PRIVACY ASSURED DATA FUSION IN VIRTUAL SENSOR NETWORK

A PROJECT REPORT

Submitted by

VIGNESHKUMAR A

in partial fulfillment for the requirement of award of the degree of

MASTER OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING
Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY,
COIMBATORE 641 049
(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2013

PRIVACY ASSURED DATA FUSION IN VIRTUAL SENSOR NETWORK

A PROJECT REPORT

Submitted by

VIGNESHKUMAR A

in partial fulfillment for the requirement of award of the degree of

MASTER OF ENGINEERING

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY,
COIMBATORE 641 049
(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2013

BONAFIDE CERTIFICATE

Certified that this project work titled **"PRIVACY ASSURED DATA FUSION IN VIRTUAL SENSOR NETWORK"** is the bonafide work of Mr. VIGNESHKUMAR A, who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other students.

Prof.N.Jayapathi M.Tech.,	Mr.S.P Siddique Ibrahim M.E.,
PROFESSOR AND HEAD	SUPERVISOR
Department of Computer Science and Engineering	Assistant Professor
Kumaraguru College of Technology	Department of Computer Science and Engineering
Coimbatore – 641 049	Kumaraguru College of Technology
	Coimbatore – 641 049

Submitted for the Project Viva-Voce examination held on _____

---------- ----------

Internal Examiner External Examiner

ABSTRACT

Recent advances enabling the deployment of large scale collaborative Wireless Sensor Networks (WSNs). Numerous factors point to the need for sensor network deployments where subsets of nodes cooperate on specific tasks or networks supporting multiple applications. Virtual Sensor Networks (VSNs) is a concept to provide such support for the formation, usage, adaptation and maintenance of subsets of sensors collaborating on a specific task. Virtual sensor provides cost effective solutions, flexibility and ensure security. Clustering is done to form virtual senor network and perform other VSN support function. Cluster head transfers the sensed data to the base station. The Wireless Sensor Network performs aggregation in sensor networks and offer mostly standard mathematical operators over homogeneous data types. The VSN always provides on-demand based access of local data, which aggregates all the sensor data within the level and dynamically updates the cluster head. Most aggregation algorithms and systems do not include any provisions for security in sensor data and consequently these systems are vulnerable to a wide variety of attacks. In particular, compromised nodes can be used to inject false data that may lead to incorrect aggregates being computed at the base station. This system addresses the security vulnerabilities and aggregation methods by applying security algorithm and presents an aggregation result that is resilient to false data injection attacks.

.

ஆய்வுச்சுருக்கம்

அண்மை முன்னேற்றங்கள் கம்பியில்லா உணரி வலையமைப்பில் பெரிய அளவிலான கூட்டு கம்பியில்லா உணரி வலையமைப்பை பயன்படுத்த எண்ணற்ற காரணிகள் இருந்தாலும் ஒரு சில ஆயத்தங்கள் குறிப்பிட்ட பணிகளை பல செயல்பாட்டுகளில் செயல்பட ஒத்துழைக்கிறது. மெய் நிகர் உணரி வலையமைப்பு என்பது ஒரு குறிப்பிட்ட பணியில் குறிப்பிட்ட உணரியை இணைத்து செயல்படுத்துவதற்கான வழிமுறைகளை உருவாக்கவும், பயன்படுத்தவும், பராமரிக்கவும் ஆதரவு வழங்க ஒரு கருத்து உள்ளது. மெய் நிகர் உணரி செயல்முறை நிலைமைகளை கணித மாதிரிகள் மற்றும் சில நேரங்களில் இயற்பியல் மாதிரிகள் துணை கொண்டு மதிப்பிடுகிறது. மெய் நிகர் உணரி செலவு தீர்வுகள், நெகிழ்வு, பன்முகத்தன்மை, ஊக்குவிப்பு, பாதுகாப்பு உறுதி மற்றும் அதிகரிப்பு திறன் ஆகியவற்றை மேன்மையாக வழங்குகிறது. மெய் நிகர் உணரியை உருவாக்கியதன் செயல்பாடுகளை செயல்படுத்த கலந்தாய்வு சார்ந்த இயக்குமுறை துணை புரிகிறது. மெய்நிகர் உணரி எப்போதும் உள்ள அனைத்து உணரிகளின் தரவு மற்றும் மாறும் கொத்து தலைவர் ஆகிய நிலைமைகளைக் கொண்டு உள்ளூர் தரவை தேவைக்கேற்ப அணுக துணை புரிகிறது. மிக அதிக ஒருங்கிணைப்பு நெறிமுறைகள் மற்றும் அமைப்புகள் பாதுகாப்பு வழிமுறைகள் கம்பியில்லா வலையமைப்பில் சேர்க்கப்படாத்தால் அது பலதரப்பட்ட தாக்குதலுக்கு உள்ளாகிறது. குறிப்பிட்ட சமரச முனைகள் தவறான தரவுகளை புகுத்துவதால் முதன்மை நிலையத்தில் தவறான கூட்டை வழிவகுக்கிறது. இந்த அமைப்பு பாதுகாப்பு வழிமுறையை பயன்படுத்துவதன் மூலம் பாதுகாப்பு குறைபாடுகள் மற்றும் ஒருங்கிணைத்தல் முறைகளை அணுகி தவறான தரவைப் புகுத்தும் தாக்குதலுக்கு உள்ளாக்காமல் வழிவகுக்கிறது.

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for enabling me to complete this project.

I express my profound gratitude to **Padmabhusan Arutselvar Dr.N.Mahalingam, B.Sc., F.I.E , Chairman**, **Dr.B.K. Krishnaraj Vanavarayar**, **Co-Chairman, Mr. M. Balasubramaniam, M.Com., M.B.A, Correspondent, Mr.Sankar Vanavarayar, M.B.A., PGDIEM** , **Joint Correspondent** and **Dr.S.Ramachandran Ph.D., Principal** for providing the necessary facilities to complete my thesis.

I take this opportunity to thank **Prof.N.Jayapathi M.Tech**., Head of the Department, Department of Computer Science and Engineering, for his support and timely motivation. Special thanks to my Project Coordinator **Dr.V.Vanitha M.E., Ph.D.**, Senior Associate Professor, Department of Computer Science and Engineering, and project committee members for arranging brain storming project review sessions.

I register my sincere thanks to my Guide **Mr.S.P.Siddique Ibrahim M.E.,** Associate Professor, Department of Computer Science and Engineering. I am grateful for his support, encouragement and ideas. I would like to convey my honest thanks to all **Teaching** and **Non Teaching Staff** members of the department and my classmates for their support.

I dedicate this project work to my **Parents** for no reasons but feeling from bottom of my heart that without their love this work would not be possible.

-VIGNESHKUMAR A

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
ARRA	American Recovery and Reinvestment Act
BPN	Back Propagation Network
BS	Base Station
DFRS	Decision Fusion Rule based on Statistics
DFRW	Decision Fusion Rule based on Weight
DOM	Document Object Model
DoS	Denial of Service
LCG	Linear Congruiential Generator
PL	Procedural Language
QoS	Quality of Service
RFID	Radio Frequency Identification
RSA	Rivest Shamir Adleman
SNR	Signal to Noise Ratio
SQL	Structured Query Language
VSN	Virtual Sensor Network
WSN	Wireless Sensor Network
XML	Extensible Markup Language

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF SENSOR NETWORKS

A sensor network is composed of a large number of sensor nodes that are densely deployed in close proximity to the phenomenon to be monitored. Each of these nodes collects data and to route this information back to a sink. The network must possess self-organizing capabilities since the positions of individual nodes are not predetermined. Cooperation among nodes is the dominant feature of this type of network, where groups of nodes cooperate to disseminate the information gathered in their vicinity to the user.

