Knowledge-based stuck-at fault analysis for wireless IoT smart meters in smart grid

I.F. Siddiqui[1], N.M.F. Qureshi[2], M.A. Shaikh[3], B.S. Chowdhry[4], A. Abbas[5] and S.U-J. Lee[6]

[1,5,6] *Department of Computer Science and Engineering, Hanyang University ERICA, Ansan, South Korea*

[2] *Department of Computer Science and Engineering, Sungkyunkwan University, Suwon, South Korea*

[3] *Pakistan Scientific and Technological Information Centre, Pakistan*

[4] *Faculty of Electrical Electronics and Computer Engineering, Mehran University of Engineering and Technology, Pakistan*

Abstract Smart meters have a built-in process of collecting and storing sensor information into smart grid repository. This data storage consists of four phases i.e. (i) detect unit information, (ii) bundle information, (iii) dispatch bundle through a communication channel, and (iv) information storage in the smart grid. At first, Static Random-Access Memory (SRAM) records unit information and bundles it into Nand flash memory. The meter then dispatches bundle information towards a distribution end unit and stores dataset into grid repository. The smart grid performs data analytics and observes the performance of smart meters related to their lifespan, health, power consumption and load management. The bundling information process functions non-stop and simultaneously decreases write endurance of flash memory. At a certain point, where flash memory consumes endurance rating of a cell, the stuck-at fault appears in the block of flash memory. The data being bundled in this block gets corrupt and transferred to distribution end. Consequently, data analytics do not produce accuracy in analyzing the performance of smart meter sensor unit. In this paper, we present Smart Meter Stuck-at Fault Analyzer (SMSFA) that performs fault analysis through unit information detected over sensor unit and predicts possible stuck-at fault meter. The simulation results show that SMSFA effectively collects flash error samples and identifies fault points that reduce computing cost of performing error analytics in overall data analytics of smart grid.

Keywords *Wireless IoT Smart meter, Smart grid, HBase, Stuck-at, Hadoop.*

Corresponding author:

Prof. Scott Uk-Jin Lee

Department of Computer Science and Engineering
Hanyang University ERICA
Ansan, South Korea

Phone: +82 31 400 4754
E-mail: scottlee@hanyang.ac.kr

1. Introduction

The smart grid has delivered the concept of automation and resource optimization in power sector. It is a huge inter-connected infrastructure that uses advancement in ICT technology and performs efficiently than traditional power grids [1]. The core components of smart grid include giant storage, power system, interoperability, electric transportations, distributed controls, integration of renewables, efficiency, and reliability of system, information technology, infrastructure security and wireless IoT devices [2], as shown in Fig. 1.

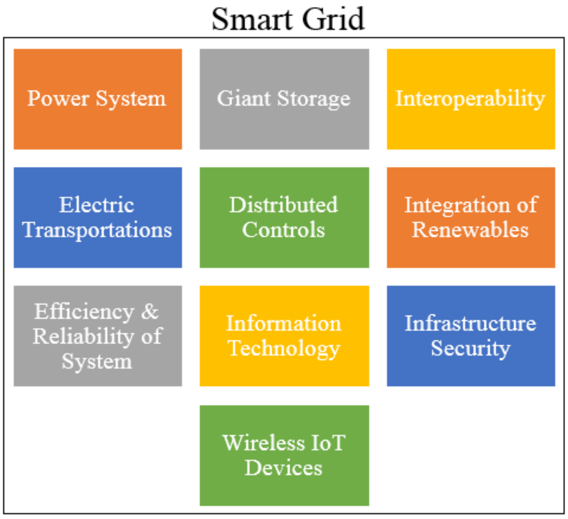

Fig.1. Components of Smart Grid

Smart grid records sensor information through wireless IoT devices i.e. smart meters [3]. The smart meter includes a complete data management system at small scale that sends data bundles to the grid repository [4]. The sensor information of unit is stored in two steps, (i) record information, and (ii) bundle information. The information recording step involves a software application that receives 5 columns of information in a single row of Static Random-Access Memory (SRAM) and assigns 5 bytes for short string type of columns. The size of SRAM is 256 KB [5]; therefore, it holds a dataset D_i= '52428' rows per transaction. The dataset D_i is bundled into a local storage facility of Nand flash memory, where size equals to 1 MB. Therefore, it generates 4 bundles of information per unit [6].

The flash memory is non-volatile in nature and each cell records charge with a limited write endurance. The meter generates bundles over SLC (Single Level Cell) layer flash memory that has endurance rating of '100,000' erases per block [7]. Therefore, when flash memory tends to finish endurance rating, stuck-at fault appears in the block of flash memory [8]. The process of bundling information is not aware of this hardware problem and smart meter dispatches dataset to the distribution end [9]. In the last, it becomes part of smart grid repository. Therefore, when smart grid performs data analytics, it does not produce accuracy in examining the status of the sensor unit and maintains a percentile of error in data analytics [10]. Thus, the overall performance of smart grid severely affected with the increasing factor error percentile over the passage of time.

A smart meter is an efficient device and erases flash data over each transaction of bundle information [11]. Therefore, when a block cannot be erased due to a stuck-at fault, meter assign random block address to store a new batch of information [12]. Therefore, it is hard to locate stuck-at problem at meter end.

