1, T = 24, R = 12$

Table 6.2: Metrics of DHT for MontRose region with R=23, $T_0 = T - 1 = 23$

DHT	R=12	R=8	R=6	R=4	R=2
MSE	1.37	1.07	1.01	0.99	1.07
MAE	0.849	0.74	0.72	0.7	0.72
MAPE	4.67	4.05	3.89	3.79	3.9
RMSE	1.17	1.03	1	0.99	1.03

Table 6.3: Metrics of DHT for Osterman region with R=23, $T_0 = T - 1 = 23$

DHT	R=12	R=8	R=6	R=4	R=2
MSE	0.92	0.56	0.49	0.48	0.5
MAE	0.59	0.53	0.52	0.51	0.52
MAPE	3	2.73	2.64	2.62	2.68
RMSE	0.96	0.74	0.7	0.69	0.71

Figures 6.1 - 6.3 present graphics comparing the behavior of the algorithm for different R values. From the graphs, it can easily be seen that all the settings are nearby the real values, and the differences are very small. It can be noticed easily for R = 4, the line between actual and predicted values matches a lot. Tables 6.1 - 6.3 provide data accuracy and how well the models fit the data set. It must be noted from the tables above that a very low RMSE, means that the model performs quite well when it comes to data accuracy.

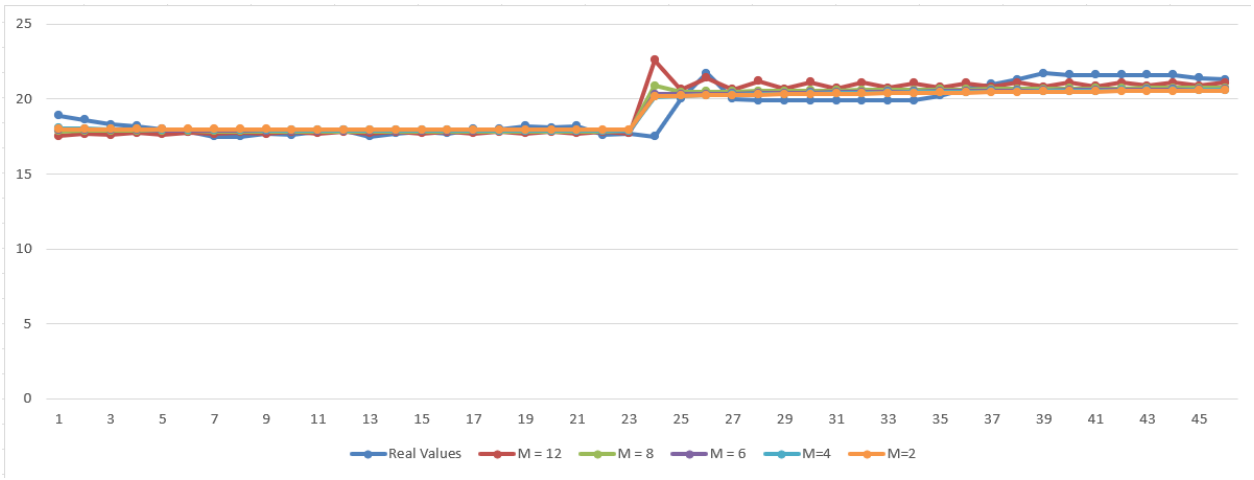


Figure 6.2: Predictions using DHT for 70 hours forecast of Osterman. Settings are as follows: $T_0 = T - 1, T = 24, R = 12$

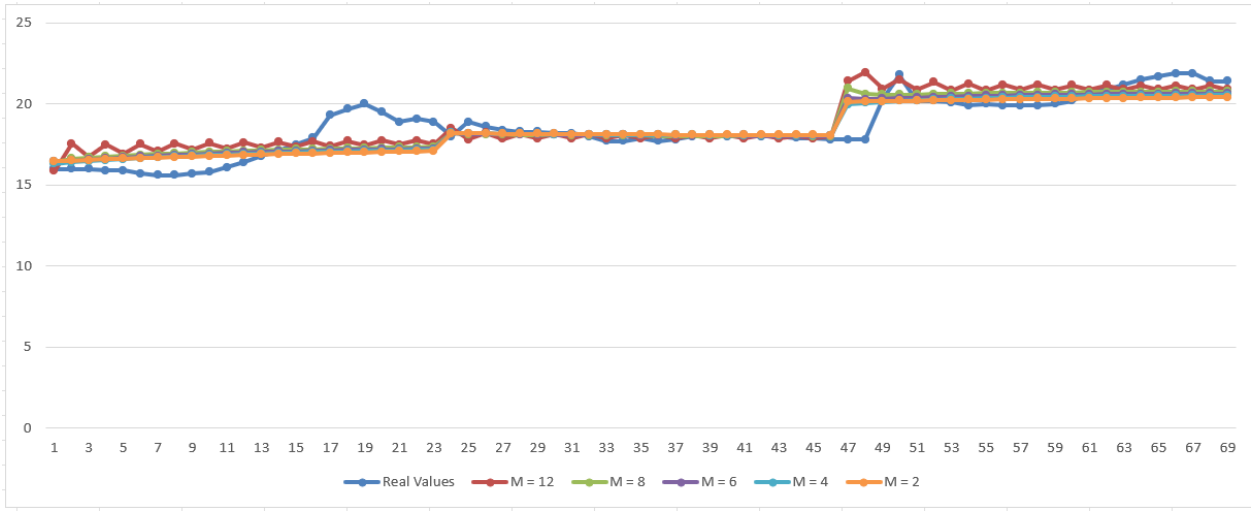


Figure 6.3: Predictions using CAS for 70 hours forecast of Calumet. Settings are as follows: $T_0 = T - 1, T = 24, R = 12$

6.4.2 Data Transmission and Energy Efficiency

The algorithm is tested with 23 input observations for generating coefficients of degrees 2, 4, 6, 8, and 12. Using the equation 4.2 from the previous chapter to calculate the percentage in reducing the transmissions and considering the RMSE with an average of 1.06 from three regions, it is necessary to transmit only once with four coefficients of the function considering the best case scenario from the evaluation of the date set. It reduces 99% of data transmission out of 70 hourly forecasts. Considering equation 4.4 in evaluating the energy efficiency on the XSN platform, table 6.5 provides information on energy saving. It must be noted that the equation used for evaluating energy consumption does not consider processing power.

