Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 235–248

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Energy efficient machine learning based SMART-A-BLE implemented Wireless Battery Management System for both Hybrid Electric Vehicles and Battery Electric Vehicles

Srikalaivani Pannerselvam[a], VAnantha Narayanan[b], Gireesh Kumar T[b]

[a]*Department of Electronics and Communication Engineering,*
Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, 641112, India
[b]*Department of Computer Science and Engineering,*
Amrita School of Computing, Coimbatore, Amrita Vishwa Vidyapeetham, 641112, India

Abstract

Electric vehicles have a larger battery pack with more than 100's cells connected in series or parallel combinations to form many battery modules. A Battery Management system (BMS) continuously monitors Current, Voltage, and Temperature data to trigger control algorithms like cell balancing and thermal management using wired communication which is then sent to the cloud for real-time analytics [1]. Wired communication restricts the flexibility in battery module assembly and makes it difficult for battery swapping. The proposed wireless communication, along with the SMART-A-BLE algorithm for BMS in the electric vehicle, uses Bluetooth Low Energy [2], [3], [4], and [5]. SMART-A-BLE algorithm predicts the best connection interval using the Decision tree model based on the link quality factor captured based on real time latency. The model uses threshold latency, expected retransmission factor, data packet length, and the number of peripherals as input features. A look-up table-based algorithm is also proposed for resource constraint devices to achieve the optimal operating point where the slaves could transmit at both low latency and low current consumption with less congestion. SMART-A-BLE control algorithm outperforms the mathematical model-based algorithm in both ideal and interference conditions within a metallic container by maintaining robust connectivity.

Peer-review under responsibility of the scientific committee of the International Conference on Machine Learning and Data Engineering

Keywords: WBMS; Bluetooth Low Energy; Machine Learning; Electric Vehicle; Model predictive control; Battery swapping

1. Introduction

Lithium-ion batteries in electric vehicles contribute to a significant part of the cost of EVs. It is crucial to ensure the prolonged life of those batteries, which directly affects the range of Electric Vehicles. Hence BMS continuously

1877-0509
Peer-review under responsibility of the scientific committee of the International Conference on Machine Learning and Data Engineering
10.1016/j.procs.2023.01.006

monitors and controls the charging and discharging cycles to reduce the battery degradation. BMS uses wired communication for sending the temperature and voltage details from the monitor device to the master BMS. The wiring gets more complex and heavy in wired communication as the number of battery modules increases. Possible wire damage will increase the repair cost as the entire battery pack should be changed. The wired solution needs isolation from the electrically noisy environment. It also reduces flexibility on battery packaging, which is challenging for battery swapping solutions. However, Wireless BMS provides more flexibility on battery packaging, no wiring complexity, and reduces around 15% of the battery package volume, i.e., around 3 miles of the wiring leading to low carbon footprint, low service, and maintenance cost. Wireless battery modules also provide the EV manufacturer with flexibility in the location and arrangement of battery modules [3] and is naturally isolated. It makes it easy for EVs to adapt to battery swapping solutions. It is also critical that wireless communication implemented should be robust and resilient enough to withstand the harsh automotive environment.

Nomenclature

T_{Lat}	Latency measured using HCI event
$T_{Lat_{Th}}$	Threshold latency
$T_{Counter}$	RTC based counter
$CE_{\max}$	Maximum connection event
CI	Connection Interval
R_{Factor}	Retransmission factor
LQF	Link quality factor
ATT_MTU	Attribute Maximum transmission unit
N_{peri}	Number of peripherals
dpl	Data Packet Length
WBMS	Wireless Battery Management System
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EV	Electric Vehicle
BLE	Bluetooth Low Energy
RTT	Round Trip Time
HCI	Host Controller Interface
LOS	Line of Sight

The proposed work uses Bluetooth Low Energy communication replacing wired communication for Battery Management systems in a star topology. Integration of BLE devices will also make the network connect easily with other android devices for data sharing [6]. Also, it is essential to make the BLE network intelligent enough to handle harsh environments by understanding the level of interference in terms of number of retransmissions at any point in time. If number of retransmissions of a data packet increases, then the latency in communication increases which is not desirable in safety critical applications[7]. If congestion happens in the wireless network, then data packets will not be delivered to the master module within supervision timeout and lead to disconnection of the BLE module from the master. Michael Spork[8] has implemented a mathematical model to capture interference in terms of the number of connection events required to successfully transmit a packet between a master and a single slave. Park[9] also used the same mathematical model to adapt the connection interval based on the number of retransmissions of a data packet but also used data rate to reduce the power consumption with a bit of sacrifice on the quality of service. The existing algorithm uses this mathematical model to dynamically adapt to a new connection interval which is one of the connection parameters based on the observed number of retransmissions such that the network is operated within the threshold latency. Even though the existing algorithms could operate the network within threshold latency and lower power consumption mode, they could not reach the optimal operating point of the network. By reaching the optimal operating point where the predicted connection interval for the given input features will achieve the lowest power consumption mode without sacrificing the link quality. For the first time, we introduce the SMART-A-BLE algorithm to efficiently operate the multi peripheral BLE network in both low current and low latency mode by tuning the connection parameters of the BLE link layer dynamically sensing the link quality. Radio on-off frequency is highly reduced by estimating optimal higher connection intervals to operate slaves in Low current consumption mode. The algorithm is verified with the BLE network, including different BLE vendor modules.

The remainder of this paper is ordered as follows: Section 2 explains the previous work related to the study conducted in the presence of interference with master-slave architecture. Section 3 explains the proposed methodology and architecture of the WBMS experimental setup. Also, data generation, machine learning model creation and SMART-A-BLE algorithm implementation is explained. Section 4 discusses the results and analysis of the SMART-A-BLE algorithm compared with the existing algorithm, and inferences from the results are discussed with supporting data plots. Section 5 concludes and adds the future scope of the paper.

