Project Proposal

Project Proposal	**April 2018**	**Note:**

Student to complete this section			
Matusowsky	**M.**	**Mr.**	**11093804**
Design of a virtual sensor using machine learning imputation techniques in a wireless sensor network.			**Study leader: Mr D. Ramotsoela**

Class group: English	**Project number:**	DR2	Revision number:	1
Degree programme enrolled for: Computer Engineering				
Student declaration: I understand what plagiarism is and that I have to complete my project on my own.	Student signature		Date	

Declaration by language editor	
I have been allowed adequate time to read this document carefully and to make corrections where necessary (date received indicated below). To the best of my knowledge, correct formatting, spelling and grammar are used throughout the document.	
S. N. Armstrong (language editor)	Date

Declaration and recommendation by study leader		
1. Have you (the study leader) been allowed adequate time to read and comment on the Project Proposal?	Yes	No
2. Is the Project Proposal a <u>correct</u> and <u>complete</u> description of what is required?	Yes	No
3. Is the Project Proposal <u>clear</u> and <u>unambiguous</u>?	Yes	No
4. Recommendation: Do you recommend that the Project Proposal be approved?	Yes	No
Mr D. Ramotsoela (Study leader)	Date	

This section to be used by the Project lecturer					
Content /20		Attended lectures:	Yes	No	
Subtract for editing errors /10		Language editing adequate:	Yes	No	
Final mark /20		Approved? (If "No", a revision must be submitted):	Yes	No	Prof. J.J. Hanekom

1. Problem statement

In this section the problem that is being addressed by the project proposal is described in terms of the motivation, context, technical challenges and limitations of the proposed project.

Motivation. Sensors are devices used in everyday objects such as mobile phones, touch-sensitive elevators and self-driving cars. There are an innumerable amounts of application where sensors are used to gather data about the environment and then used in other electronics to monitor or react to the conditions being sensed such as the lighting in a room being adjusted based on the required illumination level. A problem arises in wireless sensor networks (WSN) where, due to the nature of wireless communication, data can be lost or corrupted during transmission due to external factors such as solar radiation corrupting data from satellites or congestion in a network causing packet loss when transmitting data to other devices. Furthermore the cost of implementing or needing to replace many physical sensors in a network can become prohibitively expensive.

This project will look at designing and implementing a wireless sensor network that will make use of data imputation methods and machine learning to realise virtual sensors that can completely replace physical sensor nodes and give accurate substituted data in place of nodes with failed sensor modules.

Context. The integrity of received information is an important issue in the modern age. Sensors play a pivotal role in electronic devices of all shapes, sizes and function and especially more so in wireless sensor networks where data loss is an expected occurrence [1]. Data imputation and virtual sensors are tools that allow a system to counter-act the effect of this data loss by accurately substituting values that real sensors would most likely return [2]. This then allows the system to still make use of incomplete data rather than completely discarding affected data entries .

The main function of this project will be ensure the robustness of a wireless sensor network by making sure that damaged nodes in a wireless sensor network can be replaced by virtual sensors using imputation techniques that make use of machine learning algorithms which will allow the system to remain robust in terms of the provided sensor data even when a node malfunctions or is removed from the network thus allowing the system to continue running in real-time while gathering data that closely resembles the affected sensor nodes would-be data.

K-Nearest Neighbours (KNN, lazy learning), multi-layered perceptrons (MLP, supervised learning) and self-organizing maps (SOM, unsupervised learning) are three popular methods that have been used to great effect in data sets that do not deal with time-series analysis [3, 4, 5, 6] where KNN, MLP and SOM outperform traditional imputation techniques, such as hot-swapping, to significant degrees on multiple data sets including but not limited to a breast cancer detection data set, a seed classifying data set and sonar imaging data set.