The DHT approach does not require many calculations. They are 4x23 calculations when

Table 6.4: Energy Consumption and Saving

Algorithm	DHT for R = 4	DHT for R = 12
Energy cons. in mJ from algorithm	12.8	38.4
Energy saving in %	94 %	82%

$R = 4$ and $T_0 = 23$. In general, the processing time is to calculate a matrix of RXT_o. Once the coefficients are ready, the node can be set to sleeping mode, and the sink node can predict future values based on the DHT algorithm. Furthermore, having an accuracy of 95%-97% based on metrics provided in Table 6.1 - 6.3, it can be easily concluded that the algorithm has a good performance and can help a lot in improving the network lifetime.

6.5 Discussion

This chapter introduced a new valuable model for improving energy efficiency via prediction. The model extends DHT with a least square method to improve the data prediction. The extended model shows an excellent match for predicting sensor data in WSN without breaking data integrity. The model can achieve high accuracy when $R = 4$, meaning that we can only send coefficients in one transmission and make accurate predictions for at least the next 70 hours. Furthermore, the algorithm can work in constrained devices with limited processing capabilities.

Comparing DHT with previous algorithms (DST-I and DST-II from the previous chapter), as well as DNN - Rough sets[211] and DCT [212], provides better performance and accuracy. The table below presents a comparison of these algorithms.

Table 6.5: Energy Consumption and Saving

Algorithm	DHT	DCT-I	DCT-II	DNN [211]	DCT [212]
MSE	0.48	3.58	4.14	35.68	8.94
MAE	0.51	1.35	1.36	5.16	NI
RMSE	0.69	1.88	2.034	5.85	2.99

It can be concluded based on experimental results that the algorithm is effective and reduces data transmission. It is not highly effective as DST-I and DST-II when predicting values within the range(producing exact values), however, both (DST-I and DST-II) have higher settings (R=24). Otherwise DHT algorithm is highly accurate even when predicting within the range. The higher the R value, the more processing is required. Future work includes measuring the performance of the approach in real devices, considering erroneous data and detecting those erroneous data.

Chapter 7

Future Work

In this dissertation, we have worked on finding the best possible network architecture based on the requirements elaborated in chapter II. Then we proposed a new data architecture for the network architecture. Further, several models are proposed to improve energy efficiency via the predictions. Then, we evaluate the performance from the real dataset of the movement of water temperatures using the finite Furie series. The models created by this study address the load predictions based on the discrete interpolation of:

- Furier transformation (DFT).
- Discrete Hartley Transforms (DHT)
- Discrete Sine transform DST-I
- Discrete Sine transform DST-II
- Newton Interpolation

In particular, comparison of DHT and DST models have been made.

Problems that remain to be explored in the future:

• Generalization of these models for the multidimensional case.

• Introduction of any other factor affecting daily water temperature movement and expansion of forecasting models using the least-squares method for DFT, DCT, DST, DHT, etc.

• Combining models examined using neural networks for prediction.

• Review of DFT, DCT, DST, DHT prediction models using Monte Carlo simulation

• Definition of Max-product sampling type operators and applications to image processing

Bibliography

[1] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics reports*, vol. 424, no. 4-5, pp. 175–308, 2006.

[2] O. Michail and P. G. Spirakis, "Elements of the theory of dynamic networks," *Commun. ACM*, vol. 61, pp. 72–72, Jan. 2018.

[3] Cisco, "Internet of things." `https://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf`, accessed: October, 2015).

[4] D. Evans, "The internet of everything." `https://www.cisco.com/c/dam/global/en_my/assets/ciscoinnovate/pdfs/IoE.pdf`, accessed: December, 2018).

[5] D. Doty, "Theory of algorithmic self-assembly," *Communications of the ACM*, vol. 55, no. 12, pp. 78–88, 2012.

[6] G. B. Smith, B. Hein, D. E. Whitney, D. Fitzpatrick, and M. Kaschube, "Distributed network interactions and their emergence in developing neocortex," *Nature Neuroscience*, vol. 21, no. 11, p. 1600, 2018.

[7] F. Kuhn, N. Lynch, and R. Oshman, "Distributed computation in dynamic networks," in *Proceedings of the forty-second ACM symposium on Theory of computing*, pp. 513–522, ACM, 2010.

[8] A. Casteigts, "Finding structure in dynamic networks," *CoRR*, vol. abs/1807.07801, 2018.

[9] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: uses and interpretations," *Neuroimage*, vol. 52, no. 3, pp. 1059–1069, 2010.

[10] I. Pestov, "Dynamic network analysis for understanding complex systems and processes," *Defence Research and Development Canada*, 2009.

[11] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary perspectives on complex systems*, pp. 85–113, Springer, 2017.

[12] M. Mitchell, *Complexity: A guided tour*. Oxford University Press, 2009.

[13] G. Sargut and R. G. McGrath, "Learning to live with complexity," *Harvard business review*, vol. 89, no. 9, pp. 68–76, 2011.

[14] H. Abbas, S. Shaheen, M. Elhoseny, A. K. Singh, and M. Alkhambashi, "Systems thinking for developing sustainable complex smart cities based on self-regulated agent systems and fog computing," *Sustainable Computing: Informatics and Systems*, 2018.

[15] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[16] B. Xu, L. Da Xu, H. Cai, C. Xie, J. Hu, and F. Bu, "Ubiquitous data accessing method in iot-based information system for emergency medical services," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1578–1586, 2014.

[17] J. Elson and K. Römer, "Wireless sensor networks: A new regime for time synchronization," *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 149–154, Jan. 2003.

[18] P. Anantharam, P. Barnaghi, and A. Sheth, "Data processing and semantics for advanced internet of things (iot) Applications: modeling, annotation, integration, and perception," in *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, p. 5, ACM, 2013.