2. Related Works

To date, only a limited number of studies are available on multi peripheral BLE networks in the presence of interference. Also, many mathematical models are formulated for point-to-point Master-Slave connection and are not suitable for multi peripheral BLE networks. There are some previous works[4], [10] and [11] done on the impact of bit error rate on the BLE discovery process[12] in a simulation environment. In [4] analyzes the performance of the BLE network with PER as a performance metric and results tested only in simulation. Farej, Ziyad Khalaf, and Saeed, Aydin M.'s work [10] analyzes the BLE piconet in a simulation in ideal operating conditions and not in a real-time environment with interference. [13] Implements wireless battery management system using BLE modules explain only the advantages of having wireless communication instead of wired communication in WBMS. Lee [11] proposes an algorithm that continuously monitors the packet error rate and executes the optimization algorithm to decrease or increase the Connection Interval parameter to achieve minimum current consumption of the BLE network in ideal conditions. Also, the algorithm does not converge to the global minima and is computationally expensive as it involves iteration to find the optimal point. Spork[8] and Park[9] use a "number of connection events" to estimate the connection interval dynamically using mathematical model. The algorithm is likely to hit the local minima and will not be able to converge to the global minima. For the first time, this work generates a data set with control parameters and performance parameters as features for a multi-peripheral BLE network. A Machine Learning model and a data-driven approach to execute the SMART-A-BLE algorithm to directly find the optimal operating connection interval with a constraint on the retransmission factor allowed and threshold latency requirement as per the application requirement. It helps the network reach the low latency and low current consumption mode for all the slaves faster than the mathematical model used in algorithms in previous papers.

3. Proposed work

Since BLE targets to run on resource-constrained devices, data transmission should be controlled dynamically based on link performance metrics to ensure uninterrupted communication. $AdaptaBLE$[9] and Spork[8] algorithms adjust the connection interval, data rate, and transmission current of the slaves to decrease the energy consumption with latency constraints but do not arrive at the exact optimal operating point. In the proposed work, Spork's mathematical model was analyzed for a Master with multiple peripherals, and its limitations are observed that it could not reach the optimal operating connection interval. Instead, the existing algorithms could reach only the local minima, which can still be improved to reduce the power consumption of the slave devices. Also, there are not many datasets available on multi–peripheral BLE networks using which we can create an intelligent system-specific model to estimate connection interval. Fig 1 (a) represents the software architecture of the algorithm where the link quality is continuously monitored using HCI events and control commands are sent based on the observed quality. Host layer has the Generic Access protocol (GAP) and Generic Attribute protocol (GATT) of BLE stack. Fig 1 (b) represents the hardware architecture of WBMS with different BLE vendor modules.

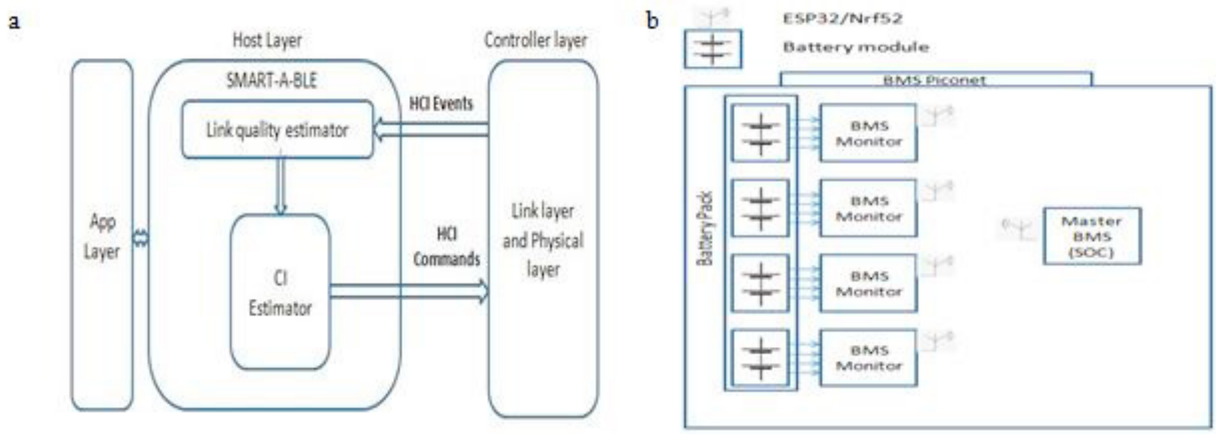

Fig. 1. (a) Software Architecture of SMART-A-BLE; (b) Hardware Architecture.

The SMART-A-BLE algorithm arrives at the optimal operating connection interval in a shorter period than the previous algorithms used. It helps to run the entire slave devices in low current consumption mode. BLE network is analyzed in ideal condition, with interference, without line of sight and the results are discussed in section 4. Since the wireless network is designed for a Battery Management System for the electric vehicle, it is likely to be operated in a harsh RF environment in a closed metallic environment. As per the literature survey, it is proved that the advertisement, connection parameters, data rate, and transmission power have a considerable impact on the current consumption of BLE modules[9]. Latency was measured using HCI-based timing information from a slave device using which link quality as shown in fig 2.

Most vendors have exposed APIs from the link layer to control the link layer and physical layer parameters like data packet length, data rate, and connection interval. In SMART-A-BLE, an automated data generator generates data by tuning these parameters to different values given as per the template and collecting the system performance parameters to statistically analyze the impact of these parameters on both latency and current consumption. After analyzing the data points, modeling is done using a machine-learning algorithm to predict the optimal connection interval for the given application's required threshold latency, data rate, number of peripherals connected, and data packet length. Using this model, it becomes easy to operate the BLE network of different sizes in its optimal operating point to ensure the BLE network operates at low latency and low current mode.

3.1 Link quality estimator

Link quality is estimated using latency experienced by the data read request from the master and read response collectively from all the slaves. i.e., a counter based on the master module's RTC is started once the read request is sent to the peripherals, from the master module. Since multiple peripherals are connected to a single master, the consolidated time taken to receive the data from all the peripheral modules is considered the latency in this case. This work uses the HCI event "BLE_GATTC_EVT_READ_RSP" to keep track of the timing between request and response to calculate the latency precisely. Using the standard HCI event mentioned above, the counter that runs based on the internal clock can be stopped once the master receives the HCI event "BLE_GATTC_EVT_READ_RSP" from all the peripherals. The proposed control algorithm can be applied to any vendor module supporting HCI standard events. As the algorithm runs only on the Master module, computation overhead is also avoided at the slaves. SMART-A-BLE will be able to dynamically change the connection parameter by observing the real-time performance parameters of the established network.