Technical Challenges. The technical challenges for this project are: (i) Designing and implementing an imputation technique using machine learning algorithms on individual sensor nodes that can act as a virtual sensor. (ii) The algorithm should be robust to ensure the integrity of the data that is being substituted by the virtual sensors. (iii) The algorithm should be efficient enough so that congestion does not occur due to computations taking place which in itself would then cause loss of more data. (iv) Implementing a low-power solution for the sensor nodes so that the nodes in the system may run independently "off-the-grid".

Limitations. The availability of bandwidth in the network will limit how many readings can be transferred and received per time interval between all the nodes and the server. The second limitation is the processing speed and power of the processing unit at each sensor node. Another limitation is the cost of developing the product thus cost-effective hardware is a necessity as well as putting a financial limit on the amount of nodes that can be physically implemented for this project.

2. Project requirements

The main aim of the project will be to create a wireless sensor network that makes use of machine learning imputation techniques to ensure the integrity and robustness of the data that is transmitted and received over the network.

2.1 Mission requirements of the product

The system will need to fulfill the following requirements

- The system must use machine learning algorithms to implement virtual sensors in the wireless sensor network.
- Communication between nodes in the network must be done using wireless communication.
- The algorithms implemented must be efficient enough so as not to introduce computational congestion into the system.
- The system must detect a malfunctioning node and replace it with a virtual sensor.
- The virtual sensors must be able to replicate data accurately as if they were real sensors.
- The data read by the all the sensors must be stored in a database.
- The virtual sensors must be implemented on every corresponding node as well as having a copy of every trained virtual sensor on the server.
- Each sensor node should consist of a power unit, a sensor module and a processing unit.

2.2 Student tasks

The following tasks will need to be completed.

- An investigation of the current machine learning imputation techniques in literature must be done.
- An investigation of where and how virtual sensors have been implemented must be done.
- An investigation of wireless communication interfaces must be done to decide on the best communication interface for the product must be done.
- An investigation for the causes of lost data in wireless sensor networks must be done.

- The imputation algorithms must be designed and implemented with a focus on robustness.
- The algorithms must be trained and tested in MATLAB or Python using data that has been collected from the environment being sensed by the hardware.
- The control algorithms on the processing units for each sensor node must be implemented.
- The circuitry for the sensor nodes must be implemented on veroboard.
- The software and GUI for the server must be designed and implemented on a PC.

3. Functional analysis

This section contains information on the subsystems and each functional unit of the system separated into the hardware and software components.

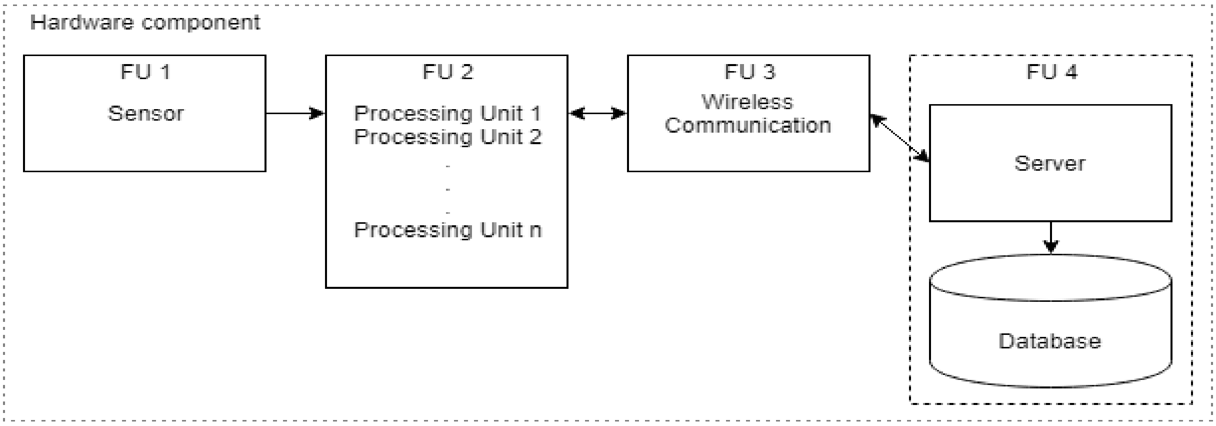

Figure 1. Overview of the hardware component.