[19] X. Jia, Q. Feng, T. Fan, and Q. Lei, "Rfid technology and its applications in internet of things (iot)," in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, pp. 1282–1285, IEEE, 2012.

[20] D. Uckelmann, M. Harrison, and F. Michahelles, "An architectural approach towards the future internet of things," in *Architecting the internet of things*, pp. 1–24, Springer, 2011.

[21] J. A. Stankovic, "Research challenges for wireless sensor networks," *ACM SIGBED Review*, vol. 1, no. 2, pp. 9–12, 2004.

[22] X. Su, J. Riekki, J. K. Nurminen, J. Nieminen, and M. Koskimies, "Adding semantics to internet of things," *Concurrency and Computation: Practice and Experience*, Jan. 2014.

[23] P. Koskela and M. Majanen, "Robust header compression for constrained application protocol," pp. 36–39, Jan. 2014.

[24] B. Madhav and A. Ralegaonkar, "Wireless sensor network: A promising approach for distributed sensing tasks," *Excel Journal of Engineering Technology and Management Science*, vol. Vol. I No.1, Jan. 2012.

[25] T. J. Dishongh, M. McGrath, and B. Kuris, *Wireless sensor networks for healthcare applications*. Artech House, 2014.

[26] M. Atto and C. Guy, "A cross layer protocol based on mac and routing protocols for healthcare applications using wireless sensor networks," *International Journal of Advanced Smart Sensor Network Systems (IJASSN)*, vol. 4, no. 2, 2014.

[27] Y. Duan and T. Guo, "Research and Application of wireless sensor networks in agriculture," in *First International Conference on Information Sciences, Machinery, Materials and Energy*, Atlantis Press, 2015.

[28] A. Z. Abbasi, N. Islam, Z. A. Shaikh, *et al.*, "A review of wireless sensors and networks' applications in agriculture," *Computer Standards & Interfaces*, vol. 36, no. 2, pp. 263–270, 2014.

[29] A. Berger, L. B. Hormann, C. Leitner, S. B. Oswald, P. Priller, and A. Springer, "Sustainable energy harvesting for robust wireless sensor networks in industrial applications," pp. 1–6, IEEE, Apr. 2015.

[30] A. A. Kumar Somappa, K. Øvsthus, and L. M. Kristensen, "An industrial perspective on wireless sensor networks—a survey of requirements, protocols, and challenges," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 3, pp. 1391–1412, 2014.

[31] Z. Zhou, D. Zhao, X. Xu, C. Du, and H. Sun, "Periodic Query Optimization Leveraging Popularity-Based Caching in Wireless Sensor Networks for Industrial IoT Applications," *Mobile Networks and Applications*, vol. 20, pp. 124–136, Apr. 2015.

[32] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[33] "Wireless sensor networks: creating a $ 2 billion market in 2021." `http://www.idtechex.com/research/articles/wireless-sensor-networks-creating-a-2-billion-market-in-2021-00003663.asp?donotredirect=true`, accessed: april, 2015.

[34] A. Chandor, J. Graham, R. Williamson, *et al.*, "Dictionary of computers," 1970.

[35] P. A. Bernstein, "Middleware: a model for distributed system services," *Communications of the ACM*, vol. 39, no. 2, pp. 86–98, 1996.

[36] A. T. Campbell, G. Coulson, and M. E. Kounavis, "Managing complexity: Middleware explained," *IT professional*, vol. 1, no. 5, pp. 22–28, 1999.

[37] D. Bakken, "Middleware," *Encyclopedia of Distributed Computing*, vol. 11, 2001.

[38] W. Emmerich, M. Aoyama, and J. Sventek, "The impact of research on middleware technology," *ACM SIGSOFT Software Engineering Notes*, vol. 32, no. 1, pp. 21–46, 2007.

[39] R. Sugihara and R. K. Gupta, "Programming models for sensor networks: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 2, pp. 1–29, 2008.

[40] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[41] K. Römer, O. Kasten, and F. Mattern, "Middleware challenges for wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 4, pp. 59–61, 2002.

[42] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE access*, vol. 5, pp. 1872–1899, 2017.

[43] T. Maitra and S. Roy, "Research challenges in ban due to the mixed wsn features: some perspectives and future directions," *IEEE Sensors Journal*, vol. 17, no. 17, pp. 5759–5766, 2017.

[44] A. Herzog, D. Jacobi, and A. Buchmann, "A3me-an agent-based middleware approach for mixed mode environments," in *2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 191–196, IEEE, 2008.

[45] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan, "Smart semantic middleware for the internet of things," in *International Conference on Informatics in Control, Automation and Robotics*, vol. 2, pp. 169–178, SciTePress, 2008.

[46] 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.

[47] K. Römer, O. Kasten, and F. Mattern, "Middleware challenges for wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 4, pp. 59–61, 2002.

[48] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Issues in designing middleware for wireless sensor networks," *IEEE network*, vol. 18, no. 1, pp. 15–21, 2004.

[49] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for internet of things: a survey," *IEEE Internet of things journal*, vol. 3, no. 1, pp. 70–95, 2015.

[50] A.-B. García-Hernando, J.-F. Martínez-Ortega, J.-M. López-Navarro, A. Prayati, and L. Redondo-López, "Problem solving for wireless sensor networks," 2008.

[51] T. C. Collier and C. Taylor, "Self-organization in sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 866–873, 2004.

[52] F. Baxhaku, G. Eleftherakis, and E. Vasilaki, "Knowledge representation, languages, tools, and frameworks in medical domain," in *9th Annual South-East European Doctoral Student Conference*, p. 433.

[53] R. Frei and G. Di Marzo Serugendo, "Advances in complexity engineering," *International Journal of Bio-Inspired Computation*, vol. 3, no. 4, pp. 199–212, 2011.

[54] S. Poslad, *Ubiquitous computing: smart devices, environments and interactions*. John Wiley & Sons, 2011.

[55] M. R. Nami and K. Bertels, "A survey of autonomic computing systems," in *Autonomic and Autonomous Systems, 2007. ICAS07. Third International Conference on*, pp. 26–26, IEEE, 2007.

