UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science
Software Engineering
Lukman Oladimeji Adekunle

Predicting Wireless sensor readings with Neural network

Master Thesis (30 EAP)

Supervisor: Satish Narayana Srirama, PhD

TARTU, 2014

Predicting wireless Sensor reading with Neural Network

Abstract

Wireless sensor networks are becoming a part of our daily lives, as they act as a bridge between the physical world and the virtual world.One of the problems encountered by this type of networks while trying to fulfill their goals is the rate of energy consumption. The approach considered in this paper was that of an artificial neural network with the aim of reducing the rate of power consumption and thereby increasing the performance and durability of the network. Support vector machines backed artificial neural model was the best of all models picked. It was then compared with a linear regression model to see if there would be any good reasons to migrate to the this new approach. At the end, it was observed that the chosen network performed slightly above the level of the existing model. The implications of the observed results were that another form of prediction model can replace the existing one or alternated with one another in the process of operation of a wireless sensor network.

Keywords:

Artificial Neural network, Arduino, Support vector machines, NEAT

Närvivõrkude abil traadita seadmete mõõtmistulemuste ennustamine

Sisukokkuvõte

Juhtmevabadest sensorvõrkudest on saamas osa meie igapäevalust. Tegemist on sillaga füüsilise ja virtuaalse maailma vahel. Üheks probleemiks seda laadi võrkude puhul on aga energia tarbimise määr. Käesolevas lõputöös uuriti lähenemist, kus kasutatakse tehisneurovõrke eesmärgiga vähendada energiatarvet ja seeläbi parendada sensorvõrgu efektiivsust ning vastupidavust. Tugivektormasinatega toetatud tehisneuromudel valiti välja kui parim vaatluse all olnud mudel. Seda mudelit võrreldi lineaarse regressiooni mudeliga, et näha kas väljavalitud mudeli puhul leidub mõjuvaid põhjuseid just seda eelistada. Lõpuks selgitati välja, et uuritava mudeli efektiivsus oli veidi kõrgem kui võrreldaval mudelil. Töö tulemustest järeldub, et olemasolevaid ennustusmudeleid võib asendada alternatiivsetega või kasutada neid vaheldumisi juhtmevaba sensorvõrgu töö käigus.

Märksnad:

Tehisneurovõrk, Arduino, Tugivektormasinad, NEAT

Contents

List of Figures

List of Tables

Acknowledgements

I would like to use this medium to thank Almighty God for giving me the capacity and strength to go through my masters degree program. Secondly I want to say a big thank you to my parents and siblings. This thesis would have not been a success without my great Professors , Associate Professors, and Technical Assistants who instilled in me the right discipline and knowledge. A big thank you to the IT Academy program and the Estonian government for their support during my studies.

1

Introduction

1.1 Introduction

Intelligent systems are a common place around the world due to the popularity of internet of things. According to survey the (6) Internet of things (IoT) can be defined as a world-wide network of entities that have a set of standard communication protocol, unique address and reach a common goal of providing useful services. This implies that identifiable heterogeneous systems work together to achieve a common goal of making the world of computing simpler and easier. IoT can be viewed from three different perspectives which includes the things perspective that looks into the field of generic non-homogeneous object integration, Secondly, the internet perspective that caters for the networking aspect where by preexisting IP infrastructure and the adoption of IEEE 802.15.4 protocol can make IoT a reality, since the IP infrastructure already targets many devices.Thirdly, the semantics perspective that looks into the description of data, gathering of reasonable information from data generated by devices connected and the architecture of the environments, as they are executed. It should be noted that the semantics aspect also handles the extreme growth rate of data as they become more challenging. A typical implementation of IoT is sustained by Wireless technologies like Radio-Frequency IDentification (RFID) tags, Wireless Sensor Networks (WSN) and RFID Sensor Networks (RSN). They have increased the accessibility of the services provided by IOT.

Wireless sensor networks can be used to determine the status of entities like location, temperature and movements within its network. They can even be made to be context

aware in order to minimize the gap between the physical and the digital realm. Applications of Wireless sensor network can be found in all aspect of life which includes transportation, Health care, military activities monitoring and controlling industrial plants and sensing of the environment.
Most sensor nodes in a wireless network report the activities going on in their environment to a central location otherwise known as an aggregator in a single or multi hop manner, but are plagued by their computing and energy resources, which makes them not to deliver their full potential. Many researchers have focused on ways on how to improve on these anomalies either by concentrating on improving the battery life or coming up with processors with better speed.
This thesis aims to focus on rectifying the problems concerning energy aspect of a wireless sensor network, the motivation for such studies is discussed in the section below.

1.1.1 Motivation

In the process of coming up with a prototype of wireless sensor network for a house old environment in (7) .Arduino micro controller, sensor hardware and real time XMPP communication between the cloud for example Amazon S2 that act as an aggregator were used. It was observed that there was the need for improvement in the following areas :

- Energy Consumption.
- Data on Demand.
- Scalability of sensor nodes to multiple aggregators.

The author in (8) saw some significant improvement with the battery life when data was requested on demand using linear regression and fuzzy for predicting values when the client is asleep, the prototype still has some area of improvement to be worked upon. Some of these areas include :

- Security and integrity of the data been transmitted.
- Better performing algorithm in the cloud.
- Energy consumption.
- Scalability of sensor nodes to multiple aggregators.

1.1.2 Contributions

Based on the aforementioned possible improvement above this thesis aims to propose the following solutions:

- Extend the current code with a mechanisms to collect features like: battery status, cpu load, bandwidth of network, accelerometer data, gyroscope data and time of when data is collected, in order to classify the data into groups.
- A better algorithm (Neural networks) for estimating the data set features when a client is sent to sleep.
- Monitor the rate of discharge of the battery attached to the microcontroller in the presence and absence of the deciding algorithms (linear regression with fuzzy logic and neural networks).

1.1.3 Outline

Chapter 2: Talks about the state of the art of components used in this thesis. Firstly, the chapter introduces the idea of wireless sensor networks, its type and their respective properties. Secondly, the hardware components that acted as sensor nodes were discussed and finally an introduction to time anaylsis and a possible means of solving means of solving time analysis problems as related to the thesis were discussed.

Chapter 3:Explains the engineering problems plagued with context aware networks that this thesis proposes to solve and talks about the consequence of the solutions.

Chapter 4: Describes the implementation details of the hardware and software components of the prototype.First and foremost, adjustment to the data transferred between the sensor module and the aggregator(cloud) is discussed. The second part, takes an overview of the neural network implementation system for estimation. Finally, the problems encountered during the realisation of the algorithm and hardware extensions were covered in details.

Chapter 5: Presents some real life scenarios with emphasis on the comparison of when the sensor network is allowed to work with and without the prediction logic with neural network and linear regression model in place.

Chapter 6:Talks about the works done by other researchers in the field of wireless sensor network as related to this thesis.

Chapter 7: Concludes with the results achieved in this thesis together with possible future research directions.

2

State of the Art

In this chapter, the author presents a complete description of Wireless sensor networks and its applications, Time series analysis, Cloud computing and finally, focused on describing Arduino microprocessor as an open source electronic platform.

2.1 Wireless Sensor Networks

A wireless sensor network (WSN) is a network of distributed devices that monitor physical and environmental conditions. Naturally, these devices or nodes are made up of radio, battery, micro controllers, analog circuits and sensor interface. The architecture of a WSN always involves the use of a gateway that allows for the connection of both wired and other wirelessly distributed devices. The common topologies supported by this type of network include

- **Star:** All Sensor nodes are connected to the gateway or aggregator directly.
- **Cluster Tree:** This type of arrangement contains node that are indirectly connected to the sink via other nodes in the hierarchy.
- **Mesh:** This is a hybrid of both cluster tree and star topologies.

Operating frequencies for a wireless network are 2.4 GHZ and 5 GHZ. Application requirements of a wireless sensor network determines that protocol standard that is used. Table 2.1 below shows the available protocols, data rates and their short description.

Table 2.1: Wireless protocols

Protocol /Frequency (GHz)	Signal	Maximum Data Rate	Notes
Legacy 802.11 / 2.4	Frequency hopping spread spectrum or direct-sequence spread spectrum (DSSS)	2 Mbps	Originally defined carrier sense multiple access with collision avoidance (CSMA-CA). Three non-overlapping channels in industrial, scientific, medical (ISM) frequency band at 2.4 GHz
802.11a / 5	Orthogonal frequency-division multiplexing	54 Mbps	12 non-overlapping unlicensed national information infrastructure (UNII) channels in 5 GHz frequency band. Data rates with varying modulation types: 6, 9, 12, 18, 24, 36, 48 and 54 Mbps
802.11b / 2.4	Frequency hopping spread spectrum or direct-sequence spread spectrum (DSSS)	11 Mbps	Three non-overlapping channels in industrial, scientific, medical (ISM) frequency band at 2.4 GHz
802.11g / 2.4	Orthogonal frequency-division multiplexing	54 Mbps	Three non-overlapping channels in industrial, scientific, medical (ISM) frequency band at 2.4 GHz.Data rates with varying modulation types: 6, 9, 12, 18, 24, 36, 48 and 54 Mbps; can revert to 1, 2, 5.5, and 11 Mbps using DSSS and CCK
802.11n / 2.4 or 5	Orthogonal frequency-division multiplexing	600 Mbps (theoretical)	12 non-overlapping unlicensed national information infrastructure (UNII) channels in 5 GHz frequency band with and without CB.Orthogonal frequency-division multiplexing (OFDM) using multiple-input/multiple-output (MIMO) and channel bonding (CB)

2.2 Arduino

Sensor networks are based on some hardware and software stack which can be mimicked with an Arduino device. Arduino(1) is an open source micro controller, made from a combination of hardware and software parts. It has an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM, a USB interface, 6 analog input pins and 14 digital input/out pins. These input/output pins allow the board to be extended in terms of computing capacity.Arduino's software part is built using Java and C++ programming language. Arduino Uno,Leonardo, Yun, Mega ADK are good examples of the arduino based boards. It should be noted that Arduino Mega ADk is being used as one of the building blocks of a sensor network node used in this work. Generally, arduino has some extra hardware components that are called shields. when they are attached to the arduino base board they extend the capability of the arduino board making it easy for sensors to be integrated. Examples of arduino shields include Arduino Ethernet, WiFi, Wireless, Motor and Wireless Proto shield. Arduino is mainly used by hobbyist, developers as well as researchers for quick prototyping of mini electronic circuits that capable of interacting and sensing with the environment. Embedded computing devices like Raspberry Pi and Beagle bone black also serve as possible substitute for arduino. A sample of the arduino base board showing the bare-bones component is shown in the Figure 2.1 below.