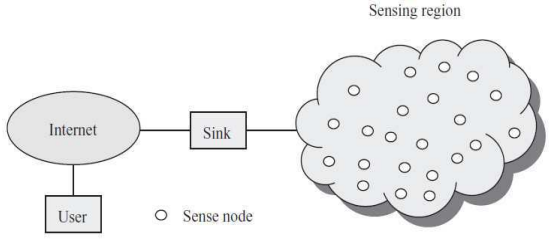

Fig 1.1 Sensor Network Architecture

A Wireless Sensor Network (WSNs) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, humidity, motion or pollutants and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional and enabling control of sensor activity. Architecture of a sensor network is shown in Fig 1.1. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. Today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring and so on.

The WSNs is built of nodes from a few to several hundreds or even thousands, where each node is connected to one or sometimes several sensors. Each such sensor network node has typically several parts. A radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning motes of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth.

Virtual Sensor Networks (VSNs) is an emerging form of collaborative wireless sensor networks. In contrast to early VSNs that were dedicated to a specific

application, VSNs enable multi-purpose, collaborative and resource efficient WSNs. The key idea difference of VSNs is the collaboration and resource sharing. By doing so nodes achieve application objectives in a more resource efficient way. These networks may further involve dynamically varying subset of sensor nodes and users. VSNs can form by providing logical connectivity among collaborative sensors. Nodes can group into different VSNs based on the phenomenon they track or the task they perform. VSNs are expected to provide the protocol support for formation, usage, adaptation, and maintenance of a subset of sensors collaborating on a specific task. Even the nodes that do not sense the particular event could be part of VSNs as far as they are willing to allow sensing nodes to communicate through them.

1.1.1 Sensor nodes

Sensor nodes are small, inexpensive, low-power, distributed devices, which are capable of local processing and wireless communication capability. Each sensor node is capable of only a limited amount of processing. But when coordinated with the information from a large number of other nodes, they have the ability to measure a given physical environment in great detail. Previously, sensor networks consisted of a small number of sensor nodes that were wired to a central processing station. However, nowadays, the focus is more on wireless, distributed and sensing nodes.

1.1.2 Components of Sensor Node

A sensor network consists of sensor nodes which are small, lightweight and portable and these nodes form a network by communicating with each other directly or through other nodes. One or more nodes among them will serve as sink

that are responsible of communicating with the user either directly or through the existing wired networks. The main components of a sensor node are microcontroller, transceiver, external memory, power source and one or more sensors. Every sensor node consists of a transducer, microcomputer, and transceiver and power source. The transducer Analog to Digital Converter (ADC) in Fig 1.2 is responsible to generate electrical signals based on sensed phenomena and physical effects. The microcontroller's work is to process and store the sensor output. The transceiver receives commands from a central computer or base station and transmits data to the computer or station. Sensor nodes are catered power with a battery. Some sensor nodes include an external memory which may be on-chip memory of a microcontroller and flash memory. Needs of memory on a sensor node are application specific. Each node may also belong to two extra components like location finding system and mobilizer. The location finding system is required for the user may in need of location with high accuracy, and the mobilizer may be needed to move sensor nodes to carry out the assigned tasks.

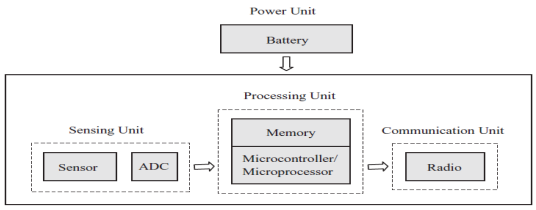

Fig 1.2 Components of a sensor node

Sensor nodes are fitted with an onboard processor. Instead of sending the raw data to the nodes responsible for the fusion, they use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data.

i. Sensing Unit

The main functionality of the sensing unit is to sense or measure physical data from the target area. The analog voltage or signal is generated by the sensor corresponding to the observed phenomenon. The continual waveform is digitized by an ADC and then delivered to the processing unit for further analysis. The sensing unit is a current technology bottleneck because the sensing technologies are much slower than those of the semi-conductors.

ii. Processing Unit

The processing unit which is generally associated with a small storage unit manages the procedures that make the sensor nodes collaborate with the other nodes to carry out the assigned sensing tasks.

iii. Transceiver

There are three deploying communication schemes in sensors including optical communication (Laser), infrared, and Radio-Frequency (RF). Laser consumes less energy than radio and provides high security, but requires line of sight and is sensitive to atmospheric conditions. RF is the most easy to use but requires an antenna.

Various energy consumption reduction strategies have been developed such as modulation, filtering and demodulation.

iv. Power Unit

One of the most important components of a sensor node is the power unit. Every sensor node is equipped with a battery that supplies power to remain in active mode. Power consumption is a major weakness of sensor networks. Any energy preservation schemes can help to extend the sensor's lifetime. Batteries used in sensors can categorize into two groups; rechargeable and non-rechargeable.

1.2 DESIGN CHALLENGES OF SENSOR NETWORK

The unique network characteristics present many challenges in the design of sensor networks, which involve the following main aspects

i. Limited Energy Capacity

Sensor nodes are battery powered and thus have very limited energy capacity. This constraint presents many new challenges in the development of hardware and software and the design of network architectures and protocols for sensor networks. To prolong the operational lifetime of a sensor network, energy efficiency should be considered.

ii. Limited Hardware Resources

Sensor nodes have limited processing and storage capacities, and thus can only perform limited computational functionalities. These hardware constraints present many challenges in software development and network protocol design for sensor networks, which must consider not only the energy constraint in sensor nodes, but also the processing and storage capacities of sensor nodes.

iii. Massive and Random Deployment

Node deployment is usually application dependent, which can either manual or random. In most applications, sensor nodes can scattered randomly in an intended area or dropped massively over an inaccessible or hostile region. The sensor nodes must autonomously organize themselves into a communication network before they start to perform a sensing task.

iv. Dynamic and Unreliable Environment

A sensor network usually operates in a dynamic and unreliable environment. On one hand, the topology of a sensor network may change frequently due to node failures, damages, additions, or energy depletion. On the other hand, sensor nodes are linked by a wireless medium, which is noisy, error prone, and time varying. The connectivity of the network may be frequently disrupted because of channel fading or signal attenuation.

v. Diverse Applications

Sensor networks have a wide range of diverse applications. The requirements for different applications may vary significantly. No network protocol can meet the requirements of all applications. The design of sensor networks is application specific.

1.3 APPLICATIONS

VSNs are useful in three major classes of applications. Each application uses nodes of the other application to relay its data to the signaling systems and to its members.

i. Geographically overlapped applications

The VSNs is mainly used in Monitoring rock slides and animal crossing within a mountainous terrain. Different types of devices that detect these phenomena can relay each other for data transfer without having to deploy separate networks. The advantage is saving the cost of hardware.

ii. Water/Wastewater Monitoring

Facilities not wired for power or data transmission can monitor using industrial wireless I/O devices and sensors powered using solar panels or battery packs. As part of the American Recovery and Reinvestment Act (ARRA), funding is available for some water and wastewater projects in most states.

iii. While logically separating multi-purpose sensor networks

The VSNs is used in Smart neighboring systems with multifunctional sensor nodes. Instead of traditional WSNs that run one single application, the VSNs enable nodes to run multiple applications.

iv. In certain dedicated applications

To enhance efficiency of a system that track dynamic phenomena such as subsurface chemical plumes that migrate, split, or merge. Such networks may involve dynamically varying subsets of sensors. The advantage is the ability to connect right set of nodes at the right time.

v. Landfill Ground Well Level Monitoring and Pump Counter

Wireless sensor networks can use to measure and monitor the water levels within all ground wells in the landfill site and monitor leach ate accumulation and removal. A wireless device and submersible pressure transmitter monitor the leach ate level. The sensor information is wirelessly transmitted to a central data logging system to store the level data, perform calculations, or notify personnel when a service vehicle is needed at a specific well.