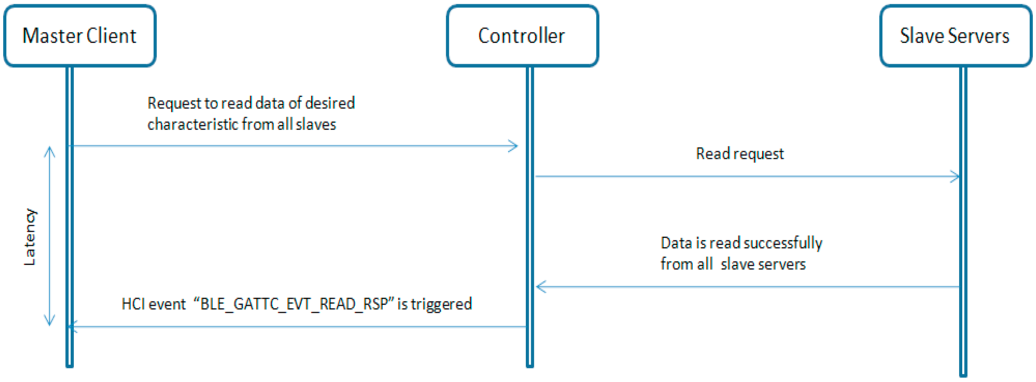

Fig. 2. Read request and response timing using HCI commands.

$$T_{Lat} = T_{Counter} \tag{1}$$

This latency measured includes the latency of data packets of all the slave modules connected to the master module. Leveraging Spork's approach for a multi peripheral scenario, a retransmission factor is calculated as shown in equation (2) which gives the SMART-A-BLE algorithm a hold on how to change the link quality dynamically based on the observed Link Quality Factor,

$$R_{Factor} = (T_{Lat} - CE_{\max}) / CI \tag{2}$$

Where R_{Factor} – Retransmission Factor, T_{Lat} - Latency (ms) measured from $T_{Counter}$, $CE_{\max}$ – maximum connection event(ms), CI - Connection Interval(ms).

Algorithm 1: Link Quality Factor estimation	
1	Define Window size N , $i = 1$ // Initializations
2	Append the latency measured to the list T_{Lat_list}
3	**While** $i\%N == 0$ // Read all the latencies till N instances
4	$T_{Lat\_list} \leftarrow [\ T_{Lat_1}, T_{Lat2}, ...T_{Lat_N}\ ]$ // List of N latencies stored in T_{Lat_list}
5	$T_{Lat\_Worst\_List} \leftarrow f(T_{Lat_n} > T_{Lat_{Th}})$ for each T_{Lat_n} where $1 \le n \le N$ // List of worst
6	$LQF = size(T_{Lat\_Worst\_List}) / size(T_{Lat\_list})$ // calculate link quality factor
7	Clear the lists T_{Lat_list} , $T_{Lat\_Worst\_List}$ and reset $i = 1$ // reset all the parameters for next
8	**end**

It is ideal for maintaining the LQF value to less than 0.2 in perfect conditions and 0.15 in harsh conditions so that smooth communication happens without any interruption or disconnection of BLE devices. Link quality and retransmission factors are used as critical features for the CI prediction model. As shown in table 1, if the window size is too large, current consumption reduction is not efficient as convergence and adaption to the latest link quality change is slower. Convergence and adaptation are faster for lower window size, so power saving is also efficient.

Table 1. Inference on different window sizes for link quality and CI estimation.

Window Size	No of adaptations			% of T_{Lat} exceeds $T_{Lat_{Th}}$			Average current consumption (uA)		
	ML	Spork	Data	ML	Spork	Data	ML	Spork	Data
20	2	27	12	2.9	8	4.8	176.08	530.76	204.34
60	2	6	2	2.8	7.6	3.1	219.28	603.15	224.11
120	2	3	2	3	9	4.1	285.28	613.65	288.98

Also, for lower window size, link quality is overestimated whenever there is a small burst of instances that crosses threshold latency, and hence many dynamic adaptations are happening. More diminutive or higher window sizes cause more number of latency instances exceeding threshold latency. A medium window size helps capture the link quality adequately and is a bit of a trade-off with faster convergence and better latency performance. If the window size is large, adaptation to the subsequent estimation of CI would be delayed.

3.2 Test Data Generator

The host layer can control the parameters such as Data packet length, maximum characteristics length (ATT_MTU), Transmission current, connection interval, and data rate using HCI commands. An analytical approach in terms of latency and retransmission factor would help us arrive at the global optimal operating point faster in both ideal and environments with interference. Dataset of size more than 10000 data points are generated for both 1Mbps, and 2Mbps speeds at the master BLE. It is inferred that the data rate 2Mbps consumes less current compared to the data rate 1Mbps. Also, the 2Mbps data rate helps us achieve almost twice the throughput achieved by 1Mbps. Data packet size is proportional to the radio ON time (transmission period) and hence affects the current consumption. It may be assumed that having more data packet length might lead to high current consumption. The maximum data packet length will help the radio transmit the data (if the payload is less than 247 bytes) as a single packet due to lesser radio ON and OFF switching. Lesser radio ON time, and hence the BLE module consumes low current. Different data packet sizes are used in the current research work, and results are noted. If the connection interval is low, radio ON and OFF switching will be more frequent, and hence more current is drawn. At the same time, if Connection Interval is very high, it increases the latency of data transmission.

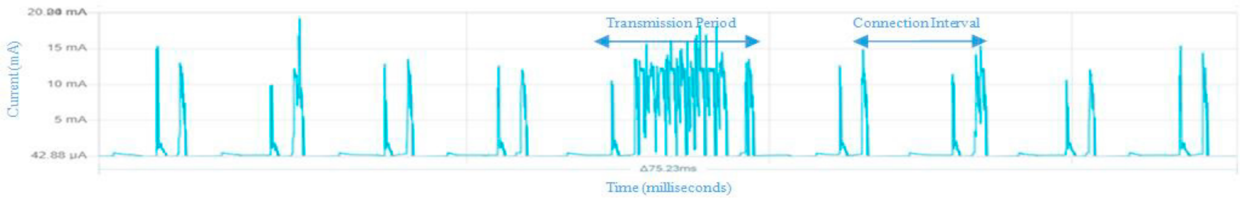

Fig. 3. Radio On/OFF during transmission (PPK -2).

Power profiler kit – 2 (PPK-2) measures current from one of the slave NRF modules as shown in fig 6 and is directly read using a python library from IRNAS[14] through a serial port as shown in fig 3. Since the Connection Interval parameter directly impacts the current consumption and latency of the application, it is considered the target variable. It is observed that the current consumption and retransmission factor are having direct correlation. The retransmission factor and connection interval have a negative correlation till the optimal point and indicate that as the connection interval increases, the retransmission factor also reduce, i.e., congestion in the network is reduced. The retransmission factor and number of peripherals have a positive correlation which implies that as the number of peripherals increases, the retransmission factor also increases.