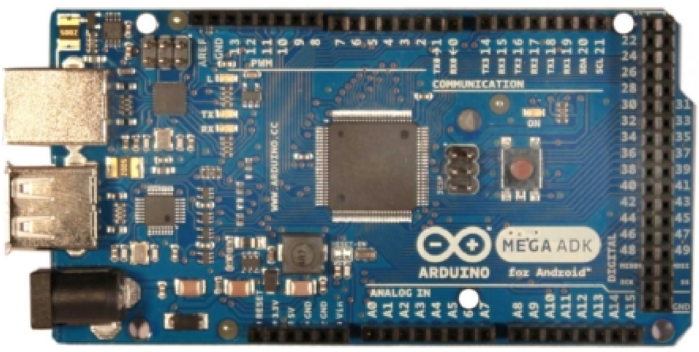

Figure 2.1: Arduino Base board.(1)

2.2.1 Arduino software

In order for Arduino hardware to function properly and effectively, its activities needs to be coordinated by the software component. Programs that are run on the hardware are written in arduino language which is is based on C/C++ programming language . There is an integrated development environment (IDE) that makes it possible to write codes for the boards. To come up with a simple program that would run on an arduino board implies writing and saving the code within the IDE, after which it is transferred to board via a USB cable. The output of the board can be monitored by setting the necessary parameters like baud rate in the monitor screen. Figure 2.2 shows an arduino IDE with a "voltagesourceneasure" source codes and its message pane which has a black background below the source code. The monitor screen that ouputs the messages from the arduino controller overlays the ide.

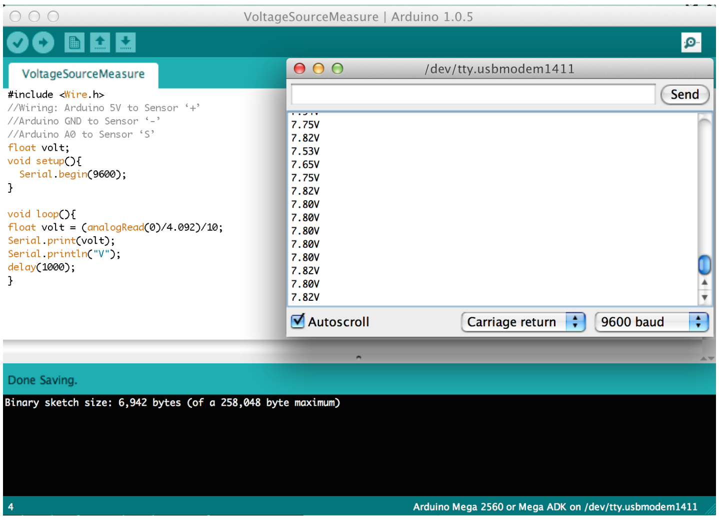

Figure 2.2: Arduino IDE.

2.2.2 Arduino Sample source code

Arduino source code popularly known as a sketch in the arduino world, is structured in a very simple manner. The main constituent of a sketch include the header ,loop and setup blocks. The header part is used to declare libraries needed within the sketch an example is the declaration of ⟨ ♯include Wire.h ⟩ in the sketch below. The setup block allows for one-time configuration of arduino parameters since it is run once during sketch's lifecycle. The loop block is area of the sketch that is constantly. whatever job that the arduino is needed to undertake is placed in the loop method A sample sketch that mesures the voltage source is show below in Listing 1. As shown in the listing, the sketch includes a library called "wire.h", declares a global variable called volt of type float, has a setup loop to initialise the serial port with a baud rate of 9600 bits/secs and finally gets the volatge by reading an analog input pin and then carrying out computations on the output value which then displayed.

```
 #include <Wire.h>
/**This skecth reports  the voltage mesured from the power
source to the serial monitor
**/
float volt;
void setup(){
  Serial.begin(9600);
}
void loop(){
        volt = (analogRead(0)/4.092)/10;
        Serial.print(volt);
        Serial.println("V");
        delay(1000);
}
```

Listing 1: VoltageSourceMeasure.ino

2.2.3 Arduino Shields

Shields and Modules serves as a point of interest for extending the functionality of an arduino board. They make building of complex circuits easy, portable and scalable. Shields are attached to the 14 (A0 - A13) digital input and out pin on the arduino boards. examples of shields available from the arduino company are Wireless, Ethernet and Motor shields. The arduino board is also compatible with shields from other manufacturers like Bitwizard, TiisaiDipJp and SainSmart.

2.2.3.1 Network Shields

Inter-connectivity plays a major role in the internet of things arduino acting as wireless sensor node has shields that makes it possible. Thus, this shields are explained below:

- **Ethernet Shield:** This shield allows for the communication and connection between an arduino board and internet routers , computers and switches. Connection of about 10/100 Mbs is possible using a CAT5 or CAT6 cable that is attached to both an RJ45 jack on the arduino shield end and the external device. Wiznet W5100 chipset is used to add both TCP and UDP capabilities to the shield enabling four simultaneous socket connections.On the software end, there is an Ethernet network library that gives developers the opportunity to access the arduino's base functionality. Files that need to be transferred over the network to other locations can be placed on a memory SD stick which is later placed in slot available on the board. The ethernet shield also has a reset controller, that ensures that the W5100 Ethernet module is reset when powered on. An Image of an ethernet shield is shown in Figure 2.3 below.

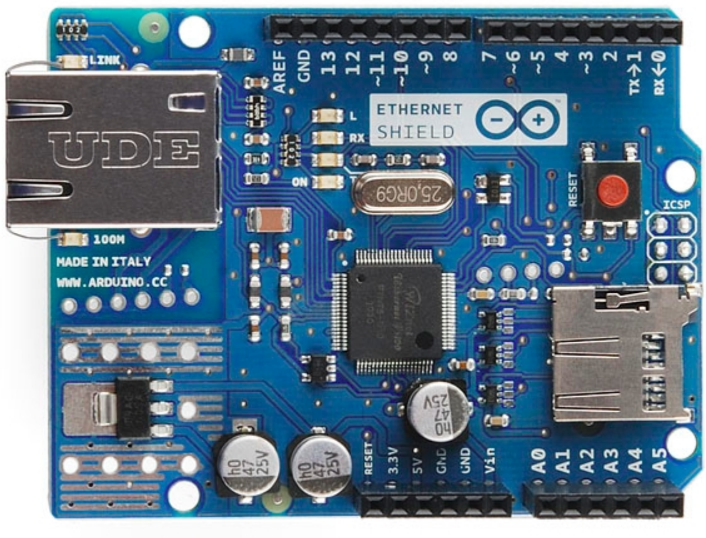

Figure 2.3: Arduino Ethernet Shield.

- **Wireless SD Shield:** This shield allows an Arduino board to communicate wirelessly by using a wireless sensor module on top of it. Xbee modules from Digi is the main constituent of the board. Its communication range with a line of sight is around 30 meters for indoor usage and approximately three times its indoor range for outdoor usage. It can be used in a command or USB replacement mode for a configuration of a variety of broadcast and mesh networking options. On the board there is an SD card slot. The board has a serial switch that is used to change the mode of communication between itself and the arduino main board. When the switch is in the micro mode , the data sent from the module is transmitted to the computer via USB(if connected) and to a remote device wirelessly simultaneously. It should be noted that the microcontroller will not be programmable via USB in this mode. On the other hand, when the switch is in the USB mode, the module can communicate directly with the computer , also bypassing the arduino base board. To program the wireless shield efficiently the board should be loaded with the BareMinimum.io before any other sketch is loaded on the board.

2.2.3.2 Mega Sensor Shield

This type of shield is used to attach TinkerKit sensors and actuators directly to the Arduino base board, without the use of a breadboard, thus making the attachment of sensors to arduino easier. The shield has 12 standard Tinker Kit 3pin connectors. Analog Inputs are labeled I0 to I5. while the analog outputs are connected to the pulse width modulation capable outputs of the arduino Board are labeled from O0 through O5. It should be noted that the analog output can be changed to Digital Inputs, in which they will report their state either as HIGH or LOW. A green LED on the board is used to show that the board is powered properly. The board also has a push button that is used for reset. Figure 2.4 shows what a typical mega sensor shield looks like.

2.2.4 Arduino Sensors

- **Accelerometer:** An accelerometer is an electro-mechanical device that senses the inertial reaction of a proof mass so as to measure linear or angular acceleration (9). This type of acceleration can be tracked statically or dynamically along the X, Y and Z axis. When measuring the acceleration statically the angle with which the device is tilted with respect to the earth can be known.Measuring the dynamic acceleration one can analyze the way the device containing and accelerometer is moving. A typical application of the device include image orientation in mobile

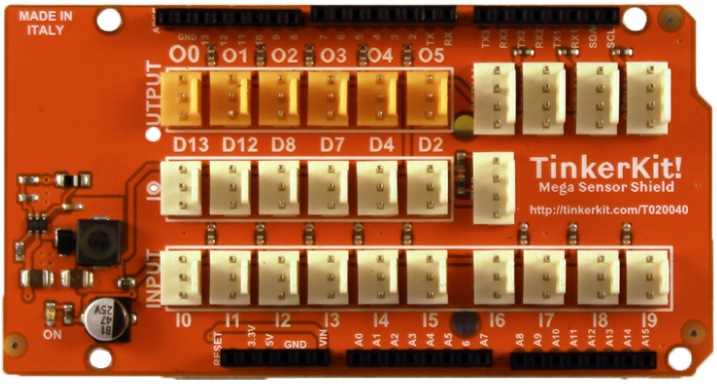

Figure 2.4: Tinkerkit Mega Sensor Shield. (2)

devices, input in controllers and smart phones. The accelerometer module provided by the Tinkerkit can output 0V to 5V on any of its two signal pins when its G acceleration-force is changed . An output voltage of approximately 2.5V can be read when there is zero G on the X or Y axis.A value within the range of 0 to 1023 can be read When the acelerometer module is moving and connected to the arduino base via tinkerkit mega shield. The major component of the module is LIS344AL by ST Microelectronics.