1.4 LITERATURE SURVEY

1.4.1 DATA FUSION

Data fusion combines several sources of raw data to produce new raw data. The expectation is that fused data is more informative and synthetic than original inputs.

1.4.1.1 Time-Slotted Voting Mechanism for Fusion Data Assurance in Wireless Sensor Networks under Stealthy Attacks

In (Hung 2010), the authors investigate the problem of data fusion assurance in multi-level data fusion or transmission. In WSNs, data fusion was often performed in order to reduce the overall message transmission from the sensors toward the base station. Different to a recent approach of direct voting where the base station polls other nodes directly regarding to the received fusion result, proposed scheme uses the time-slotted voting technique. In this scheme, each fusion node broadcasts its fusion data or vote during its randomly assigned time slot. Only the fusion result with enough number of votes will be accepted. Thus, this scheme eliminates the polling process and eases the energy consumption burden on the base station or the fusion data receiver, which could well be the intermediate nodes.

In the direct voting mechanism, the base station receives the fusion data from a randomly chosen fusion node and then polls other sensor nodes, as witnesses, for votes on the fusion data. The vote is exploited to verify the correctness of the fusion data and the base station consumes some extra power to verify the data fusion results. In WSNs with multi-level of data fusion or

transmission, however, the intermediate fusion nodes do not have such extra power to verify the incoming fusion results. Therefore, the direct voting mechanism will not work well in WSNs with multi-level of data fusions or transmission.

Time-slotted voting scheme combines the concept of time division multiple access with direct voting mechanism. Fig 1.3 represents data fusion in time slotted mechanism. Each fusion node is assigned a time slot to broadcast its fusion result, eliminating the polling process completely.

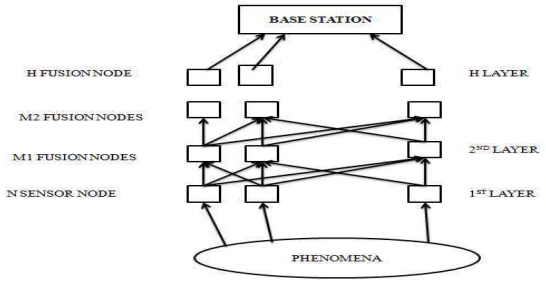

Fig 1.3 Data fusion in time slotted mechanism

The algorithm works in 3 phases

- Transmission schedule: Gives the order of the time slots.
- Promiscuous mode: Data broadcasting techniques are adopted in this scheme because the data must be received at all fusion nodes of all layers.

Advantages

Low transmission, short verification delay, exclusion of compromised nodes and no extra memory required. Reject of forged result when the number of compromised nodes is less than the number of required support votes. Every data fusion node has the same task and the assurance process in each hop is performed cooperatively by a group of sensors. Thus, data fusion assurance can implement in the WSNs easily.

Limitations

The overhead doubles when the number of the disconnected nodes increases from one to two.

1.4.1.2 Multi-Sensors Data Fusion System for Wireless Sensors Networks of Factory Monitoring Via BPN Technology

In (Wen-Tsai Sung 2010), the authors presented a Back-Propagation Network (BPN) for multi-sensors data fusion in WSNs with a node-sink mobile network structure. Wireless sensor network circuits include temperature, humidity, ultraviolet, and illumination four variable measurement components. These data fields of each sensor node contain the properties and specifications of that signal process rule, the remote engineers can manage the multi-sensors data fusion using the browser, and the WSNs system then classify the data fusion database via the Internet and mobile network. Moreover, The BPN training approach is significant that improves data fusion system in accuracy and classification with parallel computing for data fusion efficiency. The final phase of the classification fusion

system applies parallel BPN technology to process data fusion, and can solve the problem of various signal states. The algorithm works as

Set up the network parameters.Set up weighted matrixes, i.e., W_xh and W_hy, and the initial values of bias vector, i.e., ɵ_h and ɵ_y, by uniform random numbers.

- Calculate the output quantity of the hidden layer. Set up the tolerant difference quantity between the output layer and the hidden layer. Calculate the difference quantity, i.e., d, between the output layer and the hidden layer.
- Determine whether the different quantity between the output layer and the hidden layer is greater than the tolerance difference quantity U. If the difference quantity d is smaller than the tolerant difference quantity U, the optimal regression model is obtained..

Finally, compare the correlation of sensitivity correction to find out the optimal regression model. Fig 1.4 represents BPN technique for estimation of intensities at neighboring nodes. Fig 1.5 shows the data fusion model.

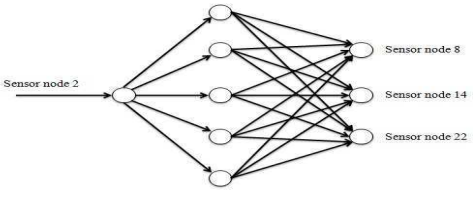

Fig 1.4 BPN for estimation of intensities at neighboring nodes

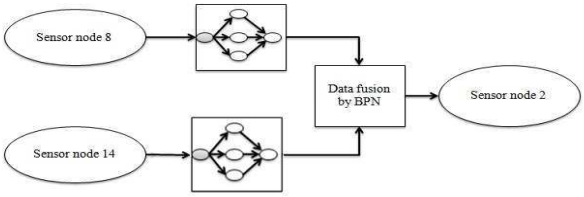

Fig 1.5 BPN based data fusion model

Advantages

This method partitions the frequency band in different resolution to distinguish the difference in a low - frequency band and reduces the feature dimensions greatly. The extracted feature expresses the stable classification rate for a different moving condition.

Limitations

Major limitations included limited storage and power. The only special node can connect to a computer and outside the network.

1.4.1.3 Decision Fusion Rules Based on Multi-Bit Knowledge of Local Sensors in Wireless Sensor Networks

In (Varshney 2005), the authors proposed distributed detection and decision fusion rules based on multi-bit knowledge of local sensors. At local

sensors, observations are quantized to multi-bit local decisions. Three quantification algorithms are investigated, which are based on weight, statistics and redundancy, respectively. Corresponding suboptimal fusion rules of the fusion center are also discussed by approximating the optimal likelihood ratio test. System level detection performance measures, namely probabilities of detection and false alarm, are derived analytically by employing probability theory. Finally, Monte Carlo (MC) methods are employed to study the performance of proposed decision fusion rules with parameters such as Rayleigh fading channel and Gaussian noise. Numerical results show that, under non-ideal channel, commonly used schemes based on weight cannot improve the system performance even with a large number and high SNR (Signal to Noise Ratio). Fortunately, schemes based on statistics and redundancy can enhance the system capability when the node is deficient and SNR is low. Furthermore, schemes based on statistics have the best stability among the three schemes, and schemes based on redundancy have the best performance among the three when quantization degree is high.

Fusion rules in WSNs can divided into two steps. First of all, observations are collected and then retreated to form local decisions by local sensor nodes. Secondly, the local decisions are transmitted to the fusion center. The final decision was made in the fusion center by local decisions from all local nodes.

All local sensor nodes have the same capability parameters, thus pdi = pd, pfi = pf, where i = 1, 2,. . . , N. Secondly, all local sensor nodes employ the same decision rules. Finally, the fusion center trusts all local sensor nodes indiscriminatingly. Decision Fusion Rule based on Weight (DFRW). The number of 1's in Xi is transformed into a binary system, thereby the high bit is twice the

weight of lower neighbor bit. Decision Fusion Rule based on Statistics (DFRS**).** Although DFRW can improve the coding efficiency, the effects on fusion center from each bit in Ui were still direct ratio with weight of the bit. Thus the detection performance was unstable under very low SNR. The Decision fusion rule based on statistics can avoid this problem, because of the same weight of each bit. Decision Fusion Rule based on Redundancy (DFRR) - The trade-off between coding efficiency and detection performance was considered. The bits with higher weight were repeatedly coded, and the stability of performance was maintained in the decision fusion rule based on redundancy.

