Driver Drowsiness And Aggressiveness Detection System

Ms.Susan M George
Asst.Prof of CSE
MBCCET
Kerala, India
susanmgeorge@mbcpeermade.com

Ms.Alphonsa George
Dept of CSE
MBCCET
Kerala, India
alphonsalicegeorge@gmail.com

Ms.Lakshmi Krishna
Dept of CSE
MBCCET
Kerala, India
lakshmiknair7@gmail.com

Ms.Neethu Merin K Suresh
Dept of CSE
MBCCET
Kerala, India
neethooz14@gmail.com

Ms.Siji Johnson
Dept of CSE
MBCCET
Kerala,India
sijijohnson2all@gmail.com

***Abstract*—** This project is entitles as "Implementing Internet of things for advanced accident detection" is developed using IOT hardware kit as the transmitter and cloud sever as receiver. Hyper terminal tool has been used for PC interface. The main objective of this project is to develop a web based application to communicate over internet server in secured manner for advanced accident detection.

***Introduction* --** The Internet of Things is an emerging topic of technical, social, and economic significance. IOT involves in various departments like Medical industries, Automobile industries, Manufacturing industries, and etc. Now a day's utility components everyday objects are being combined with Internet connectivity and powerful data analytic capabilities that promise to transform the way we work, live, and play. The term Internet of Things generally refers to scenarios where network connectivity and computing capability extends to objects, sensors and everyday items not normally considered computers, allowing these devices to generate exchange and consume data with minimal human intervention.

In this project IOT environment has been created for advanced accident detection using Dynamic Service Non Dependency Verification. This environment has been implemented using arduino UNO controller board

I. OVERVIEW

The first process of this project initiate with hardware interface. The hardware has been designed using 8051 microprocessor. The micro processor contains 40 pin. More interactions can be done using 8051 pin interaction. In added with various sensors can be connected through the controller board interface. Multi sensors have been interacted in this hardware. The major hardware interactions are follows.

- Alcohol Sensor – Detects Alcohol
- Heartbeat sensor – To identify the driver's sleep and early heart attack detection
- Smoke Sensor – Detects if driver smokes
- Arduino UNO controller board – Main interface board
- Level Convertor – For sensor interface with main board
- Power supply unit – Converts 220 V into 12 V
- COMM to USB convertor – For System interface

All the above mentioned hardware process has been implemented in a single interface controller board. The Interface part for software is follows

- IOT based Cloud Server
- Cloud Application interface.
- Cloud storage
- Mobile application for user or admin interface

All the sensors will be wired inside the vehicle. The sensors value will be uploaded in a centralized cloud server. A threshold value will be assigned for each sensor. In case of any abnormal means, the warning will be sent to the user interface end. While heart attack has been detected means, automatically the horn will turn on and the vehicle will slow down towards left corner. Secure Service Virtualization in IoT by Dynamic Service Non Dependency Verification is ensured by the hardware phase initially.

In this project, we propose a heterogeneous IOT scheme to secure communication between a sensor node and an Internet host. We prove that this scheme is indistinguishable against various conditions. This project has the following advantages. First, it achieves confidentiality, integrity, authentication, and non-repudiation in a logical single step. Second, it allows a sensor node in an identity-based Enhanced data transfer to send a message to an Internet host in a public key infrastructure method.

A data log been maintained in the cloud server. The log contains the entire sensor information during the travel. From the data log the trustworthy of the driver can be calculated using Navie Baysian Algorithm. This algorithm work efficiently over the data set and predicts the trust of the driver. This leads to select a good driver for risky transport.

This project works under two phases

1) Offline phase

The hardware shows the corresponding sensor values in the hardware itself, without any internet connection. Using LCD display the values generated in the sensors will be displayed.

2) Online Phase

The phase need internet. Using COMM port from the hardware, the sensor value will be uploaded to a cloud server over internet. Multi user rights will be provided to view the sensor data from multiple locations and in multiple devices.

II. MODULES AND DESCRIPTION

A. Hardware Interface

This is the initial Module of this project. The entire hardware and sensor interface will be available in this module. Here PIC or Arudino Microcontroller has been used for efficient sensor interface. This project supports multiple sensor interface model. All sensors will be embedded with LCD for offline communication. Microcontroller has been connected with power supply unit.