The hardware component, shown in figure 1, will consist of a sensor module (FU 1) which will sense the characteristics of the environment and be attached to a processing unit (FU 2). These two functional units cover a single independently powered sensor node. The processing unit will, at regular intervals, take environmental readings from the attached sensor(s) and package the data for transmission via wireless communication (FU 3) with the other sensor nodes as well as a central server (FU 4) which will contain the database that will store historical data for later use in data modeling and then training the virtual sensors of each sensor node as well as general storage for historical data of the network. The wireless communication interface will be attached to the processing unit where a communication protocol will control the flow of the communication between the server and sensor nodes.

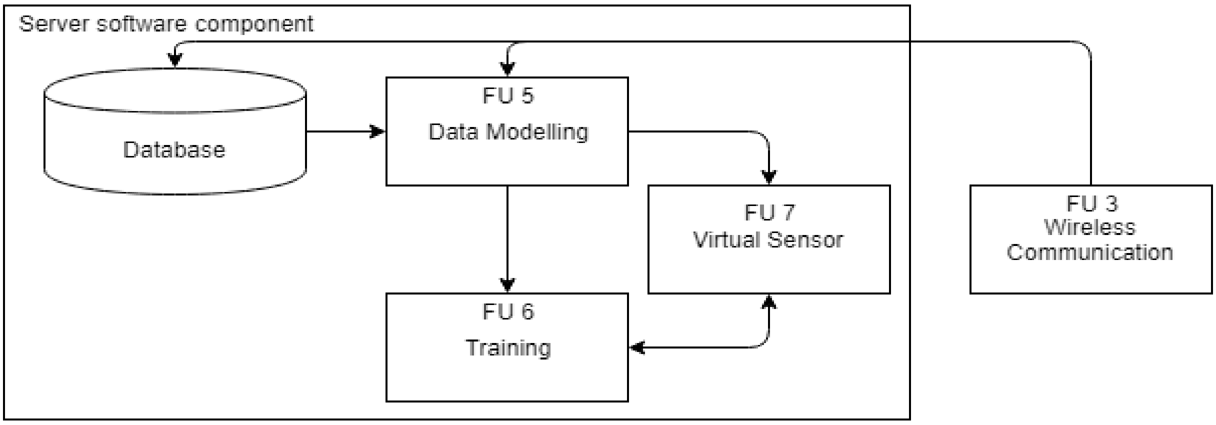

Figure 2. Overview of the server software component.

The server software component, shown in figure 2, will consist of the aforementioned database and all the software of the system. When the system is functioning in a manner that all the sensor nodes are active, the server will simply store the sensor readings received from the sensor nodes through a wireless communication link (FU 3) in the local database where imputed values will be marked as such. This data will then be modeled (FU 5) so that it may be used as training (FU 6) data for the machine learning algorithms that will be deployed as virtual sensors on both the server as well as the processing unit of the sensor nodes.

When the system is functioning in a manner that a sensor node in the network has been taken offline, the incoming sensor data to the server from the remaining sensor nodes will be modeled (FU 5) to be used as an input for the virtual sensor (FU 7) contained on the server that will be activated once it is determined that a sensor node has stopped communicating with the server.