Advantages

Under a non-ideal channel, fusion rules based on the commonly used weight cannot improve the system performance even with a large number and high SNR. Fusion rules based on statistics and redundancy can enhance the system capability when the node was deficient and SNR was low.

1.4.2 VIRTUALISING SENSOR

Virtualization of sensor was a brand new research approach demonstrates an approach of using the WSNs resources in a sensor Virtualization environment.

1.4.2.1 TinyML: Metadata for Wireless Networks

In (Nathan Ota 2005), the authors proposed TinyML, which has the important concept of virtualizing physical components. First, when associated with a platform, a virtual sensor or actuator can created from physical devices. For

example, if a platform has a thermistor that provides voltage readings as an output, a virtual sensor could be defined that would use the platform's processor to take thermistor output and, using calibration information, transform it to Celsius or Fahrenheit responses. Virtual devices can also be a collection of sensor outputs or actuator actions. There are two major types of virtual sensors/actuators. Those focused on platforms and those focused on sensor fields. Platform virtual sensors/actuators are associated only with basic sensors and/or actuators on a physical platform. For example, sensor field can have a virtual sensor or actuator associated with it. A field virtual sensor is an aggregate virtual sensor that can take readings from all the same sensors in the field and use a function such as average, maximum, or minimum as possible virtual sensor output. Virtual sensors can also be associated with groups of sensors in the sensor field. This creates subgroups of platforms that use a function to develop a composite value. For instance, consider a sensor field throughout a building. A field virtual sensor could be the temperature sensors in a room providing a single temperature reading for the room. A more sophisticated virtual sensor function might use temperature differences to determine whether there is a person in the room.

Virtual sensors have a function associated with them. The function defines how a virtual sensor takes data from basic physical sensors and transforms it to become the output of the virtual sensor. These functions have basic operators, such as add subtract, divide and multiply, as well as Boolean operators for and, or, not, etc. Further, it is possible for a virtual sensor to support intrinsic functions such as max, average, etc. The operators and functions can performed by the physical platform or by the sensor network proxy that provides external network connectivity for the sensor network. The actual implementation details will vary for operators and functions and could be a predefined list that the sensor network

supports, a set of library calls or, in more complex functions, over the net programming. The implementation is independent of TinyML since it is a feature of the network specific part of the TinyML proxy.

Advantages

Used for virtualizing physical components. TinyML was conceived and created in order to explore the interface between sensor nets and the internet services in a way the appropriate partitions and assigns data synthesis and control in a most effective manner.

Limitations

Implementing the transparent solution requires placing xml into the sensor network itself, which means each platform must have the capability to parse, interpret, respond and formulate XML (Extensible Markup Language). Creating a DOM (Document Object Model) on each node is not practical with current sensor/node limitations. There would be significant increases in network traffic and power utilized in the implementation which is counter to the design goals of most embedded networks.

1.4.2.2 VIRTUAL SENSORS: A DEMONSTRATION

In (Kabaday 2007), the authors described an application demonstration of a new abstraction that supports heterogeneous in-network aggregation over networks of resource-constrained devices. For the purpose of demonstration, this work can draw upon an intelligent job site application for the construction domain. The demonstration was carefully constructed to highlight the ability of this

approach to provide on-demand access to local data, aggregation of data across dynamically defined regions, and fusion of heterogeneous sensor data. The work briefly described the abstraction and the details of the demonstration. Currently, existing deployments of sensor networks are application specific, where the nodes were statically deployed for a particular task. Target pervasive environments in which the applications that will be deployed were not known a priori and may include varying sensing needs and adaptive behaviors.

The authors developed virtual sensors, which enable collecting low-level sensed data and transforming it to a more abstract measurement to relay to the user. This virtual sensor abstraction connects users like construction workers in a dynamic and unsafe environment, responders in an emergency, directly to sensors in the local environment. Virtual sensors can deployed on small devices, operate independently of heavyweight infrastructure, and provide on-demand, real-time connection with information that enables users to complete their tasks quickly, safely, and with the best possible information.

This has created a unique abstraction that enables domain experts to create customizable applications for dynamic networks involving resource constrained sensing devices. Specifically, this approach highlights the following three capabilities not found in today's sensor network coordination frameworks:

- On-demand local data access: Contrary to existing applications, the users in these targeted scenarios are immersed in the sensor network. As such, we target new communication and query protocols that allow users to opportunistically interact with locally available sensors.

- Regional aggregation: Emerging applications must intelligently aggregate data from a particular dynamic changing region. This requires novel algorithms for group communication and aggregation.

Demonstration work uses an application example from the domain of the intelligent construction site. In this scenario, the user requests information about the region around the base of a crane where it is unsafe to walk or drive. In this case, a virtual sensor is constructed that dynamically discovers physical sensors attached to components of a nearby tower crane like the base of the crane, the trolley along the boom, the counterweight of a crane. The virtual sensor combines the information collected from these distributed physical sensors to calculate the requested abstract data type like a dangerous circle calculated using location estimates from sensors attached to the crane. Once these sensors were discovered, the virtual sensor registers persistent queries on these particular sensors and remains connected to them.

Advantages

The virtual sensor has the potential to enable domain programmers to rapidly create expressive applications without having to directly handle the complexities associated with distributed programming.

1.4.2.3 VIRTUAL SENSORS: ABSTRACTING DATA FROM PHYSICAL SENSORS

In (Pridgen 2006), the authors proposed the virtual sensor abstraction that enables an application developer to programmatically specify an application's high-level data requirements.

1.4.2.3.1 Virtual Sensor Model

A client application runs with the support of the virtual sensor abstraction. Defines how a developer defines the application's data requests using the virtual sensors and how programs dynamically interact with data from virtual sensors through an intuitive programming interface Fig 1.6 depicts the connection to an application-defined virtual sensor represented by the dashed blue ellipse on a tower crane.

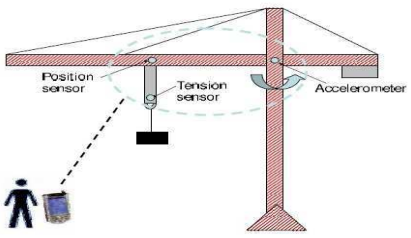

Fig 1.6 Connections to Virtual Sensor on a Tower Crane

This virtual sensor uses data from three physical sensors represented by blue dots. The virtual sensor aggregates the information from these sources into a higher level reading that represents the effective load on the crane. The software implementing this virtual sensor may run on the user's handheld device or, to reduce network communication, be deployed to one of the sensors depicted.

A virtual sensor's declarative specification allows a programmer to describe the behavior user wants to create, without requiring him to specify the underlying details of how it should be constructed. That was especially important considering the fact that in an instrumented sensor network, a user's operational context is highly dynamic. While the actual data sources change over time based on the user's location and movement, the application's data need do not change as much, and one of the benefits of a virtual sensor was that it hides the changes in data sources from the application. This approach assumes applications and sensors share knowledge of a naming scheme for the low-level data types the sensor nodes can provide location, temperature. These data types were determined by the types of sensors deployed in a network. The programmer needs to specify the following four parameters for the virtual sensor.

- Input data types: Physical low-level data types required to compute the desired abstract measurement.
- Aggregator: A generic function defined to operate over the specific heterogeneous input data types to calculate the desired measurement.
- Resulting data type: The abstract measurement type that is a result of the aggregation.

The types of queries enabled on a virtual sensor can classified into One-time queries (which return a single result from the virtual sensor), and Persistent queries (which return periodic results from the virtual sensor). To support these types, these works provide two different methods for posing queries.