[56] M. San Miguel, J. H. Johnson, J. Kertesz, K. Kaski, A. Díaz-Guilera, R. S. MacKay, V. Loreto, P. Érdi, and D. Helbing, "Challenges in complex systems science," *The European Physical Journal Special Topics*, vol. 214, no. 1, pp. 245–271, 2012.

[57] S. Bosmans, S. Mercelis, P. Hellinckx, and J. Denil, "Towards evaluating emergent behavior of the internet of things using large scale simulation techniques (wip)," in *Proceedings of the Theory of Modeling and Simulation Symposium*, p. 4, Society for Computer Simulation International, 2018.

[58] K. Henricksen and R. Robinson, "A survey of middleware for sensor networks: state-of-the-art and future directions," in *Proceedings of the international workshop on Middleware for sensor networks*, pp. 60–65, 2006.

[59] M. M. Molla and S. I. Ahamed, "A survey of middleware for sensor network and challenges," in *2006 International Conference on Parallel Processing Workshops (ICPPW'06)*, pp. 6–pp, IEEE, 2006.

[60] Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric internet of things," *Journal of Network and Computer Applications*, vol. 64, pp. 137–153, 2016.

[61] S. A. Chelloug and M. A. El-Zawawy, "Middleware for internet of things: Survey and challenges," *Intelligent Automation & Soft Computing*, pp. 1–9, 2017.

[62] M.-M. Wang, J.-N. Cao, J. Li, and S. K. Dasi, "Middleware for wireless sensor networks: A survey," *Journal of computer science and technology*, vol. 23, no. 3, pp. 305–326, 2008.

[63] R. Sugihara and R. K. Gupta, "Programming models for sensor networks: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 2, pp. 1–29, 2008.

[64] M. Kuorilehto, M. Hännikäinen, and T. D. Hämäläinen, "A survey of application distribution in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 5, pp. 1–15, 2005.

[65] X. Li and S. Moh, "Middleware systems for wireless sensor networks: A comparative survey," *Contemporary Engineering Sciences*, vol. 7, no. 13, pp. 649–660, 2014.

[66] K. Terfloth, G. Wittenburg, and J. Schiller, "Facts-a rule-based middleware architecture for wireless sensor networks," in *2006 1st International Conference on Communication Systems Software & Middleware*, pp. 1–8, IEEE, 2006.

[67] V. Terziyan, O. Kaykova, and D. Zhovtobryukh, "Ubiroad: Semantic middleware for context-aware smart road environments," in *2010 Fifth international conference on internet and web applications and services*, pp. 295–302, IEEE, 2010.

[68] T. Kato, H. Takahashi, and T. Kinoshita, "Multiagent-based autonomic and resilient service provisioning architecture for the internet of things," *IJCSNS*, vol. 17, no. 6, p. 36, 2017.

[69] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "Iot middleware: A survey on issues and enabling technologies," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1–20, 2016.

[70] P. Bonnet, J. Gehrke, and P. Seshadri, "Towards sensor database systems," in *Mobile Data Management*, pp. 3–14, Springer, 2001.

[71] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks," in *Mobile Data Management, 2007 International Conference on*, pp. 198–205, IEEE, 2007.

[72] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko, *et al.*, "Openiot: Open source internet-of-things in the cloud," in *Interoperability and Open-Source Solutions for the Internet of Things*, pp. 13–25, Springer, 2015.

[73] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for internet of things: a survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70–95, 2016.

[74] E. G. et al., "Architecting the iot paradigm: A middleware for autonomous distributed sensor networks," no. 7, p. 15, 2015.

[75] M. Islam, M. M. Hassan, G.-W. Lee, E.-N. Huh, *et al.*, "A survey on virtualization of wireless sensor networks," *Sensors*, vol. 12, no. 2, pp. 2175–2207, 2012.

[76] J.-H. Shin and D. Park, "A virtual infrastructure for large-scale wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2853–2866, 2007.

[77] V. Issarny, M. Caporuscio, and N. Georgantas, "A perspective on the future of middleware-based software engineering," *Future of Software Engineering (FOSE'07)*, pp. 244–258, 2007.

[78] Y. Labrou, "Agents and ontologies for e-business," *The Knowledge Engineering Review*, vol. 17, no. 01, pp. 81–85, 2002.

[79] M. P. Papazoglou, "Agent-oriented technology in support of e-business," *Communications of the ACM*, vol. 44, no. 4, pp. 71–77, 2001.

[80] M. Vinyals, J. A. Rodriguez-Aguilar, and J. Cerquides, "A survey on sensor networks from a multiagent perspective," *The Computer Journal*, p. bxq018, 2010.

[81] A. R. Silva, A. Romão, D. Deugo, and M. M. Da Silva, "Towards a reference model for surveying mobile agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 4, no. 3, pp. 187–231, 2001.

[82] A. Fuggetta, G. P. Picco, and G. Vigna, "Understanding code mobility," *IEEE Transactions on software engineering*, vol. 24, no. 5, pp. 342–361, 1998.

[83] S. Bosse and F. Pantke, "Distributed computing and reliable communication in sensor networks using multi-agent systems," *Production Engineering*, vol. 7, no. 1, pp. 43–51, 2013.

[84] A. Newell, "The knowledge level," *Artificial intelligence*, vol. 18, no. 1, pp. 87–127, 1982.

[85] P. A. Buhler and J. M. Vidal, "Towards adaptive workflow enactment using multiagent systems," *Information technology and management*, vol. 6, no. 1, pp. 61–87, 2005.

[86] D. Greenwood and M. Calisti, "Engineering web service-agent integration," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 2, pp. 1918–1925, IEEE, 2004.

[87] M. Mamei and F. Zambonelli, *Field-based coordination for pervasive multiagent systems*. Springer Science & Business Media, 2006.

[88] H. Qi and F. Wang, "Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks," *Proceedings of the IEEE*, pp. 147–153, 2001.

[89] K. Potiron, A. E. F. Seghrouchni, and P. Taillibert, *From fault classification to fault tolerance for multi-agent systems*, vol. 8. Springer, 2013.