- **Gyroscope:**. In it simplest form a gyroscope is a device that uses angular momentum to sense angular motion of its container with respect to inertial space about one or two axes orthogonal to the spin axis (9). Simply put, a gyroscope measures the rate or rotation around a device's x, y, and z axis. Its unit of measure is in radians per second (rad/s). A possible application of a gyroscope include facial and gesture recognition . The core module of the Tinkerkit gyroscope is LPR5150AL by ST Microelectronics, which is a two-axis gyroscope. The module gives out the same ouput value like the accelerometer mentioned above.

- **Light Sensor:** A light sensor also know as a photoresistor, is a type of resistor whose resistivity is a function of the incident electromagnetic radiation, which makes it a light sensitive device. Its resistance reduces with increasing incident light intensity. The value of the resistance vary from Mega Ohms when there is no light incident on it to a value of about a hundred ohms when exposed to light rays. Arduino library expects an input value within the range of 0 to 1023 ohms to a digital value of either 0 volts when there is no light incident on the resistor or 5 volts when there is a light respectively.

- **Thermistor:** A Thermistor is an electromechanical device that is designed to measure temperature of its environment. It decreases the resistance of the circuit as the temperature rises and increases the temperature vice-versa. Output reading from the device is between 0 and 1023, which can be later converted to Celsius and Fahrenheit degree values.

2.2.5 Arduino WiFly Wireless Module

The WiFly wireless module is an embedded wireless device. The device is based on the TCP/IP stack. In its simplest form the hardware configuration requires only the power, Transmit, Receiver, and ground pins. Once a sketch is loaded, the device can automatically accesses a WiFi network thereby sending and receiving data serially. It also allows for the migration from 802.15.4 to the proper 802.11b/g network without any hardware configuration. which means that it can be used inplace of a XBee hardware that is normally used with an arduino board.

2.3 Time Series

A time series is a set of data, usually collected at regular intervals within a time period. Time series data are naturally evident in our every day lives. examples include Scanning of brain wave activity every 2 to 8 secs with an Electrocardiogram machine in the field of Medicine and measuring the amount of rainfall or humidity daily, weekly or monthly rainfall in the field of Meteorology. Its usefulness is not only limited to a few aspects of life like econometrics, finance, recognition of patterns, communications engineering, astronomy, and control engineering. Time series has been found to useful in making predictions and forecasting the future like earthquake prediction and whether forecasting. A simple diagram showing the what a time series looks like is shown below in figure 2.5 below

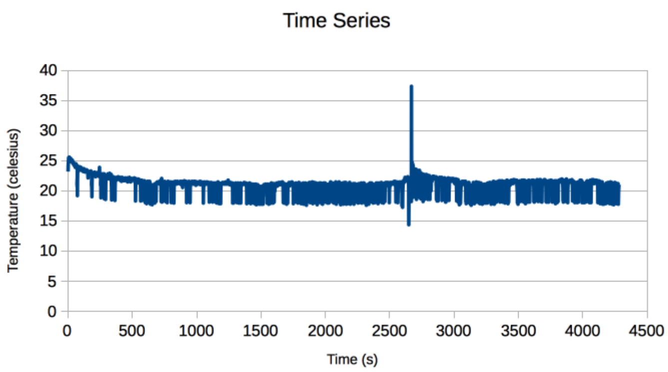

Figure 2.5: Sample Temperature time series.(3)

2.4 Time Series Analysis

Time series analysis uses established methods for examining and analyzing time series data sets in order to extract information and other properties from a data set.
Time series classification or forecasting makes use of a model that is an output of the training on past data set to predict future values. Machine learning can be incorporated into time series analysis to predict and classify variables related to time series . Machine learning approaches used to solve time series problem include Regression analysis, Support vector machines, Fuzzy Logic e.t.c .

2.4.1 Machine Learning

Machine learning a well known branch of artificial intelligence,it has to do with the study and construction of systems, based on past experience in the form of data. Machine learning can be used to teach computers how to evolve on their own without being programmed. Machine learning systems are capable of generalising and representing the data they act upon. In the process of representation, functions and the data instances are evaluated. Generalisation on the other hand makes it possible for the system to function at optimal rate when a new data sets are introduced to the system. Areas of life benefiting from machine learning include Image recognition, Cheminformatics, Bio informatics , computational finance and search engines. In order to solve big data problems many approaches like fuzzy logic, Bayesian networks, artificial neural networks, clustering and Support vector machines can be used.

2.4.2 Artificial Neural Network

Artificial neural networks (ANNs) are mathematical models capable of solving machine learning and pattern recognition problems. An ANN is inspired by the human nervous system i.e how the brain functions based on neurons. An ANN resembles the brain in two respects: 1. The network uses a learning process is to gain Knowledge. 2.Synaptic weights are used to store knowledge (10). They are naturally depicted as systems of interconnected "neurons" that can predict output values from inputs and possibly former outputs values. In its simplest form a neural network it consist of inputs, each the line connecting these inputs to the neuron is assigned a weight. Weights in the model correspond to the synaptic connections in biological neurons. The threshold in ANN is denoted by θ and its corresponding activation is given by the formula.

$$a = \sum_{i=1}^{N} w_i u_i + \theta \tag{2.1}$$

The final output value x of the neuron is a function of its activation "a" which is given by

$$x = f(a) \tag{2.2}$$

Some of the functions that can be used include binary threshold, linear threshold, sigmoid, hyperbolic tan and Gaussian.

2.4.3 Fuzzy Logic

This is one of the machine learning tools for analyzing a time series (11). Fuzzy logic is a multiple-valued logic, It seeks to know how likely a dataset is approximate rather than fixed and exact (12). Compared to traditional binary sets (where variables may take on true or false values) fuzzy logic variables may have a truth value that ranges in degree between 0 and 1.

Fuzzy logic systems as defined in (4) is a nonlinear mapper of an input data to a scalar output data.The form of input could take a vector format.A fuzzy logic system comprises of a fuzzifier, defuzzier, fuzzy rules, and an inference engine as shown in the Figure 2.6.

The fuzzifier's job is to engage in what is called fuzzification. Fuzzification allows the system inputs to have a corresponding linguistic term. For example assume the temperature of a room is 20°C this value can be mapped to a human term of "GOOD" out

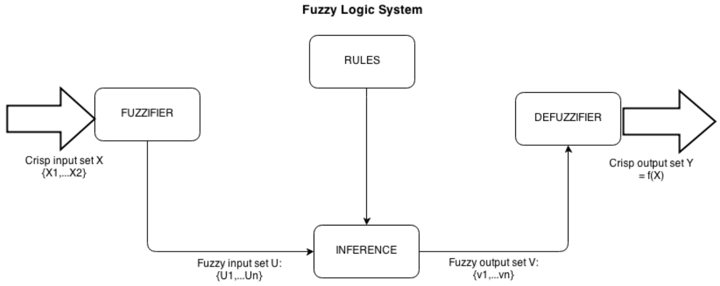

Figure 2.6: Fuzzy Logic System (4).

of the possible fuzzy set values of $\{"COLD", "WARM", "GOOD", "HOT", "COOL"\}$ with the help of the membership function shown in Figure 2.7.Membership functions are 2-dimensional graphs, where the ordinate (y-axis) represents the degrees of membership ranging from 0 to 1 and the abscissa (x-axis) represents the values like temperature.Membership functions take the form of different shapes some of which include triangular and trapezoidal shapes. It provides a measure of the degree to which a value belongs to a member of a set. The degree of membership for a given value can be determined by projecting a line from the x-axis vertically to the bounding line of a membership function.

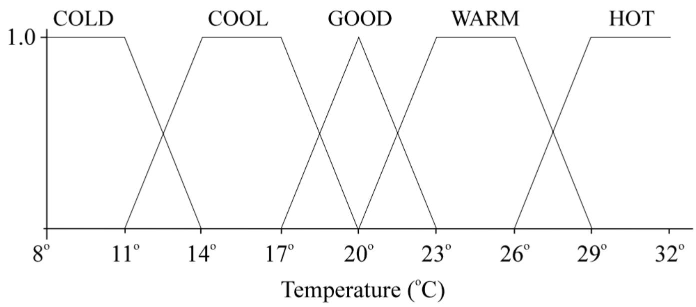

Figure 2.7: Membership Function.(3)

The inference engine is an intermediary between the fuzzifier and the defuzzifier. it maps fuzzy sets from the fuzzifier into fuzzy sets for the defuzzifier based on some already established fuzzy rules. The operations of an inference engine is analogous to the way a humans make judgment when faced with situation in their environment under some conditions. A Fuzzy rule is a combination condition and action in a conditional statement without the else part. It takes the format of " IF ⟨ condition ⟩ THEN ⟨

action ⟩”. where the condition can be a set of fuzzy set operations and action defines the output of the rule when it is activated. Boolean logical operators (AND, OR, NOT) also exist in fuzzy sets operation but connote different meanings. The AND operator represents minimum, the OR operator represents maximum, and the NOT operator represents a complement. Mathematically, NOT can be written as $\neg A(x) = 1 - A(x)$, AND or Union can be written $(A \cup B)(x) = max\{A(x), B(x)\}$ and OR or intersection can be written as $(A \cap B)(x) = min\{A(x), B(x)\}$ The defuzzifier maps output sets into crisp number via the process of defuzzification. The output from a defuzzifier can later be fed to an external system like an actuator. Applications of Fuzzy logic system include Control system applications, Scheduling and optimization, stock market analysis, prediction and forecasting of whether.