- Query ()
- Register ()

When the application needs to query the sensor network for a data type that is not provided intrinsically by the physical sensors, the developer constructs

and deploys a virtual sensor using his knowledge of the available data types. The application subsequently queries this virtual sensor directly. If the virtual sensor happens to be running remotely, a remote handle to the listener needs to be set up. In such cases, the middleware creates a proxy for the virtual sensor on the user's device. This proxy object runs within middleware and uses a unicast routing protocol to connect to the remote virtual sensor and collect the information as shown in Fig 1.7. When the query's result is ready, this proxy makes a call back to the user's result listener either for a one-time query or a persistent query.

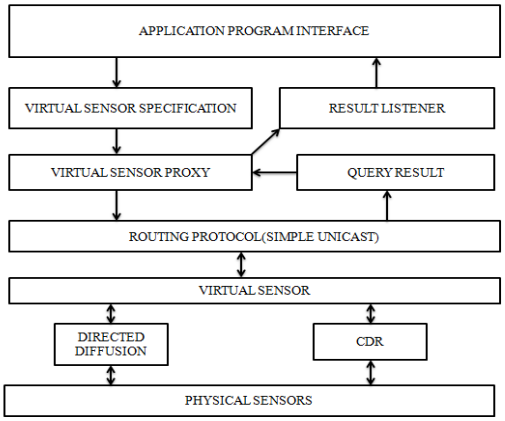

Fig 1.7 Object Diagram For Virtual Sensor Middleware

Advantages

The separation of the specification of the sensing task of the sensing behavior allows a programmer to describe the behavior of a virtual sensor, without having to specify the underlying details of how it should be constructed. Virtual sensors also offer a way to tailor a generic sensing environment to specific applications.

Limitations

Virtual sensors can move with an event of interest or in response to the user's movement.

1.4.2.4 CLUSTER TREE BASED SELF ORGANIZATION OF VIRTUAL SENSOR NETWORKS

In (Dilum Bandara 2010), the authors proposed a cluster tree based mechanism to form VSNs, facilitate inter-VSN and intra-VSN communication, and perform other VSNs support functions. VSN is a concept to provide such support for the formation, usage, adaptation, and maintenance of subsets of sensors collaborating on a specific task.

1.4.2.4.1 Virtual Sensor Network Support Functions

Formation, usage, adaptation, and maintenance of VSNs require implementation of many functions and protocols. These functions and protocols should be able to get new nodes into a VSNs, remove nodes from a VSNs, detect

multiple VSNs, merge VSNs together, split a VSNs into multiple VSNs, and facilitate communication within and across VSNs. Self-organization of VSNs members is the first step. Whenever a node detects a relevant event for the first time, it should send a VSNs formation/discovery message within the network indicating that it is aware of the particular phenomenon and wants to collaborate with similar nodes. The node may join an existing VSN or makes it possible for other nodes that wish to form VSNs, to find it. Therefore, every node that detects a relevant event for the first time executes the following function and informs other nodes about its interest to form/discover VSNs.

Inter-VSN and intra-VSN communication models are application dependant. Unicast messages are required when a message needs to be sent to a specific VSNs member. For example, if a chemical plume is predicted to be moving towards a certain direction, node in that region needs to be informed. Multicast messages are useful in delivering messages to all the members of VSNs. For example, if each node independently calculates the average chemical concentration of a plume, each other's data needs to be shared. When VSNs merge or split, it may be required to inform all the members of existing VSNs, hence broadcast within all the VSNs is also important.

Imposing some structure within the network to effectively achieve the application objectives is an attractive option for the self-organization of large-scale WSNs. Cluster based organization, and arranging clusters in the form of a tree simplifies many higher-level functions and distributed application deployments. Generic Top-down Cluster and cluster tree formation GTC algorithm are used to form a cluster tree within the network.

Algorithm

Handle VSN_Message(msg)
// Initially n=0, m=0
1. if msg.source € my.cluster_members
2. if(msg.type € my_data.VSN)
3. my_data.VSN(n)← msg.type
4. n← n+1
5. forward to parent_CH(msg,my.parent_CH)
6. my_data.VSN_table(m) ←(msg.source, msg.type)
7. m←m+1
8. else
9. if(msg.type € my_data.VSN_table)

A virtual sensor network is an emerging concept that supports collaborative, resource efficient, and multipurpose sensor networks. This work proposed a scheme for the formation of VSN using a cluster tree. The nodes observing the same phenomenon form a virtual tree that connects one or more VSNs. This scheme is much more efficient and reliable than a scheme based on Rumor Routing. The logical tree can facilitate both inter-VSN and intra-VSN communication.

1.4.3 PRIVACY

Privacy provides how securely data can transmit across sensor nodes. Trust and security can increase privacy.

1.4.3.1 Security Co-Existence of Wireless Sensor Networks and RFID for Pervasive Computing

In (Lorincz 2008), the authors proposed a Linear Congruential Generator (LCG) based lightweight block cipher that can meet security coexistence requirements of WSNs and RFID systems for pervasive computing. Recent advances in wireless networks and embedded systems have created a new class of pervasive systems such as Wireless Sensor Networks and Radio Frequency Identification (RFID) systems. WSNs and RFID systems provide promising solutions for a wide variety of applications, particularly in pervasive computing. However, security and privacy concerns have raised serious challenges on these systems.

The lack of physical security combined with unattended operations make sensor nodes prone to a high risk of being captured and compromised. The wireless broadcast nature may result in privacy breaches of sensitive information during data transmission. Therefore, security and privacy issues of WSNs have attracted a lot of research effort. Here listed a brief taxonomy of WSN attacks and their representative solutions.

Physical attacks: Sensor nodes may be left unattended for a long time. Therefore, attackers may have a high chance to compromise WSN nodes. From the hardware perspective, attackers can gain complete access to microcontrollers in sensor nodes and thus obtain sensitive information stored in node memory. From the software perspective, TinyOS, the most widely used Operating System in

WSNs, and various applications may also suffer from well-know exploitations such as buffer overflow. All these enable attackers to extract relevant secrets, and insert malicious data to the network very easily.

Attacks at physical layer: Jamming is one of the most important attacks at the physical layer. Aiming at interfering with normal operations, an attacker may continuously transmit radio signals on a wireless channel. Equipped with a powerful node, an attacker can send high-energy signals in order to effectively block wireless medium and to prevent sensor nodes from communicating. This can lead to Denial-of-Service (DoS) attacks at the physical layers.

Attacks at the link layer: The functionality of link layer protocols, such as those specified in 802.15.4/ZigBee standards, is to coordinate neighboring nodes to access shared wireless channels and to provide link abstraction to upper layers. Attackers can deliberately violate predefined protocol behaviors at the link layer. For example, attackers may induce collisions by disrupting a packet, cause exhaustion of nodes' battery by repeated retransmissions, or cause unfairness by abusing a cooperative MAC layer priority scheme. All these can lead to DoS attacks at the link layers.

Attacks at the network layer: In WSNs, attacks at the routing layer may take many forms. For example, routing control packets exchanged between sensor nodes can spoofed, replayed, or altered. In this way, routing logic can compromise. Data Packets may also be selectively dropped, replayed, or modified by compromised nodes. Besides these, WSNs also suffer from the wormhole and sinkhole attacks, in which messages may be lowered or tunneled to a particular area

through compromised nodes. Attackers may also launch a Sybil attack. Therefore, a single Node may present multiple identities to other nodes in a network.