[90] M. Eigen and P. Schuster, "The hypercycle," *Naturwissenschaften*, vol. 65, no. 1, pp. 7–41, 1978.

[91] E. Dujardin and S. Mann, "Bio-inspired materials chemistry," *Advanced Materials*, vol. 14, no. 11, pp. 775–788, 2002.

[92] F. Dressler and O. B. Akan, "A survey on bio-inspired networking," *Computer networks*, vol. 54, no. 6, pp. 881–900, 2010.

[93] C.-L. Fok, G.-C. Roman, and C. Lu, "Agilla: A mobile agent middleware for self-adaptive wireless sensor networks," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 4, no. 3, p. 16, 2009.

[94] T. Liu and M. Martonosi, "Impala: A middleware system for managing autonomic, parallel sensor systems," in *ACM SIGPLAN Notices*, vol. 38, pp. 107–118, ACM, 2003.

[95] Y. Kwon, S. Sundresh, K. Mechitov, and G. Agha, "Actornet: An actor platform for wireless sensor networks," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 1297–1300, ACM, 2006.

[96] V. Terziyan, O. Kaykova, and D. Zhovtobryukh, "Ubiroad: Semantic middleware for context-aware smart road environments," in *Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on*, pp. 295–302, IEEE, 2010.

[97] F. Aiello, G. Fortino, A. Guerrieri, and R. Gravina, "Maps: a mobile agent platform for wsns based on java sun spots," *Proceedings of the ATSN*, 2009.

[98] R. Lopes, F. Assis, and C. Montez, "Maspot: A mobile agent system for sun spot," in *2011 Tenth International Symposium on Autonomous Decentralized Systems*, pp. 25–31, IEEE, 2011.

[99] F. Aiello, G. Fortino, S. Galzarano, and A. Vittorioso, "Tinymaps: a lightweight java-based mobile agent system for wireless sensor networks," in *Intelligent distributed computing V*, pp. 161–170, Springer, 2011.

[100] O. Khriyenko and M. Nagy, "Semantic web-driven agentbased ecosystem for linked data and services," in *Proceedings of the Third International Conferences on Advanced Service Computing*, pp. 25–30, 2011.

[101] J. M. Portocarrero, F. C. Delicato, P. F. Pires, T. C. Rodrigues, and T. V. Batista, "Samson: self-adaptive middleware for wireless sensor networks," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pp. 1315–1322, ACM, 2016.

[102] A. Computing *et al.*, "An architectural blueprint for autonomic computing," *IBM White Paper*, vol. 31, 2006.

[103] C. Sarkar, A. U. N. SN, R. V. Prasad, A. Rahim, R. Neisse, and G. Baldini, "Diat: a scalable distributed architecture for iot," *IEEE Internet of Things journal*, vol. 2, no. 3, pp. 230–239, 2015.

[104] K. Lingaraj, R. V. Biradar, and V. Patil, "Eagilla: An enhanced mobile agent middleware for wireless sensor networks," *Alexandria Engineering Journal*, 2017.

[105] M. Haghighi and D. Cliff, "Multi-agent support for multiple concurrent applications and dynamic data-gathering in wireless sensor networks," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, pp. 320–325, IEEE, 2013.

[106] P. Boonma and J. Suzuki, "Bisnet: A biologically-inspired middleware architecture for self-managing wireless sensor networks," *Computer networks*, vol. 51, no. 16, pp. 4599–4616, 2007.

[107] N. M. do Nascimento and C. J. P. de Lucena, "Fiot: An agent-based framework for self-adaptive and self-organizing applications based on the internet of things," *Information Sciences*, vol. 378, pp. 161–176, 2017.

[108] H. O. Al-Sakran *et al.*, "Intelligent traffic information system based on integration of internet of things and agent technology," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 6, no. 2, pp. 37–43, 2015.

[109] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: an acquisitional query processing system for sensor networks," *ACM Transactions on database systems (TODS)*, vol. 30, no. 1, pp. 122–173, 2005.

[110] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko, *et al.*, "Openiot: Open source internet-of-things in the cloud," in *Interoperability and open-source solutions for the internet of things*, pp. 13–25, Springer, 2015.

[111] C. Chen and S. Helal, "Sifting through the jungle of sensor standards," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 84–88, 2008.

[112] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[113] Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric internet of things," *Journal of Network and Computer Applications*, vol. 64, pp. 137–153, 2016.

[114] F. Ganz, P. Barnaghi, and F. Carrez, "Automated semantic knowledge acquisition from sensor data," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1214–1225, 2014.

[115] G. Yang, S. Li, X. Xu, H. Dai, and Z. Yang, "Precision-enhanced and encryption-mixed privacy-preserving data aggregation in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 4, p. 427275, 2013.

[116] M. M. Fouad, V. Snasel, and A. E. Hassanien, "Energy-aware sink node localization algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 7, p. 810356, 2015.

[117] O. Vermesan, P. Friess, *et al.*, *Internet of things-from research and innovation to market deployment*, vol. 29. River publishers Aalborg, 2014.

[118] M. Botts, A. Robin, J. Greenwood, and D. Wesloh, "Ogc® sensorml: Model and xml encoding standard," *Technical Standard*, vol. 2, 2014.

[119] S. Cox *et al.*, "Observations and measurements-xml implementation," *OGC document*, 2011.

[120] E. Y. Song and K. Lee, "Understanding IEEE 1451-networked smart transducer interface standard-what is a smart transducer?," *Instrumentation & Measurement Magazine, IEEE*, vol. 11, no. 2, pp. 11–17, 2008.

[121] A. U. Limited, "Amon v3.1."

[122] J. C. et. al., "Media types for sensor measurement lists (senml)." `https://tools.ietf.org/html/draft-ietf-core-senml-05`, (Accessed on March, 2019).

[123] S. Matsumoto, "Echonet: A home network standard," *IEEE Pervasive computing*, vol. 9, no. 3, pp. 88–92, 2010.