3.3 Interference in ISM band:

As the BLE network is designed to withstand the harsh RF environment in the Electric vehicle, it is essential to validate the generated model and the lookup table-based algorithm in the presence of interference generated in the ISM band (2.4 GHz). Peer BLE interference is generated using ESP 32 BLE modules communicating to the raspberry pi module. Wifi interference is generated by using mobile phones streaming high-resolution videos.

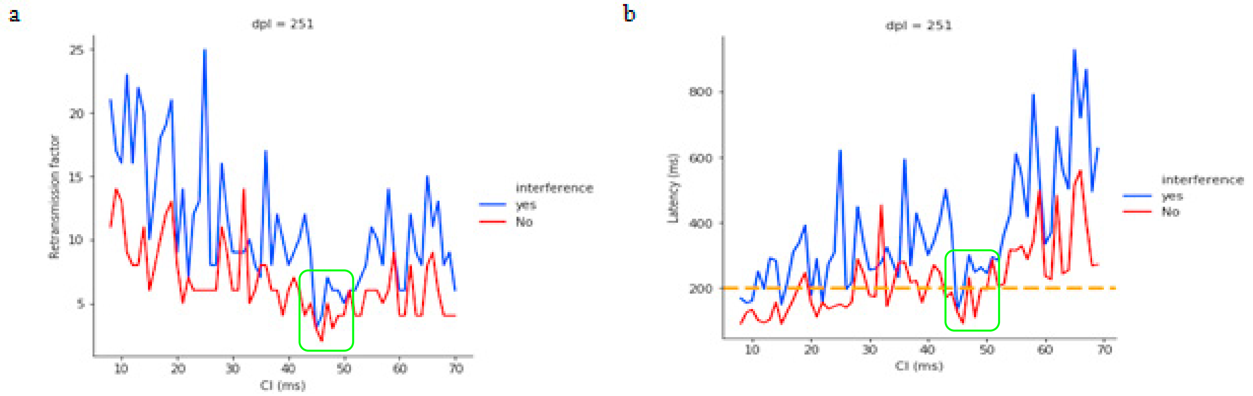

Fig. 4. (a) CI Vs Retransmission factor; (b) Latency with Interference.

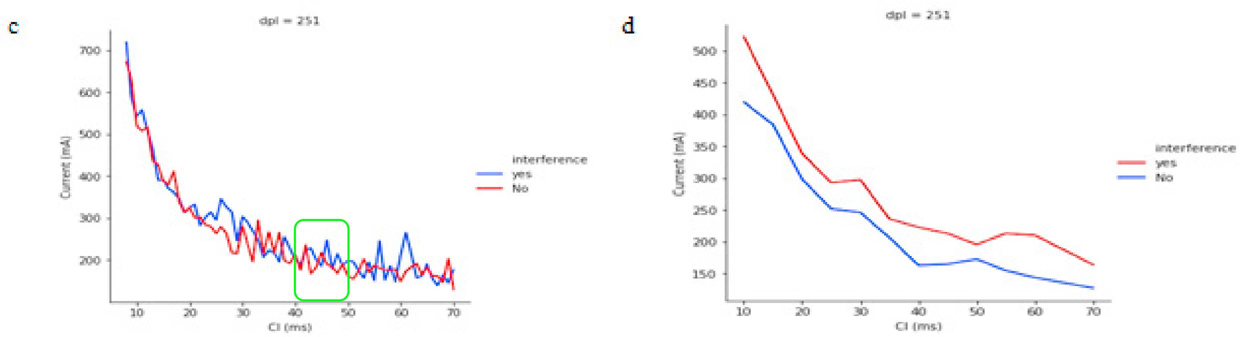

Fig. 4. (c) CI Vs Current (5 Slaves); (d) CI Vs Current (6 Slaves).

Passengers inside electric vehicles are more likely to use BLE-enabled devices and Wifi services for their daily activities. As observed in fig 4 (a, b, c, d) (test data generated for 5 slaves 1 master network), the retransmission factor is increased, latency is almost doubled, and current consumption is increased during interference. So the BLE network of the battery management system should be robust enough to handle such harsh RF interferences. It is observed that there are optimal operational points when the BLE network is operated in those points, the impact of interference is minimal, i.e., Retransmission factor in those operating points are low along with lower current consumption and lower latency as highlighted by a green box in fig 4 (a, b, c). Spork's algorithm[8] can reach the local minima of that operational point to ensure the latency requirement alone is satisfied but allows more retransmissions. However, with the SMART-A-BLE algorithm and a higher bound for latency, a lower bound of the latency is considered along with reduced current consumption

3.4 SMART-A-BLE Algorithm

SMART-A-BLE algorithm is split into four significant steps Link quality estimation, connection interval prediction, validation of the predicted connection interval, and connection interval adaptation as described in Algorithm 1, 2, 3 (a) and 3 (b). A decision tree regression model is created for predicting Connection Interval with input features such as “expected threshold latency”, “allowed Retransmission factor”, “number of peripherals”, and “data packet length” with an accuracy of 99.71% with training data and test data.

Algorithm 2: *SMART-A-BLE Scheme*

Define Window size $N = 20$, $i = 1$, R_{Factor} =2 // Initialization of variables
Append the latency measured to the list T_{Lat_list} // Last N latency measurements
While $i\%N == 0$
    Estimate Link Quality Factor
    If $LQF < 0.05$ // Link condition is good as the link quality factor is below 0.05
        IncreaseCI ($T_{Lat_{Th}}$, R_{Factor}) // Saves current consumption by increasing CI
    If $LQF > 0.15$ // Link condition is bad as the link quality factor is above 0.15
        DecreaseCI ($T_{Lat_{Th}}$, R_{Factor}) // Increases link quality by decreasing CI
        If $LQF > 0.2$
            Blacklist the current CI as it is not performing well
    end
end

Fig 4 (a, b, c, d), 5 represents the model creation process, and the data set used both data points generated without interference and with BLE and Wifi interference. Hence the created model is resilient enough to handle even harsh automotive environmental conditions as it can pick the best optimal operational point in both scenarios.