1.4.3.2 A SECURITY FRAMEWORK FOR WIRELESS SENSOR NETWORKS

In (Yang 2010), the authors proposed a flexible security management framework which combines the existed method to overcome the drawbacks of early research. To deploy the proposed framework on the sensor network to evaluate its performance and the experiment shows this method can achieve a satisfied result. Fig 1.8 shows a security management system framework

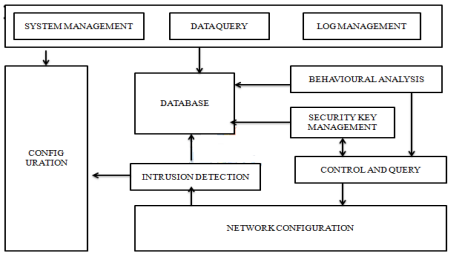

Fig 1.8 Security management system framework

Based on the research interests of the security of WSNs, there are four main points as follows:

- Key management
- Attack detections and preventions
- Security routing
- Secure location

Algorithm

- **Step 1** Receive message from WSNs.
- **Step 2** Update the information of a sensor node (node i).
- **Step 3** Read rule from the configuration module and check the data using the rule, if the result set >0 go to Step 4, else go to Step 1.
- **Step 4** According the detection to make a decision, if the result is equal to 1, goes to and else goes to Step 6.
- **Step 5** The node is compromised, notes the control module to evict it.
- **Step 6** The system need more information, notes query module to get enough data.
- **Step 7** Note the key management module to update the secret key.

Advantages

Each part of WSNs' security can interact on other part, which can enhance the system's security.

CHAPTER 2

IMPLEMENTATION

2.1 DRAWBACKS OF EXISTING SYSTEM

Wireless sensor networks were dedicated to a specific application. They do not share resources as compared to virtual sensor network. Data aggregation was existing on sensor applications which support only standard functions such as MIN, MAX. Virtual sensors are vulnerable to threats and attack. The number of sensors deployed in wireless sensor network is higher than virtual sensor network.

The following are the shortcomings of the existing system:

- Data aggregation gives only summarized value which is inaccurate.
- WSN have cost as drawback as compared to virtual sensor.
- Security is a major concern since intruder can change the values of the sensor while transmitting.
- Updating of sensor values to the base station are based on polling process and periodic update. Relationships between attributes are not specified directly.

2.2 PROPOSED SYSTEM

The objective of the proposed work as shown in Fig 2.1 is to increase security in virtual sensor and provide data fusion technique.

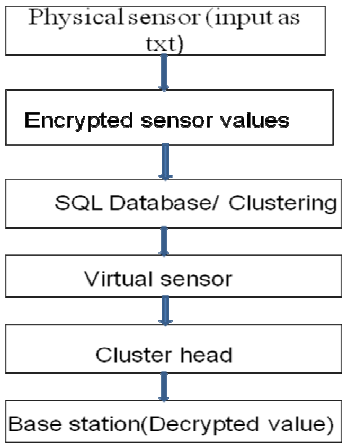

Fig 2.1 Proposed systems

The proposed system consists of 4 main parts. 1. Clustering, 2. Virtual sensor, 3. Data fusion, 4. Security

2.2.1 Clustering

The goal of clustering is to group data that are similar to each other and identify such grouping in an unsupervised manner. Input for virtual sensor is output from physical sensor. All the sensor readings are grouped and send to the base station through a cluster head.

2.2.2 Virtual sensor

A Virtual Sensor estimates process conditions based on physical sensor readings. Virtual sensor gives collaborative information which is accurate.

2.2.3 Data fusion

Data collected from various sensors are fused to give the new information whether any abnormal condition will take place or not.

2.2.4 Security

There is a possibility of intruders to modify the data while transmitting from virtual sensor to cluster head and base station. Security is enhanced to overcome a threat and attack.

2.3 PROJECT DESCRIPTION

2.3.1 PROBLEM DEFINITION

A Virtual Sensor estimates product objective or process conditions using mathematical models and sometimes in conjunction with physical sensors. The observed mathematical models use other physical sensor values to calculate the estimated property or condition. Virtual sensor provides cost effective solutions, flexibility, ensure security, promote diversity, and increase manageability. In this work, data fusion is performed based on clustering technique and privacy gets an increase in virtual sensor. To reduce the overall message transmission from the sensors toward the base station data fusion is used. Clustering is done to form a virtual sensor network and perform other VSN support function. Cluster head transfers the sensed data to the base station. In this work, for the security RSA algorithm is used to overcome a threat and attack.

2.3.2 PROJECT OVERVIEW

In order to group virtual sensor clustering technique is used. Virtual sensors are formed from physical sensor. All sensor readings are fused to give a new result which is more efficient. Security is added while transmitting the data from virtual sensor to the base station.

2.3.3 MODULES

2.3.3.1 Virtual Sensor Formation

In this work, first the physical sensor is converted into virtual sensor. Group formation is based on k-means clustering technique. Later the data collected by sensor are fused based on data fusion technique. There is for possibility of intruders to attack or change the data sent by sensor. Security is increased in order to overcome threat and attack**.** Virtualization provides the ability to do several things. First, when associated with a platform, a virtual sensor or actuator can created from physical devices. For example, if a platform has a thermistor that provides voltage readings as an output, a virtual sensor could be defined that would use the platform's processor to take thermistor output and, using calibration information, transform it to Celsius or Fahrenheit responses.

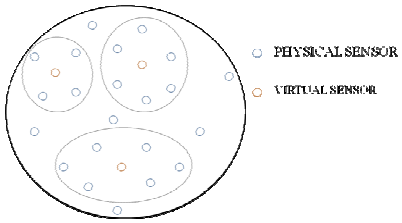

Fig 2.2 Physical Sensors and virtual sensors

Virtual devices can also be a collection of sensor outputs or actuator actions. There are two major types of virtual sensors/actuators: those focused on platforms and

those focused on sensor fields. Platform virtual sensors/actuators are associated only with basic sensors and/or actuators on a physical platform. For example, sensor field can have a virtual sensor or actuator associated with it. A field virtual sensor is an aggregate virtual sensor that can take readings from all the same sensors in the field and use a function such as Average, Maximum, or Minimum as possible virtual sensor output. Virtual sensors can also be associated with groups of sensors in the sensor field. This creates subgroups of platforms that use a function to develop a composite value. For instance, consider a sensor field throughout a building. A field virtual sensor could be the temperature sensors in a room providing a single temperature reading for the room.

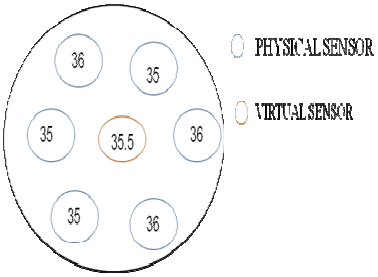

Fig 2.3 Getting values for Virtual sensor

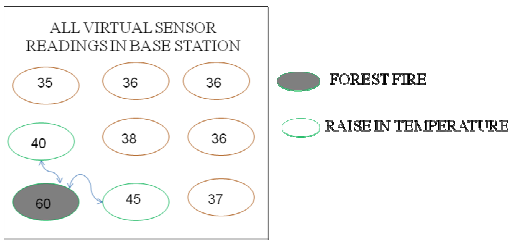

Fig 2.4 Detecting forest fire based on values from Virtual sensor

Consider a sensor network made up of platforms that have two sensors on them, represented in the diagrams as circle. Fig 2.2 shows physical and virtual sensors deployed in forest fire application. Fig 2.3 shows how virtual sensor gets value based on readings from physical sensors. Fig 2.4 shows readings of virtual sensor in base station. Based on these values, it predicts occurrence of forest fire whenever the temperature sensor readings exceed threshold value

2.3.3.2 Cluster Formation

Many high levels function and distributed application deployments get simplified by cluster formation. This work use k mean algorithm as shown in Fig 2.5 to form cluster in virtual sensor network. K means aims to partition n sensors into k clusters in which each sensors belong to cluster with nearest mean. It is an iterative refinement technique.

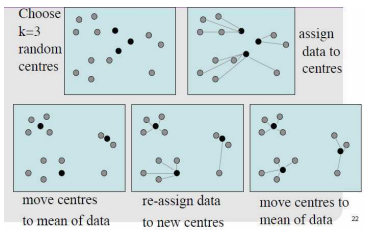

Fig 2.5 K means algorithm working steps

Algorithm

Input: n points and a number k

- Randomly place K points into the space represented by the objects (sensors) that are being clustered. These points represent initial group centroids.
- Assign each sensor to the group that has the closest centroids.
- When all objects have been assigned, recalculate the positions of the K centroids.
- Repeat Steps 2 and 3 until the stopping criteria is met.