B. COMM port Communication

COMM port is meant as communication port. This module interface hardware unit with PC. The communication has been done through RS232 tool. RS232 is a system interface tool which efficiently works with converting machine language into system language. Using Hyper terminal the data from the sensor unit will be sent to PC. Data will be split sensor wise while transferring into PC.

C. Data Upload

According to this project, in order to implement IOT, here sensor data were used. A set of data from corresponding sensor information will be used here as a data set. Uploading is the transmission of a file from one computer system to another, usually larger computer system. From a network user's point-of-view, to upload a file is to send it to another computer that is set up to receive it. Uploaded data will be stored in the web server. Here we are uploading the data set in a cloud server.

D. Configuring a Centralised Cloud Server

The public cloud environment is the IaaS/PaaS Infrastructure or Platform as a Service that we rent from Linux (IaaS) or Microsoft (PaaS). Both are enabled for web hosting. Then, your SaaS stack will run under your Internet environment most likely in a virtualized one on your own equipment which would make it private. In this project we specialize in private cloud technology. Here we execute in a cloud environment. If strict security requirements go public or hybrid and if not, try the public or community cloud environment. So that here we are implementing a web services for the output purpose as well as the environment will be shown in actual while hosting the application. So finally SaaS can be fully utilized in cloud environment as IaaS/PaaS. Thus we formed cloud environment.

E. Data Verification

This is a user define function models where admin and user can interact. In technical it is said to be the application layer, where all the data were verified and viewed by the users. According to the architecture design, user and admin will be available in the application layer, business layer will be decision making layer and all data will available in the data layer.

III. SYSTEM STUDY

1. EXISTING SYSTEM

Still many researches are under going for implementing in Internet of Things (IOT). In major companies the data transmission done manually. Else the process will be done through mail or some other communications. No company has successfully implemented internet of things successfully.

1.1. DRAWBACKS OF EXISTING SYSTEM

- Still many companies are taking manual reading to know the current status of the machine.
- While taking reading range can be checked manually
- It is a tedious process to take date wise reading from the machines
- Parameter wise reading is not possible.
- In case the machine got over loaded, there will not be any warning will be generated from the machine side.
- Maintenance is not much easy.
- Need to invest more for the maintenance

2. PROPOSED SYSTEM

The concept of the internet of things, or IOT, is spreading its wings wider and stronger in the current it scenario, and is gradually taking part in every facet of our lives. Look at the way the automobile industry wants to be connected with each and every thing associated with it. There is a high level of adoption of automobile devices that are connected to each other. In fact, the adoption level shows an increasing trend and there will be more takers for these devices in the future. The tech experts opine that like the internet, the internet of things too is going to be a part of our everyday life. With an increasing number of medical devices getting connected to the internet, the idea of interconnected automobile sphere gets more fascinating. It is also evident that several software, service, and product companies are showing interest in connecting devices with a view to make their primary product or service more attainable.

The internet of things (IOT) provides the opportunity to enable and extend digital business scenarios, helping you better connect people, processes, devices and other m2m assets, and better harness data across your business and operations. Improving efficiencies, enabling innovation and fuelling transformation are the cornerstones of Microsoft's vision for the digital business. With Microsoft azure IOT services, you can monitor assets to improve efficiencies, drive operational performance to enable innovation, and leverage advanced data analytics to transform your company with new business models and revenue streams.

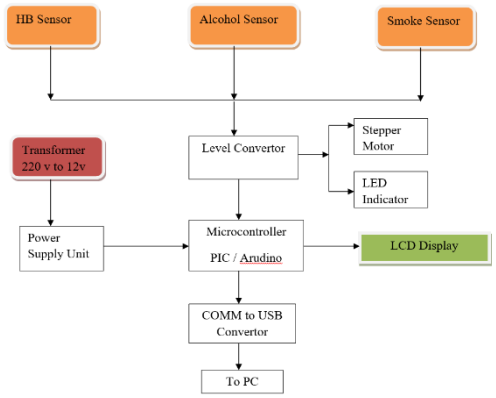
An IOT device can be tested implemented remotely. For example, a technician can connect from their own office and run diagnostics of a automobile sensor. The technician can pinpoint the problem's root cause and leverage a knowledge management application to find answers to common problems.