2.4.4 Regression Analysis

Upon data set collection , it is always paramount to determine the relationship between the dependent and independent variables.Regression analysis allows for the understanding of how the value of the dependent variable (or ’criterion variable’) changes when any one of the independent variables is varied, while other independent variables are held fixed. It is widely used for predicting,forecasting, and analyzing data set. It can also be used to understand which among the independent variables are related to the dependent variable. Regression analysis uses different model to predict the values of a dependent variable. These models include simple linear and polynomial regression. Simple linear regression estimates the value of a dependent variable that is linearly related to a single independent variable. While a polynomial model is used to estimate the possible polynomic relationship amongst variables. The least square methods approach is used to estimate the best line of fit for and model used.

2.4.5 Support Vector Machines

Support Vector Machines (SVMs) is a supervised machine learning method for regression and classification. They act as a discriminative classifier formally defined by a separating hyperplane. Support vector machines algorithm outputs an optimal hyperplane which categorizes new feature with the maximum margin from the hyperplane as shown in the Figure 2.8

.

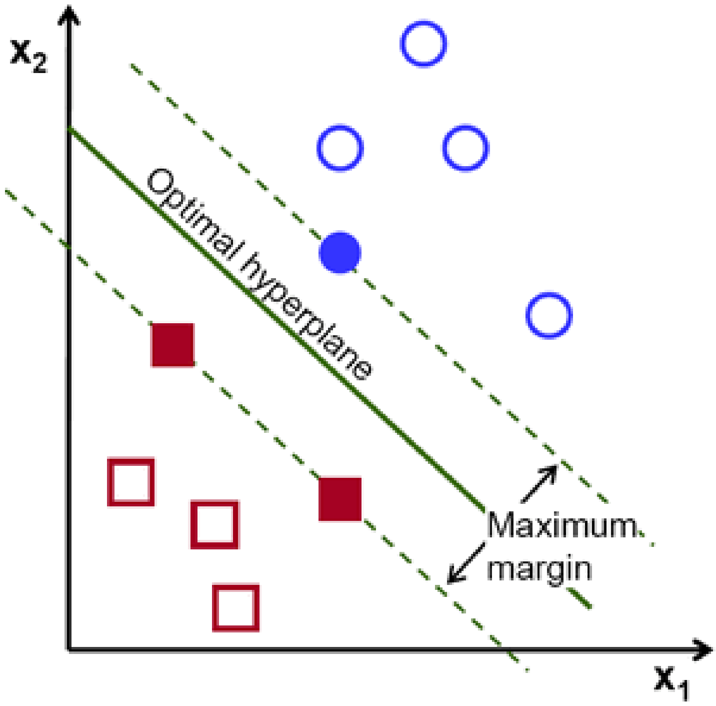

Figure 2.8: SVM Hyperplane description. ()

Put simply, the objective of SVM is to produce a model based on the training data set, to predict target values of the test data set when given only the test data attributes. Separation of data set into training and testing data is carried out for classification and regression task. Instances of the training set contains a target value which is also known as the class labels and several attributes called the features or observed variables. As shown in (13) and (14) an SVM can be generalised mathematically as :

Given training vectors $x_i \in R^n, i = 1, \ldots, l$, in two classes, and a vector $y \in R^l$ such that $y_i \in \{1, -1\}$, C-SVC

solves the following primal problem:

$$\min_{w,b,\xi} \quad \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \tag{2.3}$$
$$y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i,$$
$$\xi_i \geq 0, i = 1, \ldots, l.$$

Its dual is

$$\min_{\alpha} \quad \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$
$$0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, l, \tag{2.4}$$
$$y^T\alpha = 0,$$

where e is the vector of all ones, C is the upper bound, Q is an l by l positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, and $K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$ is the kernel. Here training vectors x_i are mapped into a higher (maybe infinite) dimensional space by the function ϕ.

The decision function is

$$f(x) = sign(\sum_{i=1}^{l} y_i\alpha_i K(x_i, x) + b).$$

Mathematically for regression :

Given a set of data points, $\{(x_1, z_1), \ldots, (x_l, z_l)\}$, such that $x_i \in R^n$ is an input and $z_i \in R^1$ is a target output, the standard form of support vector regression is:

$$\min_{w,b,\xi,\xi^*} \quad \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi_i^*$$
$$z_i - w^T\phi(x_i) - b \leq \epsilon + \xi_i,$$
$$w^T\phi(x_i) + b - z_i \leq \epsilon + \xi_i^*,$$
$$\xi_i, \xi_i^* \geq 0, i = 1, \ldots, l.$$

The dual is:

$$\min_{\alpha,\alpha^*} \quad \frac{1}{2}(\alpha - \alpha^*)^T Q(\alpha - \alpha^*) + \epsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} z_i(\alpha_i - \alpha_i^*)$$
$$\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0, 0 \leq \alpha_i, \alpha_i^* \leq C, i = 1, \ldots, l, \tag{2.5}$$

where $Q_{ij} = K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$.

The decision function is:

$$f(x) = \sum_{i=1}^{l}(-\alpha_i + \alpha_i^*)K(x_i, x) + b.$$

.

2.5 Cloud Computing

Cloud computing is one of the computing paradigms that has changed the way computing is done in recent times. It is a term that covers the delivery of computing services over the internet. These computing services are also known as cloud services. examples of these services include Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). The main points that differentiates a cloud service from traditional hosted service are that 1) it is sold on demand in a timely fashion which could be either by the minutes or hour 2) The infrastructures of the service is managed by the cloud provider therefore taking hassles of maintaining such infrastructure by the user 3) The service is elastic, this implies that users can as little or as much of the service they want at any given time. 4) A user is only obligated to pay for the amount of service he or she uses.Better access to high-speed Internet, distributed computing, virtualization, and economic standards have accelerated interest in cloud computing. A cloud can assume three possible configuration based on how the service is delivered to its users. These configurations are public, private and virtual private cloud. In a public cloud, the service is sold to anyone on the internet a typical example of a public cloud is when the user uses Amazon simple storage service (S3) to store and retrieve his or her documents. The private set up involves a company creating its own cloud service for internal use. When a company subscribes to a public cloud service in order to have its cloud service for its own internal use it is called a virtual private cloud. the main goal of this type of computing is to provide easy, scalable access to computing resources and IT services (15).

2.5.1 Cloud services

Like a web service that provides resources to clients (web or devices) over the internet, there exist different types of cloud services like Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) that provide resources to users. An explanation of these services is as follows:

- **Software as a Service (SaaS):** As the number internet users increases by the year (16). Traditional means of deployment of software to users as moved to the cloud . Software as a service is a distribution model in which a software application is made available to customers over the internet by using the software's vendor or a service provider cloud infrastructure. The benefits of this type of service are : Simple and easy administration of the software, Global accessibility, Collaboration of users using the software in real time, Users tend to use the same version of software thereby increasing robustness and compatibility and Easier update and patch management (17).

- **Platform as a Service (PaaS):** Platform as a service is a way to rent storage, operating systems hardware and network capacity over the Internet. It allows a user to rent servers and complimentary services for developing , testing and deploying applications. It is an extension of software as a service. It has several advantages for its users : Operating system can be upgraded at any time for all users, Collaboration is enhanced for geographically distributed users, total expenses can also be minimized by outsourcing the infrastructure to the service provider.

- **Infrastructure as a Service (IaaS):**. Infrastructure as a Service is also known as Hardware as a Service . In this type service an organization outsources its information technology infrastructure like servers and cooling infrastructure to a service provider that owns the equipment. The service provider in turn is responsible for housing, running and maintaining it. while the client pays for the service on a per-use (weekly, monthly, yearly) basis.Components of Iaas include computer virtualization, computer scaling, internet connectivity, policy based services, utility billing and above all automation of administrative mundane tasks.

2.5.2 Jabber

Jabber is an open source, XML-based instant messaging (IM) technology invented by Jeremie Miller in 1998 and standardized as an Extensible Messaging and Presence

Protocol (XMPP) by the The Internet Engineering Task Force (IETF) as an Internet Standard for IM and presence (**?**) . Jabber operates in a different manner when compared to other proprietary instant messaging systems. Its mode of operation is similar to e-mail, using a distributed architecture. XMPP uses decentralized client-server model, where each user connects to the server that controls its own domain. clients are assigned a jabber id (JID), which takes the format of "xxx@yyy.com" . where "xxx" represent the ID of the client on the "yyy.com" domain. This addressing scheme enables a server to read addresses from different messaging systems and reply correspondingly. In a proper IM scenario clients negotiate the server with different request containing different XML tags like $\langle presence \rangle$, $\langle iq \rangle$, $\langle stream \rangle$, $\langle resource \rangle$. Each tag serves a different purpose during the communication between the client and the server. for example the $\langle presence \rangle$ can be used to signify that a client has a session and would want to register its presence or wants to close the session with the server. XMMP was designed to be secure and extensible. Application of the protocol include publish-subscribe systems, signalling for VoIP, video, file transfer, gaming, Internet of Things applications such as the smart grid, and social networking services.
Bidirectional-streams Over Synchronous HTTP (BOSH) is an extension of XMPP over http. It is a secured way of transferring jabber xml tags over http.

2.6 Summary

Some concepts and technologies that would make the existence of wireless sensor network were outlined. The chapter started out by explaining what a wireless sensor network is and its characteristics. The arduino opensource components, shields, modules were discussed. Time series and its analysis were also outlined. Applicability of machine learning approaches like fuzzy logic and neural network was examined . The last section of the chapter looked at the cloud computing , its services and Jabber as communication protocol.

3

Problem Statement

In this chapter a brief note on sensor networks was given and the author outlines and analyses the problems that are encoutered when providing sensor data within a wireless sensors network via its sensor nodes placed at a locations. Finally, a list of the problems the author wishes to address within the context of the thesis are stated.