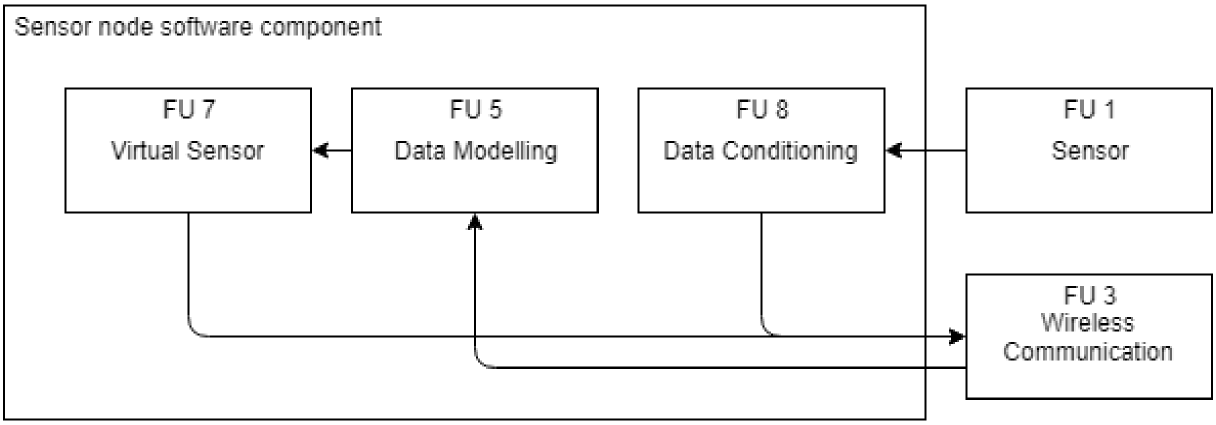

Figure 3. Overview of the sensor node software component.

The sensor node software component, shown in figure 3, will consist of two modes of operation. The first and default mode of operation will consist of the sensor node taking environmental readings using the attached sensor module (FU 1) and using digital filtering algorithms to condition the read data (FU 8). This data is then uploaded to the server using a wireless communication link (FU 3) to be stored in the server's database.

The second mode of operation will consist of the sensor node imputing environmental readings by activating the virtual sensor (FU 7) if the node detects that no sensor is attached to the processing unit. Readings from the other sensors in the network will be passed to the sensor node by the server through the wireless communication link (FU 3) and then modeled so that the data can be used as input data for the virtual sensor. The imputed sensor data from the virtual sensor is then uploaded to the server.

4. Target Specifications

This section contains the main specifications of the proposed project.

4.1 Mission-critical system specifications

The mission critical systems specifications are given in Table 1 below.

SPECIFICATION (in measurable terms)	**ORIGIN** (or motivation of this specification)	**VERIFICATION** (how will you confirm that your system complies with the specification?)
The virtual sensors should impute values, at minimum, within 30% accuracy of the actual observed values for 90% of observed data. The imputation algorithms to be implemented are the k-NN algorithm and a multilayered perceptron.	[2] and [3] both show that, on average, the imputed values from machine learning algorithms are within 30% of the actual observed values on average and within 15% in an absolute best case scenario.	The imputed values from the virtual sensors will be compared to the physical sensor readings.
The virtual sensors should not stray further than 1 standard deviation from the mean values that would be sensed by a physical sensor.	A standard deviation greater than 1 would indicate that the virtual sensor has been improperly trained and will not be of practical use. [7] suggests a standard deviation of not higher than 1 would provide a good indication of a well trained system.	Each virtual sensor node will be tested by turning off the corresponding sensor nodes ability to physically sense the environment as well as completely turning off a sensor node to ensure that the server can activate the corresponding virtual sensor. If the virtual sensor can impute the data that would be sensed by the physical sensor within the specified standard deviation from the mean, it would be deemed functioning correctly.

Imputation of a single value should take no longer than 7 seconds when running on the server or when activated on a sensor node once the data has been input into the virtual sensor.	Imputation of single values takes up to 7 seconds in [1] depending on the imputation technique being used and this is due to the computations being done by the processor. To prevent congestion due to server processes taking place, some time must be given for these computations to take place before new values are imputed.	A timer will be setup between the time that the virtual sensor has been given the input and the time that the virtual sensor gives an output. This will determine the time it takes to impute a value on the server for the given algorithm.