Data is sent from physical sensors to cluster head, if any relevant change occurs in clusters, all remaining cluster head combines the data and gives results as a prediction. Cluster head transmits data to base station. Periodically base station gets updated.

2.3.3.3 Enhancing Security

The RSA algorithm can use for both public key encryption and digital signatures. RSA involves a public key and a private key**.** The public key can known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. Its security is based on the difficulty of factoring large integers. A user of RSA creates and then publishes the product of two large prime numbers. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message. It consists of a key generation algorithm, encryption, and decryption.

Algorithm

Key Generation Algorithm

- Generate two large random primes, p and q, of approximately equal size such that their product n=pq is of the required bit length
- Compute n=pq and phi= (p-1) (q-1)
- Choose an integer e, 1< e <phi, such that gcd (e, phi) =1
- Compute the secret exponent d, 1< d< phi, such that ed=1 (mod phi).
- The public key is (n, e) and the private key is (n, d). The values of p, q and phi should also be kept secret.

Encryption

Sender A does the following;-

- Obtains the recipient B's public key (n,e).
- Represents the plaintext message as a positive integer m
- Computes the ciphertext c= m^{Ae} mod n
- Sends the ciphertext c to B.

Decryption

Recipient B does the following

- Uses his private key (n, d) to compute m= c^{Ad} mod n
- Extracts the plaintext from the integer representive m.

CHAPTER 3

RESULTS

3.1SYSTEM SPECIFICATION

3.1.1 Hardware Requirements

Processor	: Pentium IV and above
Clock speed	: 550MHz
Hard Disk	: 20GB
RAM	: 4 GB or above
Cache Memory	: 512KB
Monitor	: Color Monitor
Keyboard	: 104Keys
Mouse	: 3Buttons

3.1.2 Software Requirements

Operating System	: Windows 7
Language	: Java
Back End	: Microsoft SQL Server 2005

3.2 IMPLEMENTATION

The proposed work is implemented using Java as front end and SQL Server as back end. For the evaluation purpose, 517 sensor nodes are deployed randomly. The code is written in the Java file. The readings of physical sensor are read through Java file. The cluster is formed and is updated in the database. The values are encrypted in database. Base station gets periodic updates from the database. The result is displayed when an abnormal condition occurs. Decryption takes place while retrieving data from database.

3.3 ANALYSIS

3.3.1 Efficiency

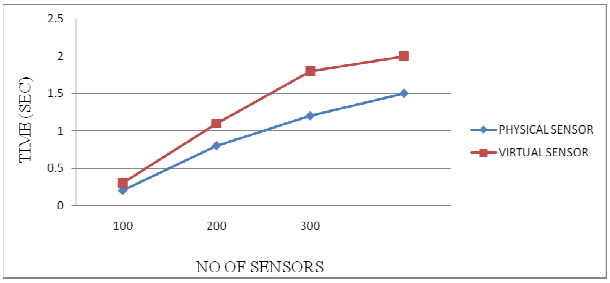

Fig 3.1 Efficiency graph

The response time of physical sensor is higher than the virtual sensor as shown in Fig 3.1. If any abnormal event occurs physical senor will immediately intimate to base station.

3.3.2 Cost metrics

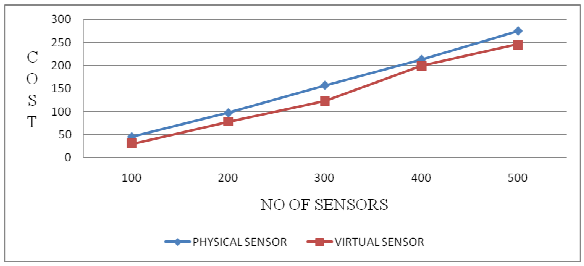

Fig 3.2 cost comparison graph

Virtual sensor will reduce the no of sensors deployed in a particular region. If 100 physical sensors are to sense a particular application, by this concept we will need 80 sensors only. The number of sensor usage will be reduced as shown in Fig 3.2.

3.4 CONCLUSION AND FUTURE ENHANCEMENT

This system enhances with a virtual sensor in the wireless sensor network with adequate security and works with reasonable cost. This system can also consider neighboring node data and periodically updates the base station value in order to make the system work effectively to predict the environmental condition and alert the end user. In future work it is worthwhile to apply more data mining concepts on the virtual sensor network, where the communications between sensors are short and frequent and the computational resource on each sensor is limited.

APPENDIX

SOURCE CODE

Read source

```
package com.test.virtual.sensor.read.file;
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.InputStreamReader;
import java.sql.ResultSet;
import com.test.rsa.encryption.decryption.RSAEncryptorUtil;
import com.test.virtual.sensor.db.util.DatabaseInitializer;
/**
 *
 * @author vigneshKumar A
 * @since 1.0
 */
public class ReadFromFile extends DatabaseInitializer
{
private static ReadFromFile object = new ReadFromFile();
private RSAEncryptorUtil util = new RSAEncryptorUtil();
public static void main(String[] args) throws Exception {
object.readFromFile();
}
private void readFromFile() throws Exception {
try {
FileInputStream fileInputStream = new FileInputStream("D:/forestfires.txt");
BufferedReader reader = new
```



```
BufferedReader(newStreamReader(fileInputStream));
  String strLine;
  String[] token;
  boolean isHeadersRead = false;
  int rowIndex = 1;
  ResultSet rs = null;
  while ((strLine = reader.readLine()) != null) {
  if (isHeadersRead) {
  try {
  token = strLine.split(",");
  String x = util.encrypt(String.valueOf(Float.parseFloat(token[0])));
  String y = util.encrypt(String.valueOf(Float.parseFloat(token[1])));
  String month = util.encrypt(token[2]);
  String day = util.encrypt(token[3]);
  String FFMC = util.encrypt(token[4]);
  String DMC = util.encrypt(token[5]);
  String DC = util.encrypt(token[6]);
  String ISI = util.encrypt(token[7]);
  String temp = util.encrypt(token[8]);
  String RH = util.encrypt(token[9]);
  String wind = util.encrypt(token[10]);
  String rain = util.encrypt(token[11]);
  String area = util.encrypt(token[12]);
  String query = "select * from physical_sensor where X = '" + x + "' and Y='" + y
+ "'";
  rs = getStmt().executeQuery(query);
  while (rs.next()) {
  throw new Exception();
  }
  System.out.println("Row updated -- X: " + x + ", Y: " + y);
```

```
query = "insert into physical_sensor values('" + x + "','" + y + "','" + month + "','"
+ day + "','"    + FFMC + "','" + DMC + "','" + DC + "','" + ISI + "','" + temp + "','" +
RH + "','" + wind + "','" + rain + "','" + area + "')";
getStmt().execute(query);
} catch (Exception e) {
System.out.println("Row skipped : " + rowIndex);
}
} else {
isHeadersRead = true;
}
rowIndex++;
}
} catch (Exception e) {
throw e;
}
}
}
```

Virtual sensor

```
package com.test.virtual.sensor;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```



```
import com.test.rsa.encryption.decryption.RSAEncryptorUtil;
import com.test.virtual.sensor.db.util.DatabaseInitializer;
/**
 *
 * @author VigneshKumar A
 * @since 1.0
 */
public class CreateVirtualSensors extends DatabaseInitializer
{
private static CreateVirtualSensors _instance;
private RSAEncryptorUtil util = new RSAEncryptorUtil();
private int NUM_CLUSTERS;
private int TOTAL_DATA;
private List<Float> xValues = new ArrayList<Float>();
private List<Float> yValues = new ArrayList<Float>();
private static ArrayList<Data> dataSet = new ArrayList<Data>();
private static ArrayList<Centroid> centroids = new ArrayList<Centroid>();
protected String selectQuery;
protected String insertQuery;
protected String updateQuery;
protected ResultSet rs;
private static CreateVirtualSensors getInstance() {
if (_instance == null) {
_instance = new CreateVirtualSensors();
}
```