3.1 Research Question

Generally, Sensor nodes are sometime connected to their aggregators in the cloud via a wireless communication link, they collectively perform the task of collecting data from their environment and sending it to their agrregators for processing. The aggregated data are either stored for future use or sent to user agents that instantly need it to carry out their daily functions. The merits provided by this type of networks is that real time communications between entities are fostered . Example of such enabled communications include real time sending of a patients medical parameters like temperature, heart beat rate and pressure to a medical clinic's data store so as to constantly monitor the patient. Another type of example is sending environmental data from the extension part (a sensor node) of mobile phone for real time during the game play.

On the other hand, since the availability of context aware data in wireless sensor network is made possible by devices that run on low powered voltage sources (batteries) which may not last more than a couple of days if they are to run nomally on their own or run for a couple of hours if they are to run under optimal conditions when embedded in the sensor node. A possible cause of such quick power source consumption have been observed to be from WIFi modules attached to the the sensors, the amount of data that is transfered from the sensor to its aggregator and the available bandwith that can accomodate the sensor data (18). In the process of solving these major problems many ideas have been proposed by researchers in (19) , (20). some of the ideas include offloading computations to cloud services, reducing the rate at which data is sent to the aggregator, and supporting reduced rate of data sent with predictions as described in (21) and (8) are some of the solutions to the ever existing power problems. Similarly, security is also a concern when transferring data from one point to the other as pointed out in (8) while using hyper text transfer protocol for the sending of data.

In the light of all the aforementioned problems the author aims to look into the following problems as listed below.

- What is the best way to transport sensor data to its own aggregator ? Is it better through the hyper text transfer protocol (HTTP) or Jabber (XMPP) ?
- Prediction has been proposed to be a way of reducing the power consumed in wireles sensor network. would a neural network in combination with fuzzy logic perform better than a linear regression with fuzzy logic system as proposed in (8) ?
- What type of neural network would be best at learning and predicting in the network.

3.2 Summary

Some of the advantages that can be gained during the usage wireless sensor networks were mentioned, so also so were the problems that surround such networks. Finally, the problems that are to be tackled in this thesis were highlighted.

4

Contributions

Prediction has been one of the main ideas of reducing the high rate of energy consumption in wireless sensor network as described in (22), (8), (23), (24). In this chapter, details about the implementation of the contribution made by the author are explained, starting with the architetcture for both parties in the data exchange, to the major contributions like neural networks and the sensors used at the server and client respectively.

4.1 General Architecture

The general architecture used was a client-server type. The server otherwise known as the aggregator was hosted on Amazon EC2 server, while the client was made up of arduino and tinkerkit components. Components of the solution proposed, both share data in the form javascript object notation (JSON) over HTTP. An explanation of the contribution at the aggregator and the client end are stated in the following sections below.

4.1.1 Aggregator's Architecture

The aggregator model includes the HTTP handler handling response and request, data store keeping a record of data from the sensor node, fuzzy logic deciding the amount of time that is needed for the node to sleep and the prediction module which uses a neural network to estimate the values of data that needs to be predicted when due. The major difference of the new model as compared to the old aggregator component in (8) is that the linear regression model is replaced with an artificial neural network

(ANN). One of the major reasons of using ANN is that with a vast amount of data, a neural network can be trained or taught about the domain in which estimation is carried out i.e the network can explicitly estimate new values after learning about the variations of past dataset. Other merits of using the ANN over a regression model are 1) The variation between a quantity measured and time need not be known before the network is trained. 2) An ANN can be trained for a specific amount of time or taught to particular error value. A detailed sequence diagram of the activities that takes place at the aggregator's end is shown in Figure 4.1 . From the diagram 4.1 it can be seen that the following major steps, on the aggregators end are taken to achieve reduction in energy consumption at the client's end.

1. arduinohandler handles request.
2. System time is initialised.
3. A POST method Json request is parsed
4. Predictions are made with Neural network.
5. is prediction within the specified error level ? if yes use the predicted value, otherwise use measured value.
6. Fuzzy engine predicts the amount of time the client is expected to sleep based on some fuzzy sets.
7. Send response to the client.

A point of interest for this thesis would be the usage of Neural netowrk to estimate timed series values for a sensor node as stated in step 4. Its applicabiliity stems from the already mentioned advantages. One of the main aim of this thesis was to choose the best neural network for prediction . Examples of neutral networks that were examined include a basic network of feedforward neurons, A neural network backed up by support vector machines and a NeuroEvolution of Augmenting Topologies (NEAT) which can take the form of a recurrent or self connected neurons. It should be noted that apart from NEAT that uses a genetic algorithm for training and SVM machines that uses SVMTrain, the basic networks used Resilient Propagation (RPROP) and Levenberg Marquardt (LMA) training technique. An explanation of these terms are given below.

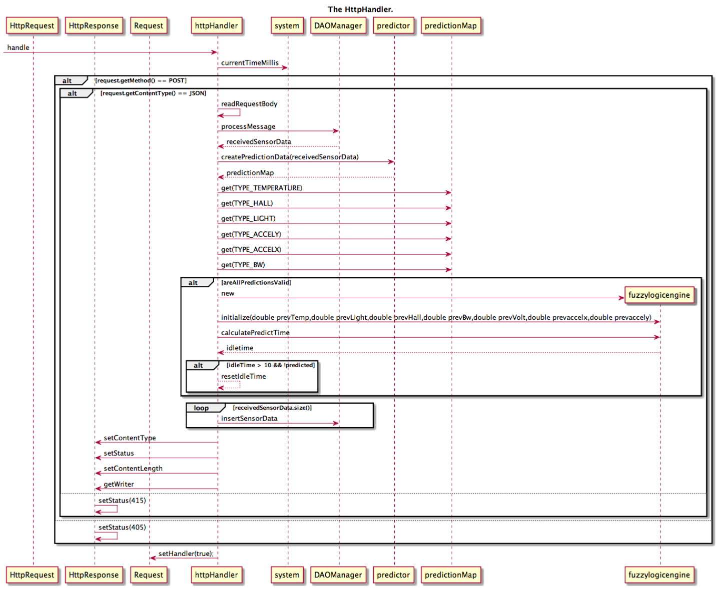

Figure 4.1: Sequence Diagram of an Aggregator.

- **Basic Artificial network** : A basic artificial neural network contains both input, hidden and output layers. It would be termed a feed forward network if all input signal end up giving one or more output without any output signal being feed back into the network. In its simplest form it would consist of a single neuron having a single input and output and no hidden layer technically as depicted in Figure 4.2 below.

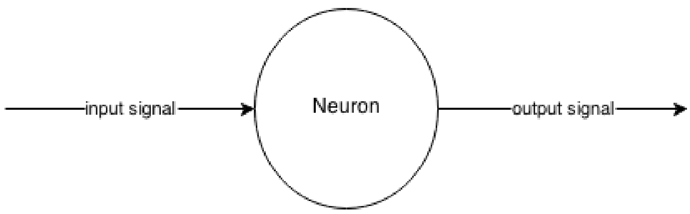

Figure 4.2: Simple Artificial neuron.

- **SVM backed neural network**: This form of neural network is backed by one or more Support Vector Machines. It typically has a single output, but can be made to simulate a neural network with multiple output neurons by using more than one neural network at the output layer. This type of network have the capacity to predict future values and classify data sets based on the type of input parameters specified to the network.

- **NeuroEvolution of Augmenting Topologies (NEAT)**: As defined in (5), it is an algorithms used for building an evolving neural networks (a neuroevolution technique). Three principles adopted by this type of network are : 1) Tracking of genes with their respective marker to allow for crossover among topologies. 2) Preserve innovations by evolution of species in the population. 3) Development of topologies gradually from simple structures. The basic idea of NEAT is that it makes it easier to figure out the optimal structure of a neural networks hidden layer, unlike the in the other network used where the network has to prunned to determine the best network with optimal hidden layer to use. A NEAT network starts with an input and output layer, just like a feedforward neural network then evolves into a recurrent or self connected network as shown in figure 4.3 below the diagram shows that the output of the network is fed back to the Hidden H1 neuron apart from the signals coming from the input (IN 2) and the biasing neuron B1.

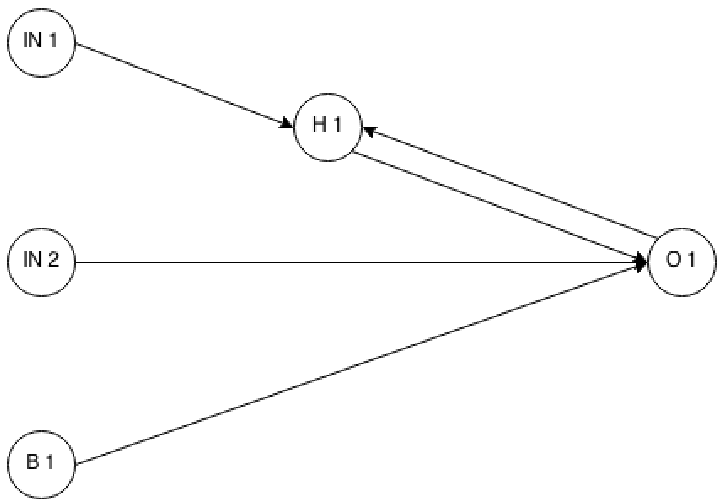

Figure 4.3: An Artificial neuron supported by NEAT. (5)

Apart from the NEAT model that evolves over time , it is worth mentioning that one the trickest part of using other neural network was coming up with the optimal neural network for each value measured. In order to solve this problem, the networks were prunned with different parameters.

4.1.1.1 Training Algorithms

Resilient Propagation (RPROP) and Levenberg Marquardt (LMA) training technique were used to train the network a brief explanation given below:

- **Resilient Propagation**: The resilient propagation training (RPROP) algorithm is one of the training algorithms used in the ENCOG neural network library. It is an efficient training algorithm for supervised feedforward neural networks. Part of the merits of using the RPROP algorithm is that it requires no parameter settings like learning rates, momentum values or update constants before using it.