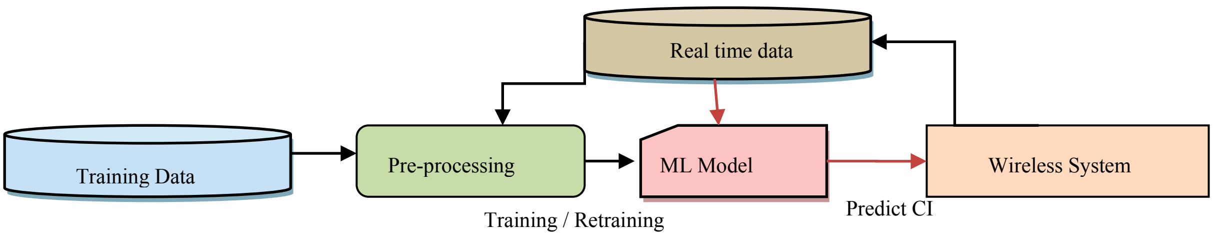

Fig. 5. Machine learning model accuracy improvement architecture.

Sometimes the generated CI prediction model predicts a non optimal operating point, resulting in more violations on threshold latency. A derived categorical feature was added to the dataset which categorizes each connection interval in the dataset based on the percentage of generated latency instances within the latency window. The connection intervals that generated more than 80% of latency measurement between 0 ms to 100 ms are labeled as category ‘1’, similarly 100 ms to 200 ms as category ‘2’, and 200 ms to 300 ms as category ‘3’. Another prediction model using a decision tree classifier algorithm is created to predict the category of the estimated connection interval. If the threshold latency is between 0 to 100 ms, the estimated CI will be valid only if the model predicts its category as '1' as defined in the dataset. Suppose the estimated category for the estimated CI is '2'. In that case, it becomes invalid as setting this CI would result in threshold latency violations, and hence CI should be re-estimated as shown in the above algorithm.

Algorithm 3 a: ***Decrease Connection Interval***

DecreaseCI($T_{Lat_{Th}}, R_{Factor}$)
    $R_{Factor} = R_{Factor} + 2$ // Increasing the R_{Factor} to decrease CI
    #estimate new CI
    EST_CI = ML($dpl, N_{Peri}, T_{Lat_{Th}}, R_{Factor}$)
    isValid = ValidateEstimatedCI_ML(dpl, N_{peri}, Est_CI)
    If(isValid) → send a CI update request to Master using Nordic softdevice API
    If(!isValid) → re-estimate CI by changing R_{Factor}
end

Algorithm 3 b: ***Increase Connection Interval***

IncreaseCI($T_{Lat_{Th}}, R_{Factor}$)
$R_{Factor} = R_{Factor} - 2$ // Reducing the R_{Factor} to increase CI
#estimate CI
EST_CI =ML($dpl, N_{Peri}, T_{Lat_{Th}}, R_{Factor}$)
isValid = ValidateEstimatedCI_ML(dpl, N_{peri}, Est_CI)
If(isValid) → send a CI update request to Master using Nordic softdevice API
If(!isValid) → re-estimate CI by changing R_{Factor}
end

SMART-A-BLE algorithm is suitable for the Master module supported by a small computing element like Raspberry pi Model 4 with at least 4GB RAM to support the environment required for deploying the model. A memory constraint environment can implement the data-based algorithm where the control algorithm can run on the master BLE module. A test-bed setup is made inside a metallic container to mimic an automotive battery pack for experimentally verifying the algorithm. Also, the verification of the algorithm is done with and without interference caused by other BLE modules and Wifi – devices, as shown in fig 6. Test-bed is validated using Spork's mathematical model and SMART-A-BLE algorithms with initial connection interval = 7.5 ms, data packet length = 251 Bytes, Characteristic size = 244 Bytes, Connection Event = 6 ms, and data rate = 2Mbps. For every 20 instances (window size) of latency recorded, the algorithm validates the link quality factor, and the estimated Connection interval is applied master and all the slaves. Connection interval or any other control parameters are changed using UART communication between Master and Slave devices using APIs exposed by the soft-device of Nordic semiconductor. The test setup has 1Master + 5 slave devices (2 ESP32 and 3 NRF 52840) connected with star topology. A payload of size 244 bytes is sent every second from the slave BLE modules to the master BLE module, and Latency, Retransmission factor, and current readings are noted with Spork's and SMART-A-BLE and data algorithms.

Algorithm 5: *Data based algorithm*

Sort the data using R_{Factor} in Ascending order
Fetch all CI from the lookup table with $T_{Lat_{Th}}$ as constraint
While $i\%N == 0$
    Estimate Link Quality Factor
    If $LQF < 0.05$
        Select CI with lower R_{Factor}
    If $LQF > 0.15$
        Select CI with higher R_{Factor}
        If $LQF > 0.2$
            Blacklist the current CI
    end
end

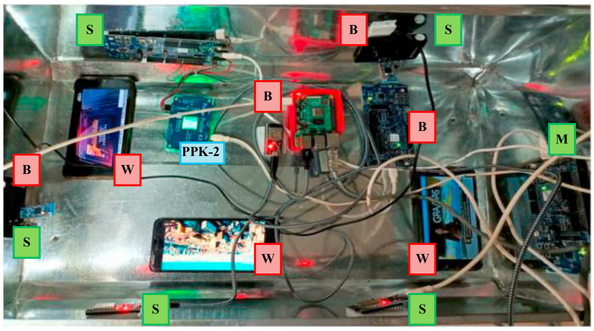

Fig. 6. Test Bed with a Master (M – green box) and 5 slave (S - green box) devices and interference generators (B Red box – BLE interference and W Red box – WIFI interference).

4. Results and discussions

Fig 7 (a, b) compares the current consumption of a slave device using the SMART-A-BLE algorithm with the existing mathematical model used in Spork's algorithm. The existing Spork's method mainly focused on operating the network within threshold latency but was unable to reduce the device's power consumption. Park's[9] method also uses the same mathematical model used by Spork's method to estimate a connection interval which is not optimal but was able to reduce power consumption by increasing the data rate. The proposed algorithm with ML model reduced more than 50% of the current consumption in ideal conditions and up to 50% of current consumption during interference and without line of sight conditions. If the connection interval of BLE network is high, then the radio ON/OFF switching happens at a lower frequency, hence consuming less current. If the connection interval is small, then the radio ON/OFF switching occurs at a higher rate; hence, current consumption is increased. From fig 7 (c,d) it can be observed that the predicted connection interval by existing Spork's model is always lower and is not optimal value, whereas it is higher in case of ML and Data based estimation, which corresponds to lower current consumption as shown in fig 7 (a,b). From fig 7 (c, d, e, f, g, h), it is clear that ML and Data algorithms perform better than the Spork algorithm[8] by estimating an optimal connection interval such that power consumption is highly reduced without sacrificing link quality.