2.1. ADVANTAGES OF PROPOSED SYSTEM

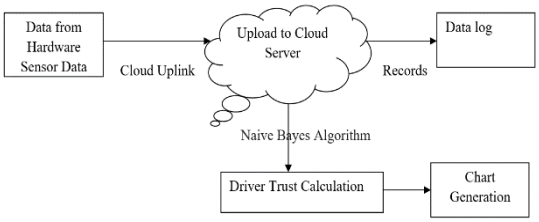
- Can monitor the vehicle in remote
- Using Enhanced cryptography, the encryption will process will done before reaching the cloud server.
- Data can be accessed in all types of mobile applications
- No need to configure the cloud environment always, once the configuration done means the connection will get established automatically.
- Using COMM port, the data validation has been done from the hardware side. So no need to change the code for data validation.
- In future offline can be processed can be done using temporary memory.

IV. SYSTEM ARCHITECTURE

Hardware Unit



Software Unit



V. ALGORITHM USED

NAIVE BAYES – ALGORITHM

Step 1: Begin the process.

Step 2: Read the training data from sensor

Step 3: Read the testing data from a previous data

Step 4: Set K,L,M as threshold value to some value.

Step 5: Normalize the attribute values in the range 0 to 1. Get Threshold input

Step 6: Value = Value / (1+Value);

Step 7 : OLEDB.4.0;Data Source=" + Sensor data + "; Extended Properties= sensor data 8.0;"

Step 8 : Find the K nearest neighbours in the training data set based on the Euclidean distance

Step 9: dist((x, y,z), (K, L.M)) =

$$\sqrt{(x - K)^2 + (y - L)^2 + (z - M)^2}$$

Step 10 : Predict the class value by finding the maximum class represented in the for K,L,M

Step 11: dist((x, y,z), (K,L,M)) =

$$|x - K| + |y - L| + |z - M|$$

Step 12 : Comparing the dataset value and range value for calculating occurrence percentage of trust calculation

Step 13: Convert.ToDouble(red1[1].ToString()) for displaying the value of calculated trust

Step 14 : The above process continued for all test and finally the percentage of occurrence is calculated.

VI. IMPLEMENTATION

As we note in the principles that guide our work, ensuring the security, reliability, resilience, and stability of Internet applications and services is critical to promoting trust and use of the Internet. As users of the Internet, we need to have a high degree of trust that the Internet, its applications, and the devices linked to it are secure enough to do the kinds of activities we want to do online in relation to the risk tolerance associated with those activities. The Internet of Things is no different in this respect, and security in IoT is fundamentally linked to the ability of users to trust their environment.

If people don't believe their connected devices and their information are reasonably secure from misuse or harm, the resulting erosion of trust causes a reluctance to use the Internet. This has global consequences to electronic commerce, technical innovation, free speech, and practically every other aspect of online activities. Indeed, ensuring security in IoT products and services should be considered a top priority for the sector. As we increasingly connect devices to the Internet, new opportunities to exploit potential security vulnerabilities grow. Poorly secured IoT devices could serve as entry points for cyberattack by allowing malicious individuals to re-program a device or cause it to malfunction. Poorly designed devices can expose user data to theft by leaving data streams inadequately protected. Failing or malfunctioning devices also can create security vulnerabilities. These problems are just as large or larger for the small, cheap, and ubiquitous smart devices in the Internet of Things as they are for the computers that have traditionally been the endpoints of Internet connectivity. Competitive cost and technical constraints on IoT devices challenge manufacturers to adequately design security features into these devices, potentially creating security and long-term maintainability vulnerabilities greater than their traditional computer counterparts.

Along with potential security design deficiencies, the sheer increase in the number and nature of IoT devices could increase the opportunities of attack. When coupled with the highly interconnected nature of IoT devices, every poorly secured device that is connected online potentially affects the security and resilience of the Internet globally, not just locally. For example, an unprotected refrigerator or television in the US that is infected with malware might send thousands of harmful spam emails to recipients worldwide using the owner's home Wi-Fi Internet connection.