Table 1. Mission-critical system specifications

4.2 Field conditions

REQUIREMENT	**SPECIFICATION** (in measurable terms)
The sensor nodes must be directly connected to the server.	The connection will either need to work over Wi-Fi or Bluetooth.
The server must run on a computer capable of processing large streams of data.	Any modern computer with a CPU speed of at least 2 GHz.

Table 2. Field conditions

5. Deliverables

5.1 Technical deliverables

Table 3 shows the technical deliverables that will be required to complete the project.

DELIVERABLE	DESIGNED AND IMPLEMENTED BY STUDENT	OFF-THE-SHELF
Sensor modules for the processing units.		X
Processing units for the sensor nodes.		X
Wireless communication modules.		X
Desktop PC for the server.		X
Desktop monitor.		X
Software libraries for accessing the database.		X
Processing unit sensor control software and algorithms.	X	
Processing unit digital filter software and algorithms.	X	
Processing unit communication protocol.	X	
Database on the server to store the data received from the physical sensors.	X	
Server communication protocol.	X	
Imputation algorithms using machine learning in MATLAB/Python for simulation.	X	
Imputation algorithms using machine learning implemented on the sensor node processing units.	X	
Imputation algorithms using machine learning implemented on the server.	X	
Data modeling software and algorithms on the server.	X	
Data modeling software and algorithms on the sensor nodes.	X	
Data filtering software and algorithms on the sensor nodes.	X	
Human-Machine interface for the PC application.	X	

Table 3. Deliverables

5.2 Demonstration at the examination

1. All the software will be pre-loaded onto the sensor nodes and the server.

2. The physical system will then be demonstrated by switching on all the sensor nodes and the server and activating the system.

3. The Human-Machine Interface will be used to illustrate the outputs of all the sensors in the system.

4. A sensor on one of the nodes will be disabled to demonstrate the ability of a sensor node to activate a virtual sensor if the physical sensor fails and the values compared to what was being read moments before thus illustrating the imputation method.

5. One of the sensor nodes will be powered down to simulate a sensor node that has completely malfunctioned and stopped communicating to demonstrate the corresponding virtual sensor activating on the server to replace the malfunctioning sensor node.

6. References

[1] Y. Li and L. Parker, "Nearest neighbour imputation using spatial-temporal correlations in wireless sensor networks," *Information Fusion*, vol. 15, pp. 64–79, 2014.

[2] J. Jerez, I. Molina, P. Garcia-Laencina, E. Alba, N. Ribelles, M. Martin, and L. Franco, "Missing data imputation using statistical and machine learning methods in a real breast cancer problem," *Artificial Intelligence in Medicine*, vol. 50, no. 2, pp. 105–115, 2010.

[3] P. Garcia-Laencina, J. Sancho-Gomez, A. Figueiras-Vidal, and M. Verleysen, "K nearest neighbours with mutual information for simultaneous classification and missing data imputation," *Neurocomputing*, vol. 72, no. 7-9, pp. 1483–1493, 2009.

[4] L. Li, J. Zhang, F. Yang, and B. Ran, "Robust and flexible strategy for missing data imputation in intelligent transportation system," *Intelligent Transport Systems*, vol. 12, no. 2, pp. 151–157.

[5] K. K. Nalanda, "Using fuzzy c means and multi layer perceptron for data imputation: Simple v/s complex dataset," in *3rd International Conference on Recent Advances in Information Technology*. IEEE, March 2016.

[6] Y. Li and L. Parker, "spatial-temporal imputation technique for classification with missing data in a wireless sensor network," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*. IEEE, 2008.

[7] A. Wang, Y. Chen, N. An, J. Yang, L. Li, and L. Jiang, "Microarray missing value imputation: A regularized local learning method," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. IEEE, 2018.