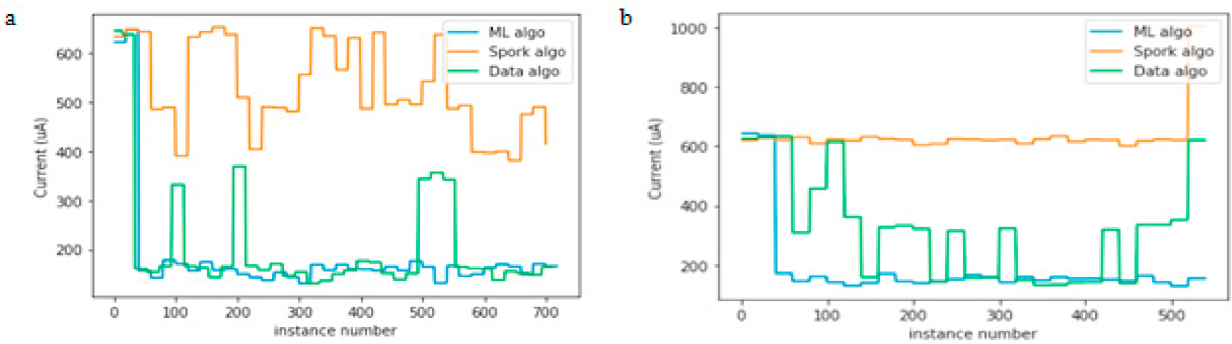

Fig. 7. (a) Current consumption in ideal condition; (b) Current consumption with interference

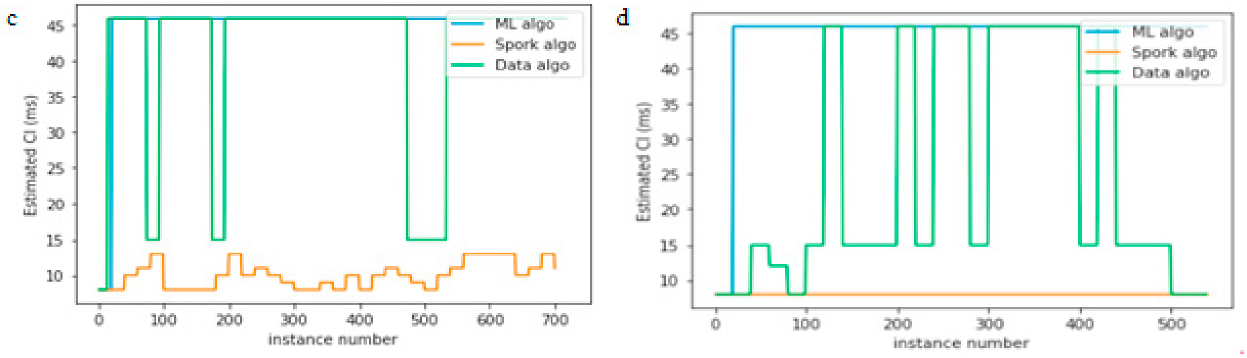

Fig. 7. (c) Connection Interval in ideal condition; (d) Connection Interval with interference

Sometimes ML algorithm estimates CI that might be the near-optimal point but not exactly the optimal operating point. These instances can be reduced by adding the proper validation on the estimated CI using category estimation model. Fig 7 (e, f) shows that all three algorithms can operate the network within the threshold latency of 100 ms. There are always instances where the threshold latency is violated while operating the network using all three algorithms due to delayed data packets. However, all three algorithms have more latency violation cases with interference, as shown in fig 7 (h), compared to an ideal condition, as shown in fig 7 (g). ML and data-based algorithms are better, with fewer latency violations than Spork's algorithm. From fig 7 (i, j), ML and Data algorithm reduces retransmission factor from an average of 8 to 2 in ideal conditions and interference conditions compared to Spork[8]. Since the retransmission factor is proportional to the number of times, the data packets are retransmitted, reducing it will also reduce current consumption as the transmission of each packet will have higher current consumption, as shown in fig 3.

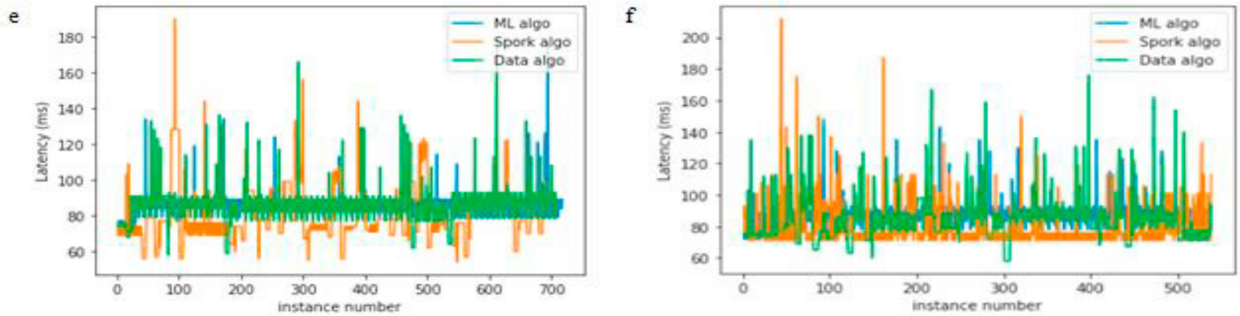

Fig. 7. (e) Latency in ideal condition; (f) Latency with interference

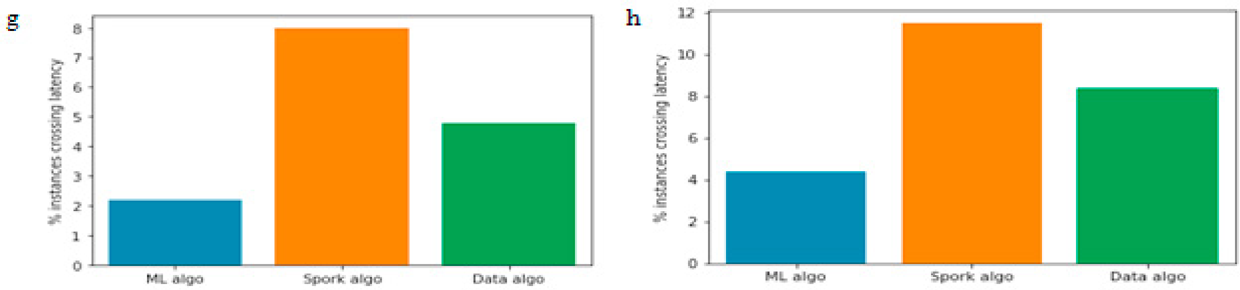

Fig. 7. (g) Latency exceeding threshold in ideal condition; (h) Latency exceeding threshold with interference