[124] A. Eisma, "Data capture in ibm websphere premises server™," *OSGi Alliance*, vol. 28, no. 2011, pp. 82–94, 2008.

[125] S. Lindo, "Xml vs. json - a primer." `http://www.programmableweb.com/news/xml-vs.-json-primer/how-to/2013/11/07`, Nov, 2013 (accessed: November, 2015).

[126] G. Bender, "Xml vs json based web services: Which is the best choice?." `http://www.seguetech.com/blog/2013/04/01/xml-vs-json-web-services-best-choice`, Mar, 2013 (accessed: November, 2015).

[127] P. Nappey, C. El Kaed, A. W. Colombo, J. Eliasson, A. Kruglyak, R. Kyusakov, C. Hübner, T. Bangemann, and O. Carlsson, "Migration of a legacy plant lubrication system to soa," in *Industrial Cloud-Based Cyber-Physical Systems*, pp. 167–182, Springer, 2014.

[128] R. Eastman, C. Schlenoff, S. Balakirsky, and T. Hong, "A sensor ontology literature," 2013.

[129] D. Fensel, A. J. Hendler, and J. Domingue, "Introduction to the Semantic Web Technologies," pp. 1–28, 2012.

[130] T. Berners-lee and J. Hendler, "The Semantic Web," *Scientific American*, vol. 21, no. May 17, pp. 29–37, 2001.

[131] I. Jurisica, J. Mylopoulos, and E. Yu, "Ontologies for Knowledge Management: An Information Systems Perspective," *Knowledge and Information Systems*, vol. 6, pp. 380–401, Mar. 2004.

[132] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge Engineering : Principles and methods," vol. 25, pp. 161–197, 1998.

[133] S. Szewcyzk, K. Dwan, B. Minor, B. Swedlove, and D. Cook, "Annotating smart environment sensor data for activity learning," *Technology and Health Care*, vol. 17, no. 3, p. 161, 2009.

[134] W. Wei and P. Barnaghi, "Semantic annotation and reasoning for sensor data," in *Smart Sensing and Context*, pp. 66–76, Springer, 2009.

[135] A. Sheth, "Citizen sensing, social signals, and enriching human experience," *IEEE Internet Computing*, no. 4, pp. 87–92, 2009.

[136] M. C. Machado, D. Rebholz-Schuhmann, T. Freitas, and M. F. Couto, "The semantic web in translational medicine: current applications and future directions.," *Briefings in bioinformatics*, 11 2013.

[137] v. F. Harmelen, F. P. Patel-Schneider, and I. Horrocks, "Reference description of the daml+ oil (march 2001) ontology markup language," *DAML+ OIL Document, URL http://www.daml.org/2000/12/reference.html*, 2001.

[138] Dave, B. , and Brian M., "RDF/XML syntax specification (revised)," *W3C recommendation*, vol. 10, 2004.

[139] e. a. Dean, M., "OWL Web Ontology Language 1.0 Reference, W3C Working Draft 29 July 2002," *Latest version is available at http://www.w3.org/TR/owl-ref*, 2002.

[140] O. W. Group *et al.*, "Owl 2 web ontology language: document overview. w3c recommendation (27 october 2009)," 2012.

[141] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, "The SSN ontology of the W3c semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, Dec. 2012.

[142] K. J. Witt, J. Stanley, D. Smithbauer, D. Mandl, V. Ly, A. Underbrink, and M. Metheny, "Enabling sensor webs by utilizing swamo for autonomous operations," in *8th NASA Earth Science Technology Conference*, 2008.

[143] C. Goodwin and D. J. Russomanno, "An ontology-based sensor network prototype environment," in *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks*, pp. 1–2, 2006.

[144] K. Togias, C. Goumopoulos, and A. Kameas, "Ontology-based representation of upnp devices and services for dynamic context-aware ubiquitous computing applications," in *2010 Third International Conference on Communication Theory, Reliability, and Quality of Service*, pp. 220–225, IEEE, 2010.

[145] F. G. TC, "Fipa device ontology specification," 2001.

[146] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko, *et al.*, "Openiot: Open source

internet-of-things in the cloud," in *Interoperability and open-source solutions for the internet of things*, pp. 13–25, Springer, 2015.

[147] A. Ilari Maarala, X. Su, and J. Riekki, "Semantic reasoning for context-aware internet of things applications," *arXiv e-prints*, pp. arXiv–1604, 2016.

[148] N. R. Roy and P. Chandra, "Analysis of data aggregation techniques in wsn," in *International conference on innovative computing and communications*, pp. 571–581, Springer, 2020.

[149] H. M. Al-Kadhim and H. S. Al-Raweshidy, "Energy efficient data compression in cloud based iot," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 12212–12219, 2021.

[150] B. Srbinovski, M. Magno, F. Edwards-Murphy, V. Pakrashi, and E. Popovici, "An energy aware adaptive sampling algorithm for energy harvesting wsn with energy hungry sensors," *Sensors*, vol. 16, no. 4, p. 448, 2016.

[151] L. C. Monteiro, F. C. Delicato, L. Pirmez, P. F. Pires, and C. Miceli, "Dpcas: Data prediction with cubic adaptive sampling for wireless sensor networks," in *International Conference on Green, Pervasive, and Cloud Computing*, pp. 353–368, Springer, 2017.

[152] H. Wang, Z. Yemeni, W. M. Ismael, A. Hawbani, and S. H. Alsamhi, "A reliable and energy efficient dual prediction data reduction approach to wsns based on kalman filter," *IET Communications*, 2021.

[153] M. Lewandowski and B. Płaczek, "Data transmission reduction in wireless sensor network for spatial event detection," *Sensors*, vol. 21, no. 21, p. 7256, 2021.

[154] G. B. Tayeh, A. Makhoul, C. Perera, and J. Demerjian, "A spatial-temporal correlation approach for data reduction in cluster-based sensor networks," *IEEE Access*, vol. 7, pp. 50669–50680, 2019.

[155] S. Samarah, M. Al-Hajri, and A. Boukerche, "A predictive energy-efficient technique to support object-tracking sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 656–663, 2010.