- **Levenberg Marquardt**: The Levenberg Marquardt algorithm (LMA) is an effiecient training method for neural networks. it is a combination of both Newtons Method and backpropagation algorithm. backpropagation converges to a local minimum slowly. Newtons Method is quite fast but often fails to converge. LMA is created by using a damping factor to interpolate between the two algorithms.


```
 {"location": 1, "data": [
    {"type": 1,"value": 7.47 },{"type": 2,"value": 842.00},
    {"type": 3,"value": 512.00},{"type": 4,"value": 9.00},
    {"type": 5,"value": 8.87}, {"type": 6,"value": 426.00},
        {"type": 7,"value": 551.00}, {"type": 8,"value": 2.3}
]}
```


Listing 2: Sample JSON Data exchanged between Node and Aggregator.

4.1.2 Client's Architecture

The Client end has been enhanced with more sensors like voltage, and accelerometer sensors apart from light, temperature and humidity sensors that were used before. A sample of the JSON structure sent from the node to the server was shown in Figure 2. As it can be seen in listing 2 , There are 7 different sensors values transmitted. The maping of these sensor values are shown in table 4.1 with the physical figure 4.4 shows an image of the setup when communicating with an aggregator.

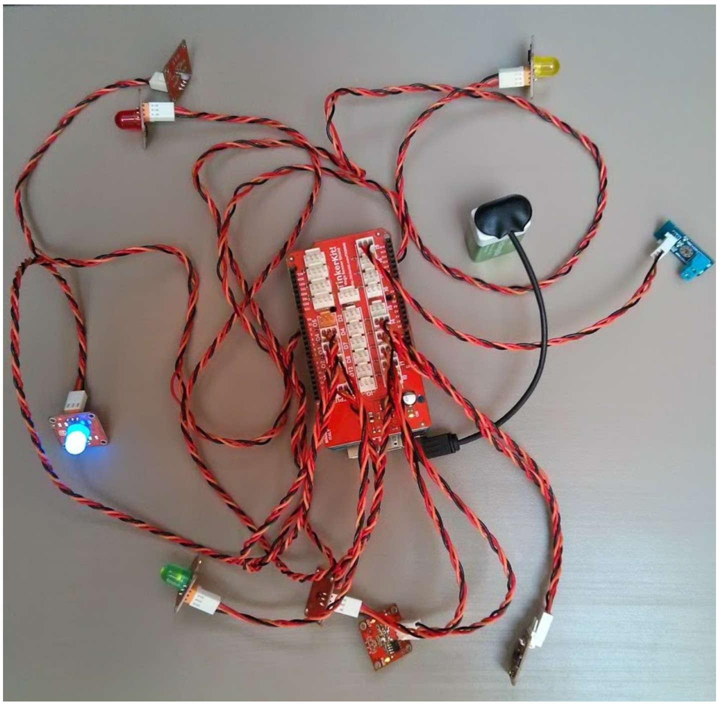

Figure 4.4: A physical Sensor node (Arduino and components).

4.2 Summary

This chapter outlined the basic components of the prediction module, the algorithm behind the training of the networks and lastly, the author talked about the major components of the clients and their respective functions.

Table 4.1: Sensor Data Mapping

Type Value	Data Acquired by Sensor/ Measured
1	Temperature
2	Humidity
3	Light
4	Battery Status
5	CPU Usage
6	X value of accelerometer
7	Y value of accelerometer
8	Band width

5

Case Studies

The intent of this chapeter is to discuss the physical implementation of the proposed contribution as regards the existing and new environment. Some tests were carried out before carrying out the implementation of the case study so as to answer the problems declared in chapter 3.

5.1 HTTP Communication

In order to answer part of the first question stated in the problem statement, HTTP setup was first considered with and without the prediction Logic as outlined below :

5.1.1 HTTP communication without prediction logic

As usual, the setup was made up of the server and the client. The server (aggregator) comprised of the HTTPHandler, and the datastore module while the client is just a simple wireless node with the eight (arduino) sensors measuring the environmental features. Figure 5.1 shows a diagram depicting such environment:

The idea behind this form of communication was to see how long a 9v battery would last normally. Table 5.1 shows that on the average a 9v battery would last for 62.3 mins with eight different sensors collecting data from the environment.:

5.1.2 HTTP communication with Linear regression and Fuzzy Logic

In order to see if the suggested neural network as proposed in this thesis would be a gain or a disadvantage, it is worth examining the process flow which is listed as follows:

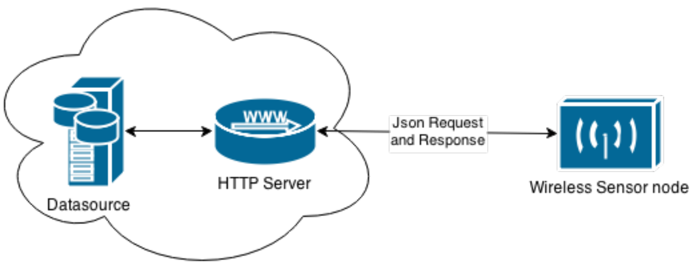

Figure 5.1: HTTP communication without logic.

Test	1	2	3	4	5	6	7	8	9	10
Voltage	9.0	9.0	8.78	9.0	9.0	9.0	8.91	8.78	8.98	9.0
Time (min)	60	66	40	83	60	59	68	62	65	60

Table 5.1: Duration for normal HTTP with a 9v battery

1. Data is initially sent to the aggregator for a period of 5 minutes in order to build the knowledge base of the aggregator without any prediction model in place.

2. After the duration of building the initial dataset, subsequent request are sent and are subjected to the prediction module:

3. Under the assumption that the first request comes in after building the knowledge base. The past 4 minutes values for each sensor are first sampled and aggregated into the simple linear regression model. For each sensor there is a predicted value ρ_i and a tendency of predictability value α_i these parameters are defined as follows :

$$\rho_i = ax_i + b \tag{5.1}$$

$$\alpha_i = 2(\epsilon_i/R_i) - 1 \tag{5.2}$$

for equation 5.1, ρ_i is the predicted value, **a** is the slope of the linear model and **b** is the intercept on the axis of the sensor **i** considered. Similarly, for equation 5.2, α_i is the predictability value, ϵ is the regression error and **R** is the root mean square error of the sampled data for sensor **i** .

Test	1	2	3	4	5	6	7	8	9	10
Voltage	9.0	9.0	8.78	8.90	9.0	8.87	8.91	8.78	8.98	9.0
Time (min)	141	120	73	120	90	110	90	89	70	90

Table 5.2: Duration for HTTP communication with Linear Regression and Fuzzy Logic with a 9v battery

4. For each sensor **i** the validity and the predictability are checked if they are valid and greater than the set threshold which is 90% they are passed onto the fuzzy logic module to determine predicting duration at the aggregator.

5. The fuzzy Logic engine is initialised and each sensor **i** is passed as input at once. Fuzzification of the input is also started which implies that the predictability and the predicted value for each sensor is mapped to a member of a predicted and predictability fuzzy set respectively via its own declared membership function.

6. Fuzzy rules are then applied to the results of the fuzzification. Each variable has three rules defined and is possibly fired at any time. These rules are stated below:

   (a) IF *x* IS *xChangedLeft* OR *xChangedRight* AND *xPredictability* IS *notPredictable* OR *predictable* THEN *request*.

   (b) IF *x* IS *xSame* AND *xPredictability* IS *notPredictable* THEN *request*.

   (c) IF *x* IS *xSame* AND *xPredictability* IS *predictable* THEN *predict*.

7. Results from the application of the rules are then passed through the defuzzifier to obtain a crisp value which is the duration of time for prediction at the aggregators end.

8. Finally, a response specifing the amount of time to sleep is then sent to the sensor node.

Figure 5.3 depicts all the steps mentioned and table 5.2 shows the results from the test carried out with the setup. On the average the battery would last for about 98.4 mins which is about 37% increase as compared to the ordinary HTTP experiments carried out.

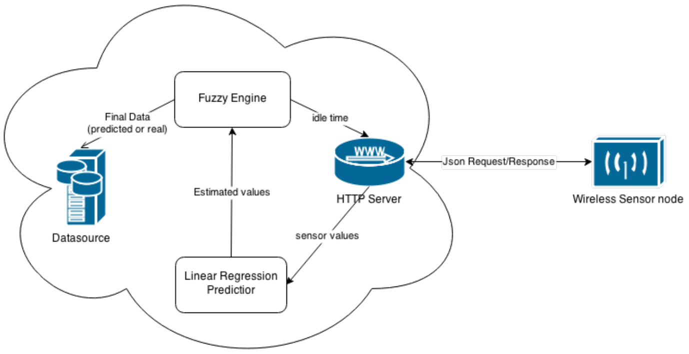

Figure 5.2: HTTP communication with Linear Regression and Fuzzy Logic.

Test	1	2	3	4	5	6	7	8	9	10
Voltage	9.0	9.0	8.78	9.0	8.85	9.0	8.91	8.78	8.98	9.0
Time (min)	100	80	73	85	90	70	71	72	70	80

Table 5.3: Duration for XMPP with a 9v battery

5.2 XMPP Communication

XMPP communication followed the same mode of operation like the HTTP communication mentioned above. In order to comunicate with the server , a TCP connection is established, an exchange of xml opening stream tags are done, SASL authentication is carried out, resource binding between both parties is done, the client establishes its presence, data is then sent from the client to the server, once the client is done the sending the data , a closing stream tag is sent to the server in which the server responds with a closing stream tag too. Finally, the TCP connection is then closed. The amount of time it takes a 9v battery to undertake such activity is given in table 5.3

As it can be seen from the values obtained, XMPP mode of operation does not help in any way because of the average time of 79.1 minutes which is less than that of the normal HTTP communication.