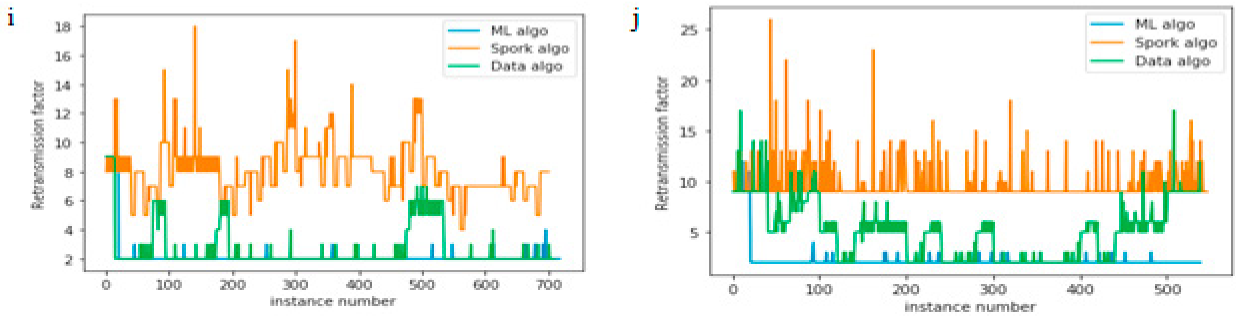

Fig. 7. (i) Retransmission Factor in ideal condition; (j) Retransmission Factor with interference

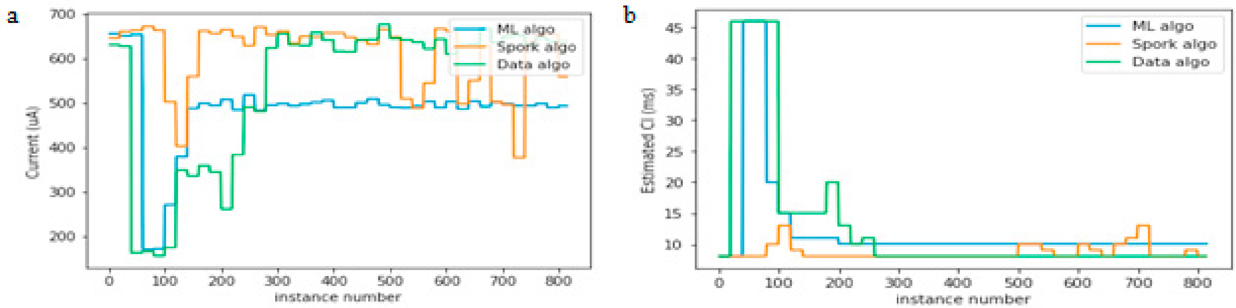

Fig. 8. (a) Current consumption without LOS; (b) Connection interval without LOS

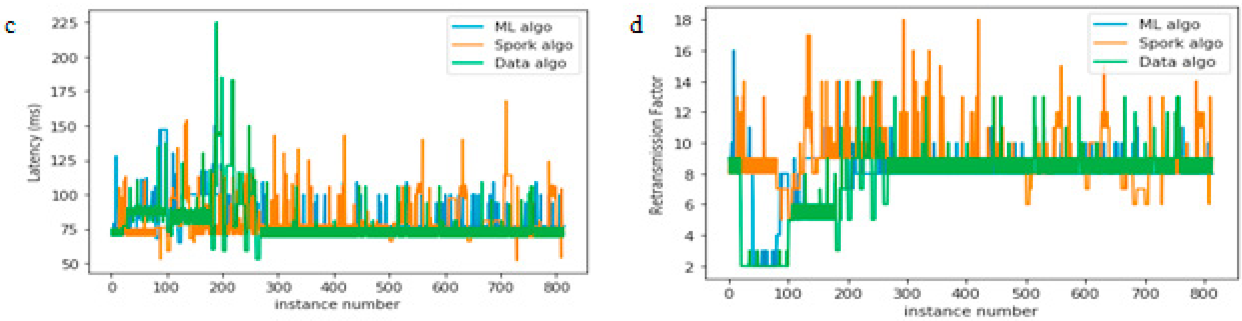

Fig. 8. (c) Latency without LOS; (d) Retransmission factor without LOS

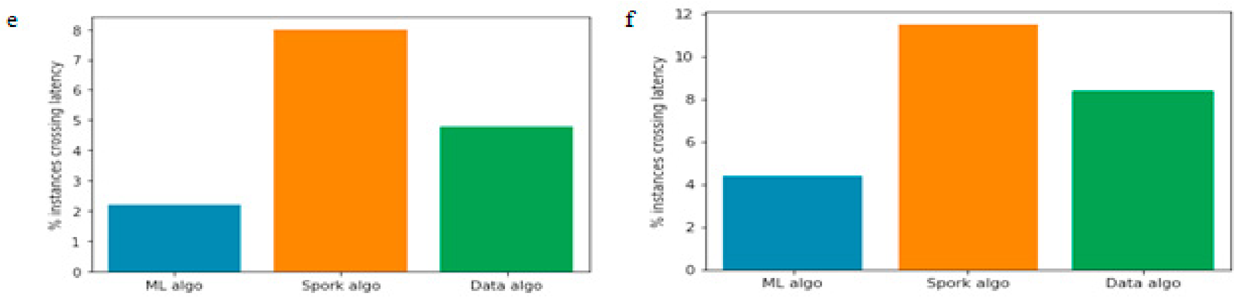

Fig. 8. (e) Latency crossing threshold without LOS; (f) Average current consumption without LOS

From fig 8 (a, b, c, d, e, f), ML and Data algorithms can reduce current consumption and retransmission factor without compromising link quality even in without line of sight, keeping the master device at a distance of 30 cms from the closed metallic box with slaves devices as compared with the existing Spork's algorithm. Hence, all the BLE slave modules in the network can be operated at a low current consumption mode without sacrificing the link quality. The retransmission factor representing the number of retransmissions is reduced along with current consumption, leading to less congestion in the network, indirectly ensuring robust connectivity. Even without line of sight, SMART-A-BLE and Data based algorithm is able to perform well in terms of both maintaining better link quality as well as low current consumption.