[156] M. Alam, A. A. Aziz, S. Latif, and A. Awang, "Error-aware data clustering for in-network data reduction in wireless sensor networks," *Sensors*, vol. 20, no. 4, p. 1011, 2020.

[157] Y. Fathy, P. Barnaghi, and R. Tafazolli, "An adaptive method for data reduction in the internet of things," in *2018 IEEE 4th World Forum on Internet of things (WF-IoT)*, pp. 729–735, IEEE, 2018.

[158] X.-Y. Liu, Y. Zhu, L. Kong, C. Liu, Y. Gu, A. V. Vasilakos, and M.-Y. Wu, "Cdc: Compressive data collection for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2188–2197, 2014.

[159] N. Ahmed and K. R. Rao, *Orthogonal transforms for digital signal processing*. Springer Science & Business Media, 2012.

[160] R. E. Blahut, *Fast algorithms for signal processing*. Cambridge University Press, 2010.

[161] Z.-c. Yang, "Predictive modeling of hourly water-level fluctuations based on the dct least-squares extended model," *Water resources management*, vol. 32, no. 3, pp. 1117–1131, 2018.

[162] Z.-C. Yang, "Discrete cosine transform-based predictive model extended in the least-squares sense for hourly load forecasting," *IET Generation, Transmission & Distribution*, vol. 10, no. 15, pp. 3930–3939, 2016.

[163] Y. Zong-chang, "Fourier analysis-based air temperature movement analysis and forecast," *IET Signal Processing*, vol. 7, no. 1, pp. 14–24, 2013.

[164] Z.-C. Yang, "Eigen-temperature model for the annual air temperature movement evaluation and forecast," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 4, no. 03, p. 1350008, 2013.

[165] M. Crawford, "7 creative applications of solar energy." `https://www.asme.org/topics-resources/content/7-creative-applications-of-solar-energy`, 2020 (Accessed on October 4, 2021).

[166] Z.-C. Yang, "Modeling and forecasting monthly movement of annual average solar insolation based on the least-squares fourier-model," *Energy conversion and management*, vol. 81, pp. 201–210, 2014.

[167] Z.-C. Yang, "Electric load movement forecasting based on the dft interpolation with periodic extension," *Journal of Circuits, Systems and Computers*, vol. 24, no. 08, p. 1550123, 2015.

[168] Z.-c. Yang, "Electric load movement evaluation and forecasting based on the fourier-series model extend in the least-squares sense," *Journal of Control, Automation and Electrical Systems*, vol. 26, no. 4, pp. 430–440, 2015.

[169] Z.-c. Yang, "The least-square fourier-series model-based evaluation and forecasting of monthly average water-levels," *Environmental Earth Sciences*, vol. 77, no. 9, pp. 1–11, 2018.

[170] P. Jain and A. Jain, "Regressive structures for computation of dst-ii and its inverse," *International Scholarly Research Notices*, vol. 2012, 2012.

[171] H. Kekre, T. Sarode, and P. N. Halarnkar, "Partial image scrambling using walsh sequency in sinusoidal wavelet transform domain," in *Intelligent Systems Technologies and Applications*, pp. 471–484, Springer, 2016.

[172] J. Panda and S. Meher, "A novel approach of image interpolation using dst," in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, pp. 606–611, IEEE, 2019.

[173] M. T. Nguyen and K. A. Teague, "Energy-efficient data collection in clustered wireless sensor networks employing distributed dct," *International Journal of Wireless & Mobile Networks (IJWMN) Vol*, vol. 8, 2016.

[174] Z.-c. Yang, "Predictive modeling of hourly water-level fluctuations based on the dct least-squares extended model," *Water resources management*, vol. 32, no. 3, pp. 1117–1131, 2018.

[175] M. Püschel and J. M. Moura, "Algebraic signal processing theory," *arXiv preprint cs/0612077*, 2006.

[176] K. Jones, "The regularized fast hartley transform," 2010.

[177] K. R. Rao and P. Yip, *Discrete cosine transform: algorithms, advantages, applications*. Academic press, 2014.

[178] P. K. Meher, T. Srikanthan, and J. C. Patra, "Scalable and modular memory-based systolic architectures for discrete hartley transform," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 5, pp. 1065–1077, 2006.

[179] V. A. Coutinho, R. J. Cintra, and F. M. Bayer, "Low-complexity three-dimensional discrete hartley transform approximations for medical image compression," *Computers in biology and medicine*, vol. 139, p. 105018, 2021.

[180] X. Zhang and Q. Su, "A spatial domain-based color image blind watermarking scheme integrating multilevel discrete hartley transform," *International Journal of Intelligent Systems*, vol. 36, no. 8, pp. 4321–4345, 2021.

[181] R. K. Rdwan and M. M. Amer, "Fingerprint matching based on two-dimensional discrete hartley transform," 2021.

[182] K. Narendra and S. Satyanarayana, "Hartley transform based correlation filters for face recognition," in *2016 International Conference on Signal Processing and Communications (SPCOM)*, pp. 1–5, IEEE, 2016.

[183] S. D. Thepade and M. M. Kalbhor, "Ensemble of machine learning classifiers for improved image category prediction using fractional coefficients of hartley and sine transforms," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–5, IEEE, 2018.

[184] S. Nakhate *et al.*, "Fast hartley transform based elliptic curve cryptography for resource constrained devices," in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 71–76, IEEE, 2020.

[185] P. Barnaghi, S. Meissner, M. Presser, and K. Moessner, "Sense and sens' ability: Semantic data modelling for sensor networks," in *Conference Proceedings of ICT Mobile Summit 2009*, 2009.

[186] I. Khan, R. Jafrin, F. Z. Errounda, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "A data annotation architecture for semantic applications in virtualized wireless sensor networks," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 27–35, IEEE, 2015.

[187] M. Lewis, D. Cameron, S. Xie, and B. Arpinar, "Es3n: A semantic approach to data management in sensor networks," in *Semantic Sensor Networks Workshop*, 2006.

[188] A. Zafeiropoulos, N. Konstantinou, S. Arkoulis, D.-E. Spanos, and N. Mitrou, "A semantic-based architecture for sensor data fusion," in *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM'08. The Second International Conference on*, pp. 116–121, IEEE, 2008.