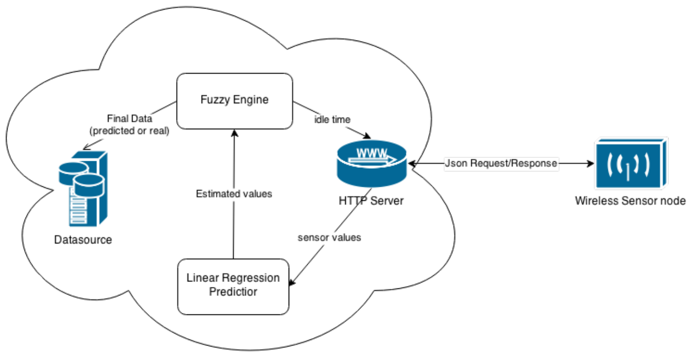

Figure 5.3: XMPP squence diagram.

5.3 HTTP and Neural Network Communication

In this scenario the major idea was to substitute the linear prediction model with a neural network, mainly because it would be advantageous for the following reasons:

1. To learn data over wide range of time.
2. A particular level of accuracy can be predefined for training the data
3. There is no need to determine the possible shape of the relationship between the variable and time before selecting the right regressional model to use.

Three types of neural networks were tried, in order to come up with the best network to use. These networks include a Basic neural network, NEAT network and an SVM backed neural network. The stages involved before the best neural networks were picked are stated in the section below: the neural networks with a maximum of 2 hidden layers.

5.3.1 Training

5.3.1.1 Basic

Starting with the simplest form of configuration the basic neural nework was trained using the Resilient propagation and LMA training techniques. The Target error values for both cases were 0.01. Each network configuration was built and then prunned to get the optimal network. A dataset from previous normal http communication were used to train the data. Three-quarter of the total dataset were used to train the network.

$RMSE \diagdown MINS of data$	30	25	20
$RHall_{train}$	0.45	0.04	0.05
$RHall_{test}$	0.02	0.02	0.02
$RTemperature_{train}$	0.06	0.06	0.05
$RTemperature_{test}$	0.01	0.03	0.02
$RLight_{train}$	0.01	0.01	0.01
$RLight_{test}$	0.01	0.01	0.01
$RBandwidth_{train}$	0.03	0.03	0.04
$RBandwidth_{test}$	0.01	0.02	0.01
$RAccelx_{train}$	0.04	0.04	0.05
$RAccelx_{test}$	0.01	0.03	0.03
$RAccely_{train}$	0.03	0.04	0.04
$RAccely_{test}$	0.02	0.02	0.02
$RBattery_{train}$	0.04	0.04	0.04
$RBattery_{test}$	0.01	0.02	0.04

Table 5.4: RMSE Values for Resilient algorithm

In the case of resilient propagation training for a basic network it was observed that with small amount of data, the best network after prunning, would converge to the error value as soon as possible but with a large dataset it would always take more than one to two hours to train the network. Similarly, LMA is also a good algorithm but it converges slower than the resilient algorithm. The table 5.4 and 5.5 shows the RMSE values with respect to the amount data collected within 30, 25, and 20 minutes for the resilient algorithm. Variables with the train subscript show the training RMSE and those with test subscript show test RMSE.

5.3.1.2 NEAT

NEAT by default starts with an input and an output layer which would naturally evolve into the best network given an amount of population. Generally, with a higher number of population the network is well trained but it was the discovered that the best population to use for the network was 1000. With a population size of 1000, one single input and output the network and their respective RMSE values for training are

$RMSE \diagdown MINS$	30	25	20
$RHall_{train}$	0.45	0.04	0.05
$RHall_{test}$	0.02	0.02	0.02
$RTemperature_{train}$	0.06	0.06	0.05
$RTemperature_{test}$	0.01	0.03	0.02
$RLight_{train}$	0.01	0.01	0.01
$RLight_{test}$	0.01	0.01	0.01
$RBandwidth_{train}$	0.03	0.03	0.04
$RBandwidth_{test}$	0.01	0.02	0.01
$RAccelx_{train}$	0.04	0.04	0.05
$RAccelx_{test}$	0.01	0.03	0.03
$RAccely_{train}$	0.03	0.04	0.04
$RAccely_{test}$	0.02	0.02	0.02
$RBattery_{train}$	0.04	0.04	0.04
$RBattery_{test}$	0.01	0.02	0.04

Table 5.5: RMSE Values for LMA

shown in table 5.6

5.3.1.3 SVM backed Neural Network

This form of network was also trained with different parameters as specified by the regression based support vector machine. Radial basis Function kernel type was used. Parameters like ε and C for SVM as described in the equation 5.4 and 5.3 in (25) were also used.

$$C = max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|) \quad (5.3)$$

$$\varepsilon = \tau\sigma\sqrt{(ln(n)/n)} \quad (5.4)$$

Equation 5.3 shows that C is the maximum of the absolute value of three times the mean values of the output plus the deviation from the mean and absolute value of three times the mean values of the output minus the deviation from the mean. for Equation 5.3 n is the number of datasets sampled and τ is taken to be eqauls to 3. On the

$RMSE \diagdown MINS$	30	25	20
$RHall_{train}$	0.45	0.04	0.05
$RHall_{test}$	0.05	0.05	0.05
$RTemperature_{train}$	0.12	0.59	0.31
$RTemperature_{test}$	0.01	0.03	0.30
$RLight_{train}$	0.6	0.7	0.6
$RLight_{test}$	0.6	0.6	0.6
$RBandwidth_{train}$	0.05	0.05	0.05
$RBandwidth_{test}$	0.02	0.02	0.03
$RAccelx_{train}$	0.04	0.05	0.05
$RAccelx_{test}$	0.03	0.05	0.05
$RAccely_{train}$	0.05	0.05	0.05
$RAccely_{test}$	0.04	0.05	0.05
$RBattery_{train}$	0.04	0.04	0.04
$RBattery_{test}$	0.01	0.02	0.04

Table 5.6: RMSE Values for NEAT

average most SVM neural network ran faster than their counterpart. Table 5.7 shows the value of the RMSE for the SVM structured neural network

After training and testing of the networks in a standalone manner. Some factors that were considered in order to select the best network were 1)Lower RMSE value since it is a form of accuracy measure generally. SVM backed neural network showed to be promising in this respect. $RMSE_{test}$ values for each sensor were compared and lowest for each sensor per neural network was used to drive the aggregator . 2) The fastness of the computation was also put into consideration. Figure 5.4 shows a comparison of the lowest $RMSE_{test}$ value for each network.

5.4 Validation

On placing the neural network into the aggregators as shown in figure 5.5. The flow stated in the http communication with logic were observed except for step 3 that was substituted with ANN for predictions. The values for the measured error was kept 0.01 and regression is five times the value of the measured error and for each sensor that

$RMSE \diagdown MINS$	30	25	20
$RHall_{train}$	0.05	0.06	0.05
$RHall_{test}$	0.04	0.05	0.03
$RTemperature_{train}$	0.05	0.05	0.05
$RTemperature_{test}$	0.05	0.03	0.05
$RLight_{train}$	0.18	0.17	0.177
$RLight_{test}$	0.19	0.1	0.1
$RBandwidth_{train}$	0.09	0.09	0.09
$RBandwidth_{test}$	0.07	0.07	0.08
$RAccelx_{train}$	0.06	0.07	0.1
$RAccelx_{test}$	0.6	0.1	0.1
$RAccely_{train}$	0.06	0.05	0.03
$RAccely_{test}$	0.02	0.01	0.01
$RBattery_{train}$	0.04	0.04	0.04
$RBattery_{test}$	0.01	0.02	0.04

Table 5.7: RMSE Values for SVM

Test	1	2	3	4	5	6	7	8	9	10
Voltage	8.9	9.0	8.78	9.0	8.85	9.0	8.91	8.78	8.98	9.0
Time (min)	120	120	74	120	87	110	90	89	90	90

Table 5.8: Duration for SVM substituted into the network.

was used. The network was observed and the neccessary readings were obtained. A summary of how well the setup did is shown in table 5.8. From the table above it can be deduced that average time it would the take the battery to last is 99 mins, which is like 0.6 % increase.

5.5 Summary

This chapter looked at various cases of Wireless sensor network with Normal HTTP communication , HTTP communication with logic, XMPP and HTTP comunication with neural network predictions. From the experiment carried out it was observed that

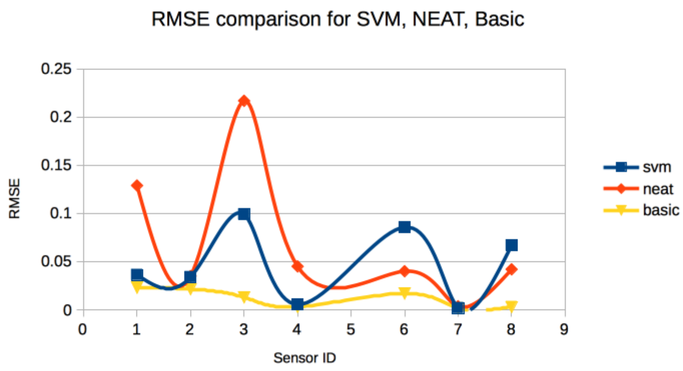

Figure 5.4: RMSE comparison for SVM, NEAT, Basic

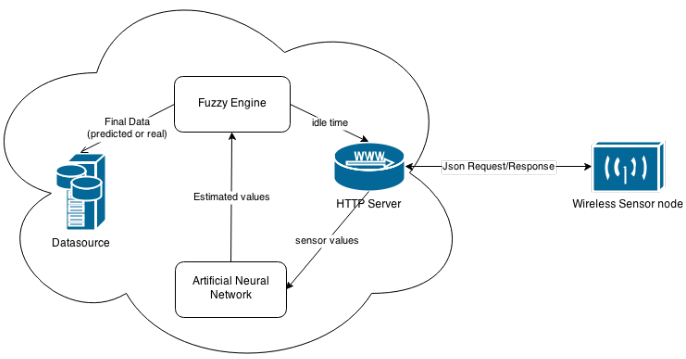

Figure 5.5: Aggregator module with neural network

the HTTP mode of communication is far better than the XMPP. A replacement of the linear regression predictor with a neural network did not yield much different results as proposed.

6

Related Work

In this chapter, past works carried out by other researchers would be discussed and the effect of how they have shaped work in this thesis.