To complicate matters, our ability to function in our daily activities without using devices or systems that are Internet-enabled is likely to decrease in a hyper connected world. In fact, it is increasingly difficult to purchase some devices that are not Internet-connected because certain vendors only make connected products. Day by day, we become more connected and dependent on IoT devices for essential services, and we need the devices to be secure, while recognizing that no device can be absolutely secure. This increasing level of dependence on IoT devices and the Internet services they interact with also increases the pathways for wrongdoers to gain access to devices. Perhaps we could unplug our Internet-connected TVs if they get compromised in a cyber attack, but we can't so easily turn off a smart utility power meter or a traffic control system or a person's implanted pacemaker if they fall victim to malicious behavior. This is why security of IoT devices and services is a major discussion point and should be considered a critical issue. We increasingly depend on these devices for essential services, and their behavior may have global reach and impact.

Many IoT deployments will consist of collections of identical or near identical devices. This homogeneity magnifies the potential impact of any single security vulnerability by the sheer number of devices that all have the same characteristics. For example, a communication protocol vulnerability of one company's brand of Internet-enabled light bulbs might extend to every make and model of device that uses that same

protocol or which shares key design or manufacturing characteristics.

Many Internet of Things devices will be deployed with an anticipated service life many years longer than is typically associated with high-tech equipment. Further, these devices might be deployed in circumstances that make it difficult or impossible to reconfigure or upgrade them; or these devices might outlive the company that created them, leaving orphaned devices with no means of long-term support. These scenarios illustrate that security mechanisms that are adequate at deployment might not be adequate for the full lifespan of the device as security threats evolve. As such, this may create vulnerabilities that could persist for a long time. This is in contrast to the paradigm of traditional computer systems that are normally upgraded with operating system software updates throughout the life of the computer to address security threats. The long-term support and management of IoT devices is a significant security challenge.

Many IoT devices are intentionally designed without any ability to be upgraded, or the upgrade process is cumbersome or impractical. For example, consider the 2015 Fiat Chrysler recall of 1.4 million vehicles to fix a vulnerability that allowed an attacker to wirelessly hack into the vehicle. These cars must be taken to a Fiat Chrysler dealer for a manual upgrade, or the owner must perform the upgrade themselves with a USB key. The reality is that a high percentage of these autos probably will not be upgraded because the upgrade process presents an inconvenience for owners, leaving them perpetually vulnerable to cybersecurity threats, especially when the automobile appears to be performing well otherwise.

Many IoT devices operate in a manner where the user has little or no real visibility into the internal workings of the device or the precise data streams they produce. This creates a security vulnerability when a user believes an IoT device is performing certain functions, when in reality it might be performing unwanted functions or collecting more data than the user intends. The device's functions also could change without notice when the manufacturer provides an update, leaving the user vulnerable to whatever changes the manufacturer makes.

Some IoT devices are likely to be deployed in places where physical security is difficult or impossible to achieve. Attackers may have direct physical access to IoT devices. Anti-tamper features and other design innovations will need to be considered to ensure security.

1. HARDWARE IMPLEMENTATION (Protocol)

The protocols build on recent algorithmic research (Ad-hoc On-demand Distance Vector, neuRFon) to automatically construct a low-speed ad-hoc network of nodes. In most large network instances, the network will be a cluster of clusters. It can also form a mesh or a single cluster. The current profiles derived from the ZigBee protocols support beacon and non-beacon enabled networks.

In non-beacon-enabled networks (those whose beacon order is 15), an unslotted CSMA/CA channel access mechanism is used. In this type of network, ZigBee Routers typically have their receivers continuously active, requiring a more robust power supply. However, this allows for heterogeneous networks in which some devices receive continuously, while others only transmit when an external stimulus is detected. The typical example of a heterogeneous network is a wireless light switch: The ZigBee node at the lamp may receive constantly, since it is connected to the mains supply, while a battery-powered light switch would remain asleep until the switch is thrown. The switch then wakes up, sends a command to the lamp, receives an acknowledgment, and returns to sleep. In such a network the lamp node will be at least a ZigBee Router, if not the ZigBee Coordinator; the switch node is typically a ZigBee End Device.