[189] V. Huang and M. K. Javed, "Semantic sensor information description and processing," in *Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on*, pp. 456–461, IEEE, 2008.

[190] S. Krco, M. Johansson, and V. Tsiatsis, "A commonsense approach to real-world global sensing," in *Proceedings of the SenselD: Convergence of RFID and Wireless Sensor Networks and their Applications workshop, ACM SenSys*, 2007.

[191] C. A. Henson, J. K. Pschorr, A. P. Sheth, and K. Thirunarayan, "Semsos: Semantic sensor observation service," in *Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on*, pp. 44–53, IEEE, 2009.

[192] W. M. Trochim and J. P. Donnelly, "Research methods knowledge base," 2001.

[193] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 126–137, ACM, 2003.

[194] J. Polley, D. Blazakis, J. McGee, D. Rusk, and J. S. Baras, "Atemu: a fine-grained sensor network simulator," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 145–152, IEEE, 2004.

[195] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pp. 455–462, IEEE, 2004.

[196] J. L. Peterson, "Petri net theory and the modeling of systems," 1981.

[197] A. Moschitta and I. Neri, "Power consumption assessment in wireless sensor networks," *ICT-Energy-Concepts Towards Zero-Power Information and Communication Technology*, 2014.

[198] A. Ageev, D. Macii, and D. Petri, "Experimental characterization of communication latencies in wireless sensor networks," in *Proc. 16th Intl. Symp. Exploring New Frontiers of Instrumentation and Methods for Electrical and Electronic Measurements (IMEKO TC4)*, pp. 258–263, 2008.

[199] C. B. Margi, V. Petkov, K. Obraczka, and R. Manduchi, "Characterizing energy consumption in a visual sensor network testbed," in *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006.*, pp. 8–pp, IEEE, 2006.

[200] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance measurements of motes sensor networks," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pp. 174–181, ACM, 2004.

[201] D. Fensel, H. Lausen, A. Polleres, J. De Bruijn, M. Stollberg, D. Roman, and J. Domingue, *Enabling semantic web services: the web service modeling ontology*. Springer Science & Business Media, 2006.

[202] G. Eleftherakis, D. Pappas, T. Lagkas, K. Rousis, and O. Paunovski, "Architecting the IoT paradigm: a middleware for autonomous distributed sensor networks," *Inter J Dist Sens Net*, vol. 11, no. 12, p. 139735, 2015.

[203] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[204] Y. Ding, H. Sun, and K. Hao, "A bio-inspired emergent system for intelligent web service composition and management," *Knowledge-Based Systems*, vol. 20, no. 5, pp. 457–465, 2007.

[205] G. Eleftherakis, O. Paunovski, K. Rousis, and A. Cowling, "Emergent Distributed Bio-organization: A Framework for Achieving Emergent Properties in Unstructured Distributed Systems," in *Intelligent Distributed Computing VI*, pp. 23–28, Springer, 2013.

[206] T. R. Burchfield, S. Venkatesan, and D. Weiner, "Maximizing throughput in zigbee wireless networks through analysis, simulations and implementations," in *Proc. Int. Workshop Localized Algor. Protocols WSNs*, pp. 15–29, 2007.

[207] G. M. Phillips, *Interpolation and approximation by polynomials*, vol. 14. Springer Science & Business Media, 2003.

[208] H. Harb, C. A. Jaoude, and A. Makhoul, "An energy-efficient data prediction and processing approach for the internet of things and sensing based applications," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 780–795, 2020.

[209] W. B. Heinzelman, *Application-specific protocol architectures for wireless networks*. PhD thesis, Massachusetts Institute of Technology, 2000.

[210] C. District, "Beach weather stations - automated sensors.." `https://data.cityofchicago.org/Parks-Recreation/Beach-Weather-Stations-Automated-Sensors/k7hf-8y75`, 2019 (Accessed on April 4, 2022).

[211] R. Kaluri, D. S. Rajput, Q. Xin, K. Lakshmanna, S. Bhattacharya, T. R. Gadekallu, and P. K. R. Maddikunta, "Roughsets-based approach for predicting battery life in iot," *arXiv preprint arXiv:2102.06026*, 2021.

[212] Z.-c. Yang, "Predictive modeling of hourly water-level fluctuations based on the dct least-squares extended model," *Water resources management*, vol. 32, no. 3, pp. 1117–1131, 2018.

Appendix A - Describing sensor information with SenML

```
{"e":
[
{ "n": "temperature", "t": 15364,"v":36.5, "u":"degC" },
{ "n": "glucose", "t": 15364, "v":36.5, "u": " mmol/L"}
]
}
107 bytes
```

Appendix B - Describing sensor information with SensorML

```
<sml:inputs>
        <sml:InputList>
            <sml:input name="body temperature">
                <sml:ObservableProperty definition="http://url_link.com/something/
                temperature.owl#Temperature/">
            </sml:input>
            <sml:input name="glucose">
                <sml:ObservableProperty definition="http://url_link.com/something/
                glucose.owl#glucose"/>
            </sml:input>
        </sml:InputList>
    </sml:inputs>
   <!-- ================ Outputs = Quantities================== -->
   <sml:outputs>
      <sml:OutputList>
         <sml:output name="the_disease_parameters">
            <swe:DataRecord>
               <swe:field name="body temperature">">
                  <swe:Quantity definition="http://definitionLink.com/ont/swe/
                     property/BodyTemperature">
                     <swe:label>Body Temperature"></swe:label>
                     <swe:uom code="cel"/>
                  </swe:Quantity>
               </swe:field>
               <swe:field name="glucose">
                  <swe:Quantity definition="http://definitionLink.com/ont/swe/
                     property/Glucose">
                     <swe:label>Glucose</swe:label>
                     <swe:uom code="mmol/L"/>
                  </swe:Quantity>
               </swe:field>
            </swe:DataRecord>
```

```
      </sml:output>
   </sml:OutputList>
</sml:outputs>
1100 bytes
```