It is a known fact that this work has been pioneered by the different researchers in the past. This implies that most of the work carried out in this thesis draw inspiration from past work other researchers.

Reducing the rate of transmission of the source and the number of data transmitted from the source to its aggregator in the cloud are one of the many ways of reducing the energy consumed by a continous wireless sensing node in it is network. Based on the fact that environmental parameters like temperature are almost constant during a specific time interval, one can place a prediction model and a knowledge base at either the sink (aggregator) or the node. in order to achieve a long lasting network.

Previous work like (24) suggested an automatic auto regressive-integrated moving average (ARIMA) modeling. The prototype includes the automatic creation of a model at the sensor end, which is then used to estimate the future data values based on some predefined error values. If the deviation of the predicted to the sensed value is too large at the node, the sensed value is used to build the model all over again and the new parameters of the ARIMA model are sent to the aggregator so that it can be in sync with the node. Despite the improvement shown by this work it has been shown in (23) that ARIMA model is computationally intensive. Authors in (26) and (27) have also shown that a large amount data set in the knowledge base does not influence the prediction outcome but rather increases the computational cost.

Authors in (23) did a comparative study of existing prediction models such as Naive, fixed-Weighted Moving Average (WMA), Least-Mean-Square(LMS) and Auto regressive integration moving average (ARIMA) for application specific WSN monitoring. Authors in (23) were able to show that a Naive prediction model outshines other prediction models. Mathematically, the Naive model could be written as $\hat{P}_{t+1} = P_t$. where P_t is the last observed value and $\hat{P}_{t+1}$ is the prediction for the future. They were also able to show that by placing complementary algorithms both at the source and sink end that 1) Naive is less computationally expensive 2) While using the naive prediction model there was need of training the model, the only thing needed was to prefill the knowledge base of the source and sink for certain amount of time before prediction starts. Naive model is able to achieve up to 96% transmission reduction efficiency. Authors in (22) also proposed SWIFTNET, of which compression with a combination of the naive model were used. The basic idea was to compress the data sent from the nodes over the network to the aggregators in order to reduce the rate of energy consumption. The outcome of the research showed that on the average over 50% improvement was achieved in terms of energy savings without loss in accuracy over the naive model approach that is widely used

Finally, the work in (8) is worth mentioning, the author postulated and showed that a combination of linear regression model using method of least squares and fuzzy logic could simply alleviate the problem of energy consumption by 80% in the day and 110% during the night in a context aware network for mobile users.

Considering all the aforementioned related work, this thesis was built upon (8) with modification to the prediction module in order to cater for better prediction values. The main reason linear regression module was substituted with Artificial Neural Network (ANN) was that, linear regression models are good at coming up with interpolated values as compared to extrapolated or predicted values and with a neural network predictions can be made easily at a reduced error vlaue after existing data set has been used to train the best ANN from the experiment conducted.

7

Conclusions and Future Research Directions

In this write up, an idea of increasing the performance of a wireless sensor was conceived. Artificial neural network (ANN) as an approach of machine learning was proposed to increase the perfomance at the aggregators end. The best out of the three neural network considered based on performance and other factors was picked. The existing regression method was replaced with best Artificial neural network (SVM backed network) in order to compare its performance. Results from the observations showed that ANN was able to decrease the rate of energy consumption. As compared to the the existing method they both yield almost the same level of results. This implies that from the author findings the network could be used for obtataining a performance for a wireles sensor network.

As regards research directions, the author would like to consider two areas of improvements. They are software and hardware components. For the software components the neural network algorithm can be optimised to be able to crunch more data that span more time range. Secondly, the neural network and the fuzzy logic engine could be combined into one neural network powered by fuzzy logic in other to reduce the number computations don by two disparate system. Security at the data level is another major concern that should be investiagted. Compression of the data sent across the should be investigated since less data sent across the network would mean low energy consumption. Considering the hardware side, hardware devices like Wispes W24TH (28) and Beaglebone Black (29) should be examined for their optimal operations and low energy comsumption as compared to the arduino environment.

8

License

Non-exclusive licence to reproduce thesis and make thesis public

I, Lukman Oladimeji Adekunle, herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1. (a) Reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   (b) Make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright

Prediction in Wireless sensor networks - A neural network approach
supervised by Satish Narayana Srirama,

2. I am aware of the fact that the author retains these rights

3. I certify that granting the non-exclusive licence does not infringe the intel- lectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 26 May, 2014

Bibliography

[1] Arduino, What is arduino, [Online; accessed 14-April-2014].
URL http://www.arduino.cc/ viii, 8, 10

[2] TinkerKit, Tinker kit components, [Online; accessed 14-April-2014].
URL http://www.tinkerkit.com/ viii, 15

[3] M. G. Simoes, Introduction to fuzzy control, Colorado School of Mines, Engineering Division, Golden, Colorado. viii, 18, 22

[4] J. Mendel, Fuzzy logic systems for engineering: a tutorial, Proceedings of the IEEE 83 (3) (1995) 345–377. doi:10.1109/5.364485. viii, 21

[5] J. Heaton, Programming Neural Networks with Encog 3 in Java, Heaton Research, Incorporated, 2011.
URL http://books.google.ee/books?id=KLmbngEACAAJ viii, 37, 38

[6] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Comput. Netw. 54 (15) (2010) 2787–2805. doi:10.1016/j.comnet.2010.05.010.
URL http://dx.doi.org/10.1016/j.comnet.2010.05.010 2

[7] H. Kaarel, Context sensor data on demand for mobile users supported by, Master's thesis, University of Tartu (May 2012). 3

[8] L. Kruusame, Energy-aware sensor data collection for mobile users, Master's thesis, University of Tartu (May 2013). 4, 32, 33, 34, 35, 56

[9] Ieee standard for inertial sensor terminology., IEEE Std 528-1994 (1994) i–doi:10.1109/IEEESTD.1994.121447. 15, 16

[10] A. Kar, Future image prediction using artificial neural networks. 20

[11] Wikipedia., Time series — wikipedia, the free encyclopedia, [Online; accessed 11-April-2014] (2014).
URL http://en.wikipedia.org/w/index.php?title=Time_series&oldid=602811031 20

[12] Wikipedia., Fuzzy logic — wikipedia, the free encyclopedia, [Online; accessed 12-April-2014] (2014).
URL http://en.wikipedia.org/w/index.php?title=Fuzzy_logic&oldid=602400581 20

[13] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92, ACM, New York, NY, USA, 1992, pp. 144–152. doi:10.1145/130385.130401.
URL http://doi.acm.org/10.1145/130385.130401 24

[14] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297. doi:10.1023/A:1022627411411.
URL http://dx.doi.org/10.1023/A:1022627411411 24

[15] M. Rouse., Cloud computing, [Online; accessed 14-April-2014] (2010).
URL http://searchcloudcomputing.techtarget.com/definition/cloud-computing 27

[16] Wikipedia., Global internet usage — wikipedia, the free encyclopedia, [Online; accessed 14-April-2014] (2014).
URL http://en.wikipedia.org/w/index.php?title=Global_Internet_usage&oldid=595556263 28

[17] M. Rouse., Software as a service, [Online; accessed 14-April-2014] (2010).
URL http://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service 28

[18] K. Kumar, Y.-H. Lu, Cloud computing for mobile users: Can offloading computation save energy?, Computer 43 (4) (2010) 51–56. doi:10.1109/MC.2010.98. 32

[19] A. Klein, C. Mannweiler, J. Schneider, H. Schotten, Access schemes for mobile cloud computing, in: Mobile Data Management (MDM), 2010 Eleventh International Conference on, 2010, pp. 387–392. doi:10.1109/MDM.2010.79. 32

[20] H. Jiang, S. Jin, C. Wang, Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks, Parallel and Distributed Systems, IEEE Transactions on 22 (6) (2011) 1064–1071. doi:10.1109/TPDS.2010.174. 32

[21] G. Li, Y. Wang, Automatic arima modeling-based data aggregation scheme in wireless sensor networks., EURASIP J. Wireless Comm. and Networking 2013 (2013) 85.
URL http://dblp.uni-trier.de/db/journals/ejwcn/ejwcn2013.html 32

[22] F. Aderohunmu, G. Paci, L. Benini, J. Deng, D. Brunelli, Swiftnet: A data acquisition protocol for fast-reactive monitoring applications, in: Industrial Embedded Systems (SIES), 2013 8th IEEE International Symposium on, 2013, pp. 93–96. doi:10.1109/SIES.2013.6601478. 34, 55

[23] F. Aderohunmu, G. Paci, D. Brunelli, J. Deng, L. Benini, M. Purvis, An application-specific forecasting algorithm for extending wsn lifetime, in: Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on, 2013, pp. 374–381. doi:10.1109/DCOSS.2013.51. 34, 55

[24] G. Li, Y. Wang, Automatic arima modeling-based data aggregation scheme in wireless sensor networks, EURASIP Journal on Wireless Communications and Networking 2013 (1) (2013) 1–13. doi:10.1186/1687-1499-2013-85.
URL http://dx.doi.org/10.1186/1687-1499-2013-85 34, 54

[25] V. Cherkassky, Y. Ma, Selection of meta-parameters for support vector regression, in: J. Dorronsoro (Ed.), Artificial Neural Networks ICANN 2002, Vol. 2415 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2002, pp. 687–693. doi:10.1007/3-540-46084-5_112. URL http://dx.doi.org/10.1007/3-540-46084-5_112 50

[26] D. Tulone, S. Madden, Paq: Time series forecasting for approximate query answering in sensor networks, in: Wireless Sensor Networks, Springer, 2006, pp. 21–37. 55

[27] S. Santini, K. Romer, An adaptive strategy for quality-based data reduction in wireless sensor networks, in: Proceedings of the 3rd International Conference on Networked Sensing Systems (INSS 2006), 2006, pp. 29–36. 55

[28] Wispes, Wispes w24th sensor node (2014). URL www.wispes.com 58

[29] B. Bone, Beaglebone black (2014). URL http://beagleboard.org/Products 58