In beacon-enabled networks, the special network nodes called ZigBee Routers transmit periodic beacons to confirm their presence to other network nodes. Nodes may sleep between beacons, thus lowering their duty cycle and extending their battery life. Beacon intervals may range from 15.36 milliseconds to 15.36 ms * 2^{14} = 251.65824 seconds at 250 kbit/s, from 24 milliseconds to 24 ms * 2^{14} = 393.216 seconds at 40 kbit/s and from 48 milliseconds to 48 ms * 2^{14} = 786.432 seconds at 20 kbit/s. However, low duty cycle operation with long beacon intervals requires precise timing, which can conflict with the need for low product cost.

In general, the ZigBee protocols minimize the time the radio is on so as to reduce power use. In beaconing networks, nodes only need to be active while a beacon is being transmitted. In non-beacon-enabled networks, power consumption is decidedly asymmetrical: some devices are always active, while others spend most of their time sleeping.

Except for the Smart Energy Profile 2.0, which will be MAC/PHY agnostic, ZigBee devices are required to conform to the IEEE 802.15.4-2003 Low-Rate Wireless Personal Area Network (WPAN) standard. The standard specifies the lower protocol layers—the physical layer (PHY), and the media access control (MAC) portion of the data link layer (DLL). This standard specifies operation in the unlicensed 2.4 GHz (worldwide), 915 MHz (Americas) and 868 MHz (Europe) ISM bands. In the 2.4 GHz band there are 16 ZigBee channels, with each channel requiring 5 MHz of bandwidth. The center frequency for each channel can be calculated as, $F_C = (2405 + 5 * (ch - 11))$ MHz, where ch = 11, 12, ..., 26.

The radios use direct-sequence spread spectrum coding, which is managed by the digital stream into the modulator. BPSK is used in the 868 and 915 MHz bands, and OQPSK that transmits two bits per symbol is used in the 2.4 GHz band. The raw, over-the-air data rate is 250 kbit/s per channel in the 2.4 GHz band, 40 kbit/s per channel in the 915 MHz band, and 20 kbit/s in the 868 MHz band. Transmission range is between 10 and 75 meters (33 and 246 feet) and up to 1500 meters for zigbee pro, although it is heavily dependent on the particular environment. The output power of the radios is generally 0 dBm (1 mW).

The basic channel access mode is "carrier sense, multiple access/collision avoidance" (CSMA/CA). That is, the nodes talk in the same way that people converse; they briefly check to see that no one is talking before they start. There are three notable exceptions to the use of CSMA. Beacons are sent on a fixed timing schedule, and do not use CSMA. Message acknowledgments also do not use CSMA. Finally, devices in Beacon Oriented networks that have low latency real-time requirements may also use Guaranteed Time Slots (GTS), which by definition do not use CSMA.

1.1. ZIGBEE RF4CE

The RF4CE (Radio Frequency for Consumer Electronics) Consortium agreed to work with the ZigBee Alliance to jointly deliver a standardized specification for radio frequency-based remote controls. ZigBee RF4CE is designed to be deployed in a wide range of remotely-controlled audio/visual consumer electronics products, such as TVs and

set-top boxes. It promises many advantages over existing remote control solutions, including richer communication and increased reliability, enhanced features and flexibility, interoperability, and no line-of-sight barrier.

1.2. SOFTWARE AND HARDWARE

The software is designed to be easy to develop on small, inexpensive microprocessors. The radio design used by ZigBee has been carefully optimized for low cost in large scale production. It has few analog stages and uses digital circuits wherever possible.

Even though the radios themselves are inexpensive, the ZigBee Qualification Process involves a full validation of the requirements of the physical layer. This amount of concern about the Physical Layer has multiple benefits, since all radios derived from that semiconductor mask set would enjoy the same RF characteristics. On the other hand, an uncertified physical layer that malfunctions could cripple the battery lifespan of other devices on a ZigBee network. Where other protocols can mask poor sensitivity or other esoteric problems in a fade compensation response, ZigBee radios have very tight engineering constraints: they are both power and bandwidth constrained. Thus, radios are tested to the ISO 17025 standard with guidance given by Clause 6 of the 802.15.4-2006 Standard. Most vendors plan to integrate the radio and microcontroller onto a single chip getting smaller devices.