5. Conclusion and Future Scope

In this study, a connection-oriented BLE network is established with a single master and 5 slave devices (1 NRF 52840 DK and 4 NRF 52840 dongles) in a closed metallic container which would be similar to an automotive battery pack environment. A dataset involving performance parameters like latency and retransmission factor is captured by varying connection interval, data rate, data packet length and transmission power. The retransmission factor is calculated based on the mathematical model used in Spork's algorithm to capture the level of interference. In data visualization, the optimal connection interval is found as shown in fig 4 (a, b, c), where the number of retransmissions are reduced along with lower current consumption without violating the threshold latency. An existing mathematical model for estimating connection interval is analyzed in the established network and found that it was able to reach only the local minima and was not able to reach the global minima. An ML-based or system-specific data-based control algorithm converges to the optimal operating point faster than a generic mathematical model with a bit of sacrifice in memory. The proposed and existing algorithm are verified in the established BLE network with different vendor modules (2 NRF 52840 dongle, 1 NRF 52840 DK, and 2 ESP32 Wroom). It is observed that the proposed algorithm with ML model is able to predict optimal connection interval dynamically applying a constraint on retransmission factor and threshold latency by monitoring the current link quality of the real-time network. By selecting the appropriate connection interval, the network achieved a lower number of data packet retransmissions of around 2 and lower current consumption up to 50% without violating threshold latency in both ideal and interference scenarios. The objective of the paper is achieved by achieving efficient current consumption by estimating the appropriate connection interval, which reduces the number of retransmissions without violating the threshold latency requirement. This control algorithm can be used in any real-time safety-critical system to establish robust connectivity with less congestion. With the same dataset generated in this work, another decision tree classification model to identify the presence of interference is created. For the given observation, the model can predict the presence of interference in the battery pack, which can be intimated as an alert to the passengers and the central server so that passengers can minimize Bluetooth usage or Wifi usage. Since this implementation is used at the master module, the BLE module from any vendor that supports BLE 5.2 can be used as slaves. As a future scope, this algorithm can be validated with more peripherals and with real-time vehicle batteries for its performance. Other performance parameters like throughput and packet delivery rate with larger payloads can also be explored to fine-tune the algorithm based on these parameters. Battery swapping is an emerging solution for Electric vehicles to resolve consumer anxiety. The proposed work can be helpful in battery swapping applications to easily swap the battery module and connect to the existing BLE network, even if it involves BLE devices from multiple vendors. Implementation of these features would ease the after-sales expenditure of the customer and the service provider.

The AFH algorithm in BLE devices can also be optimized and standardized to get expected behavior among different BLE vendor modules during harsh RF environments. Security enhancement is the next vital factor for a Wireless Battery Management System. Suppose the security of the established wireless network in BMS is compromised, it is possible to change the SOC of the battery with the wrong data, and it might trigger a thermal runaway in vehicle batteries[15], [16].

References:

[1] Thodupunoori H, Nair BB. (2019) “IoT Based Real Time Vehicle Vital Parameter Monitoring and Analytics”, in International Conference on Intelligent Computing, Information and Control Systems. Springer, Cham: 365-373.

[2] Samanta A, Williamson SS. (2021) “A Survey of Wireless Battery Management System: Topology, Emerging Trends, and Challenges” Electronics,10(18):2193.

[3] Mahmoud MS, Mohamad AA. (2016) “A study of efficient power consumption wireless communication techniques/modules for internet of things (IoT) applications”

[4] Del-Valle-Soto C, Valdivia LJ, Velázquez R, Rizo-Dominguez L, López-Pimentel JC. (2019) “Smart campus: An experimental performance comparison of collaborative and cooperative schemes for wireless sensor network”, Energies,12(16):3135.

[5] Liya ML, Aswathy M. (2020) “LoRa technology for Internet of Things (IoT)”, in 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) IEEE:8-13.

[6] Nagarajan B, Shanmugam V, Ananthanarayanan V, Bagavathi Sivakumar P. (2020) “Localization and indoor navigation for visually impaired using bluetooth low energy”, in Smart systems and IoT: innovations in computing. Springer, Singapore:249-259.

[7] Rondón R, Gidlund M, Landernäs K.(2017) “Evaluating bluetooth low energy suitability for time-critical industrial iot applications”, International Journal of Wireless Information Networks,24(3):278-290.

[8] Spörk M, Boano CA, Römer K. (2020) “Improving the timeliness of Bluetooth low energy in dynamic RF environments”, ACM Transactions on Internet of Things,1(2):1-32.

[9] Park E, Lee MS, Kim HS, Bahk S. (2020) “AdaptaBLE: Adaptive control of data rate, transmission power, and connection interval in bluetooth low energy”, Computer Networks 181:107520

[10] Farej ZK, Saeed AM. (2020) “Analysis and Performance Evaluation of Bluetooth Low Energy Piconet Network”, Open Access Library Journal,7(10):1-1.

[11] Lee T, Han J, Lee MS, Kim HS, Bahk S. (2017) “CABLE: Connection interval adaptation for BLE in dynamic wireless environments”, In 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON),IEEE:1-9.

[12] Jeon WS, Dwijaksara MH, Jeong DG. (2016) “Performance analysis of neighbor discovery process in Bluetooth low-energy networks”, IEEE Transactions on Vehicular Technology 66(2):1865-71.

[13] Shell C, Henderson J, Verra H, Dyer J. (2015) “Implementation of a wireless battery management system (WBMS)”, In IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings,IEEE:1954-1959.

[14] IRNAS. (2020) “Python API for PPK-2”.

[15] Pallavi S, Narayanan VA. (2019) “An overview of practical attacks on ble based iot devices and their security”, In 5th international conference on advanced computing & communication systems (ICACCS),IEEE:694-698.

[16] Thomas A, Gireesh Kumar T, Mohan AK. (2018) “Neighbor attack detection in internet of things”, In Advanced Computational and Communication Paradigms, Springer, Singapore:187-196.