```
return _instance;
}
public static void main(String[] args) throws Exception {
try {
getInstance().readInput();
getInstance().readValuesFromDB();
getInstance().initializeCenteroids();
getInstance().initializeKMeanCluster();
getInstance().createVirtualSensors();
System.out.println("Virtual sensors created successfully");
} catch (Exception e) {
e.printStackTrace();
}
}
private void readInput() throws IOException {
try {
System.out.println("Enter the number of clusters needed: ");
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
String noOfClusters = reader.readLine();
NUM_CLUSTERS = Integer.parseInt(noOfClusters);
} catch (Exception e) {
System.out.println("Invalid input");
readInput();
}
```


SCREEN SHOTS

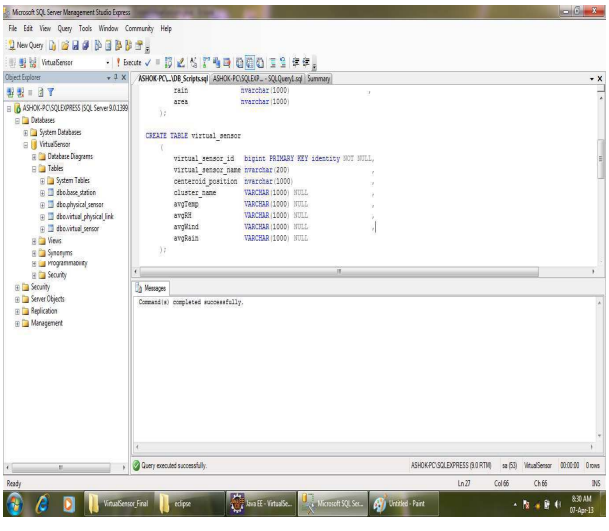

Fig .A1.Initial table creation

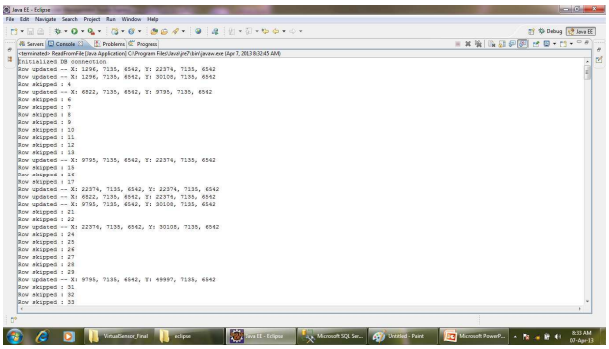

Fig .A2. Reading input from text file with replication elimination

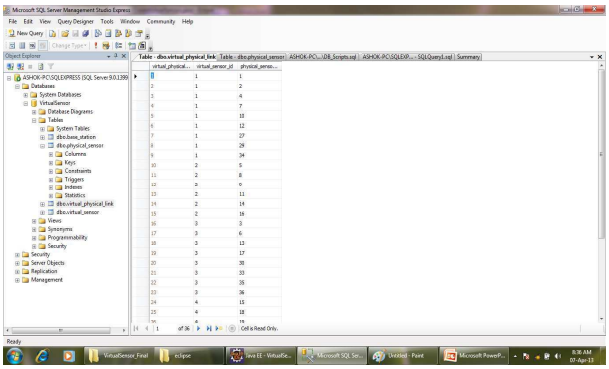

Fig . A4. Cluster formation

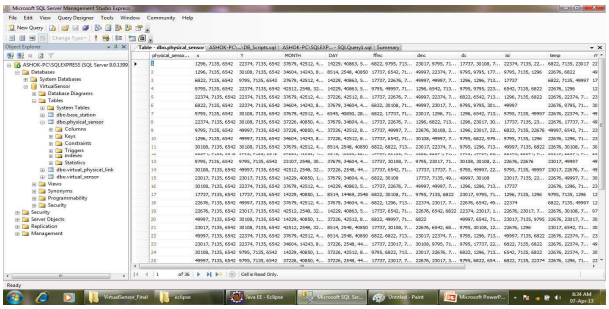

Fig .A3.Encryption

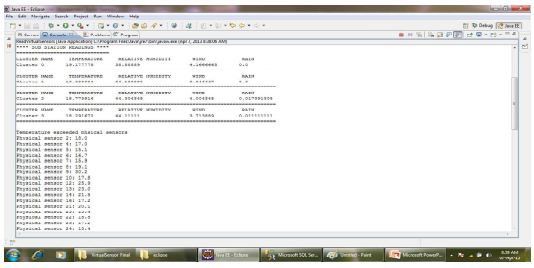

Fig . A5. Base station readings

REFERENCES

1. Bandara, Anura P. Jayasummana "Cluster Tree Based Self Organization of Virtual Sensor Networks*"*, IEEE Network 20(3) (2008) 20–25.

2. Bo Sun, Yang Xiao, Chung Chih Li, Hsiao-Hwa Chen d, T. Andrew Yang, "Security co-existence of wireless sensor networks and RFID for pervasive computing", Computer Communications 31 (2008) 4294–4303,2008

3. Chan. H, A. Perrig, D. Song, "Secure hierarchical in-network aggregation in sensor networks", in: Proc. of ACM CCS '06, Alexandria, VA, Nov. 2006, pp.278–287

4. Chen. B, Y. Lin, P.K. Varshney, "Decision fusion rules based on BPN in multi-hop wireless sensor networks", IEEE Transactions on Aerospace and Electronic Systems 41 (2) (2005) 475–488,2005

5. Hellerstein. J, S. Madden, M. Franklin, and W. Hong. TinyDB: "An Acquisitional query processing system for sensor networks", ACMTrans. on Database Systems", 30(1):122–173, 2005

6. Hung-Ta Pai, Jing Deng, Yunghsiang S. Han, "Time-slotted voting mechanism for fusion data assurance in wireless sensor networks under stealthy attacks", Computer Communications 33 (2010) 1524–1530,2010

7. Kabadayi S., Pridgen A., Julien C., "Virtual Sensors: Abstracting Data from Physical Sensors". Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks; Buffalo-Niagara Falls, NYUSA 26–29 June 2006.

8. Krunz. M, O.Younis, S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges", IEEE Network 20(3) (2006) 20–25.

9. Nathan Ota, William T.C. Kramer, "TinyML:Meta-data for wireless sensor networks". http://dnclab. keley.edu/~ta/research/TinyML/TinyML2.htm, 2005

10. Sanem Kabaday, Christine Julien, William O'Brien, and Drew Stovall, "Virtual Sensors: A Demonstration", TR-UTEDGE-2007-003,2007

11. TIAN Bin, YANG Yi-xian, LI Dong, LI Qi, XIN Yang, "A security framework for wireless sensor networks", December 2010, 17(Suppl. 2): 118–122,2010

12. Wen-Tsai Sung, "Multi-sensors data fusion system for wireless sensors networks of factory monitoring via BPN technology", Expert Systems with Applications 37 (2010) 2124–2131,2010

13. Yao. Y "The cougar approach to in-network query processing in sensor networks", ACM SIGMOD Record, 31(3):9–18, 2002.

LIST OF PUBLICATIONS

1. Vigneshkumar.A, Siddique Ibrahim S.P, "Enhance sensor data based on time slot voting mechanism improved analysis", Fourth national conference, Sri Krishna college of Engineering and Technology, 8th February 2013.
2. Vigneshkumar. A , Siddique Ibrahim S.P, "Privacy assured data fusion in virtual sensor network", Fifth international conference on advances in computer and communication, Sun college of Engineering and Technology, 18th February 2013.