These octal buffers and line drivers are designed to have the performance of the popular 'HC240 series devices and to offer a pin out with inputs and outputs on opposite sides of the package. This arrangement greatly facilitates printed circuit board layout. The 3-state control gate is a 2-input NOR. If either output-enable (OE1 or OE2) input is high, all eight outputs are in the high-impedance state. The'HCT541 devices provide true data at the outputs.

VII. SPECIFICATIONS

Operating Voltage Range of 4.5 V to 5.5 V
High-Current 3-State Outputs Interface
Directly With System Bus or Can Drive Up
To 15 LSTTL Loads
Low Power Consumption, 80-µA Max ICC
Typical tpd = 12 ns
±6-mA Output Drive at 5 V
Low Input Current of 1 µA Max
Inputs Are TTL-Voltage Compatible
Data Flow-Through Pin out (All Inputs on
Opposite Side from Outputs)

VIII. CONCLUSION

IOT methodologies with automobile are undoubtedly the core technologies of future IoT. Many researchers have studied various research issues on integrating IOT automobile and sensor technologies. At present, efforts are being made to integrate these two technologies on the same IoT platform in different fields. Unlike conventional studies that provide IoT platforms at the architecture level only, this study proposed an implementation model of an sensor data repository on the basis of MongoDB. Furthermore, based on logistic process simulation of automotive parts, the proposed RFID/sensor data repository was empirically validated in terms of even distribution of data and query speed.

This phase is implemented up to IOT hardware unit, which works more perfect than expected.

REFERENCES

[1] E. Ilie-Zudor, Z. Kemeny, F. Blommestein, L. Monostori, and A. Meulen, "A survey of applications and requirements of unique identification systems and RFID technique," Comput Ind, vol. 62, pp. 227-252, 2011.

[2] L. D. Xu, W. He, and S. Li, "Internet of things in industries: a survey," IEEE T Ind Inform, vol. 10, no. 4, pp. 2233-2243, 2014.

[3] J. Mitsugi, T. Inaba, B. Pátkai, L. Theodorou, J. Sung, T. S. López, D. Kim, D. cFarlane, H. Hada, Y. Kawakita, K. Osaka, and O. Nakamura, Architecture Development for Sensor Integration in the EPCglobal Network, Auto-ID Labs White Paper, WP-SWNET-018, 2007.

[4] Y.-S. Kang, H. Jin, O. Ryou, and Y.-H. Lee, "A simulation approach for optimal design of RFID sensor tag-based cold chain systems," J. Food Eng, vol. 113, pp. 1-10, 2012.

[5] L. Jiang, L. D. Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT-oriented data storage framework in cloud computing platform," IEEE T Ind Inform, vol. 10, no. 2, pp. 1443-1451, May 2014.

[6] D. E. O'Leary, "'Big Data', The 'Internet of things' and the 'internet of signs'," Intell. Syst. Account. Finance Manag., vol. 20, pp. 53-65, 2013.

[7] J. S. Veen, B. Waaij, and R. J. Meijer, "Sensor data storage performance: SQL or NoSQL, Physical or Virtual," in Proc. 5th IEEE Cloud, pp. 431-438, 2012.

[8] A. Castiglione, M. Gribaudo, M. Lacono, and F. Palmieri, "Exploiting mean field analysis to model performances of big data architectures," Future Gener Comput Syst., vol. 37, pp. 203-211, 2014.

[9] G. Noorts, J. Engel, J. Taylor, D. Roberson, R. Bacchus, T. Taher, and K. Zdunek, "An RF spectrum observatory database based on a hybrid storage system," in Proc. IEEE Dyspan, pp. 114-120, October 2012.

[10] EPCglobal Inc., EPC Information Services (EPCIS) version 1.0, EPCglobal ratified specification, Available: http://www.gs1.org/gsmp/kc/epcglobal/epcis.

[11] C. Strauch, U. L. S. Sites, and W. Kriha, "NoSQL databases," Lecture Notes, Stuttgart Media University, 2011.

[12] R. Cattell, "Scalable SQL and NoSQL data stores," Sigmod Rec., vol. 39, no. 4, pp. 12-27, 2011.