To resolve this issue, we propose Smart Meter Stuck-at Fault Analyzer (SMSFA) that performs fault analysis within individual blocks of bundle information. SMSFA detects a pattern of abnormality in bundle data and predicts stuck-at victim smart meter, hence bundle information processing could no longer be processed over same memory block.

The significant contributions of the proposed approach are:

- A bundle information dataset scanner for smart meter dataset.
- A HBase based bundle information analyzer.
- A precise prediction strategy to figure out stuck-at victim smart meter.

The remaining paper is organized as follows. Section II briefly explains proposed approach SMSFA. Section III elaborates experimental environment and results. Finally, section IV presents conclusion and future research directions.

2. Smart Meter Stuck-at Fault Analyzer (SMSFA)

The proposed approach SMSFA performs block analysis over individual blocks of bundle information and searches abnormal pattern of column data. The abnormal patterns are stored into an error analyzer and prediction analysis is performed to figure out victim meter. SMSFA consists of three components i.e. (i) Bundle information dataset scanner, (ii) HBase error bundle analyzer, and (iii) Stuck-at fault predictor.

2.1 Bundle information dataset scanner

The internal working of bundle information dataset scanner is categorized in two components i.e. (i) Block scanner, (ii) Error sample bundler.

2.1.1. Block Scanner

Block scanner receives block B_i of smart meter SM_i having 5 columns of row information i.e. SMID, unit location, temperature, load and time stamp. The column information of block B_i can be represented as,

$$B_i = (SM_m, Loc_i, T_n, L_n, Time) \qquad (1)$$

SMSFA performs modified TSPLFC [13] algorithm over block B_i with two changes i.e. (i) scanning 5 columns at a time using multiple threads and (ii) concurrent identification of abnormal elements into block B_i.

Let's assume that we scan block B_i = [0…m-1,0…n-1, 0…k-1, 0…z-1, 0…c-1] through abnormal patterns i.e. M_m = [0…w-1], L_i = [0…r-1], N_n = [0…y-1], G_n = [0…g-1] and D = [0…v-1]. The modified algorithm TSPLFC compares columns of B_i over set of patterns (M_m, L_i, N_n, G_n, D_i) and fetches a result set in array of R_i. The scanning processing continues till set (SM_m [0], SM_m[m-1]) do not find M_m[i+1,w-1], set (Loc_i [0], Loc_i[n-1]) do not find L_i[i+1,r-1], set (T_n [0], T_n[k-1]) do not find N_n[i+1,y-1], set (L_n [0], L_n[z-1]) do not find G_n[i+1,g-1] and set ($Time$ [0], $Time$[c-1]) do not find D[i+1,v-1] and terminate its execution through returning index value as seen from Fig. 2.

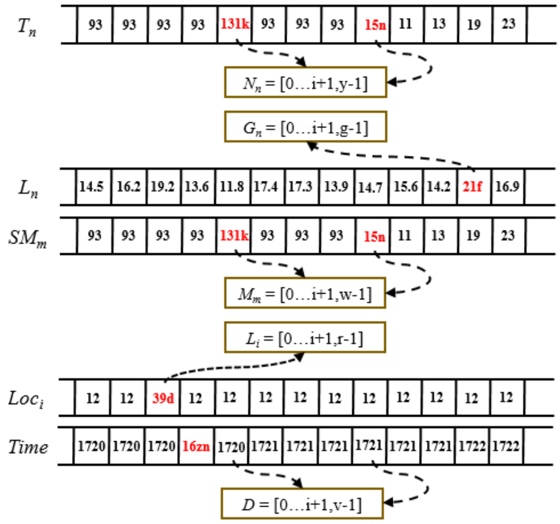

Fig. 2. Modified TSPLFC based Block Scanning

2.1.2. Error Sample Bundler

The Error Sample Bundler (ESB) is a function that writes N_n, G_n, M_m, L_i and D error bundle into a DFS file. At first, *ESB* method fetch row of N_n, G_n, M_m, L_i and D and stores into ESB_{Buffer}. Thus, we observe a single buffer row as,

$$ESB_{Buffer_i} = \sum_{R_i \leq B_i}^{Columns \in String(i)} (N_n, G_n, M_m, L_i, D) \quad (2)$$

DFS write socket then fetches single row of ESB_{Buffer_i} through index $Index_i$ method and uses *textoutputformat* [14] to write entries into a text file $ESB_{File[i]}$ as shown in Fig. 3.

SM ID	Location	Load	Temp	Time

Fig. 3. Error Sample Bundler dataset format

2.2 HBase Error Bundle Analyzer

The SMSFA uses HBase for storing file content of text file $ESB_{File[i]}$. It is a project associated with Hadoop ecosystem [15] and performs analytics over large datasets of the smart grid. SMSFA perform error analytics through table $SMError_{Table}$ in HBase repository as shown in Fig. 4.

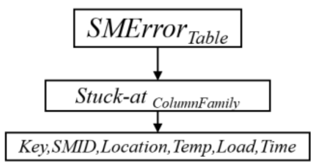

Fig. 4. HBase Smart Meter Error Repository

2.3 Stuck-at Fault Predictor

SMSFA predicts a stuck-at fault in two phases i.e. (i) Training of table dataset, and (ii) Stuck-at fault prediction.

2.3.1 Training of table dataset

The trainer performs a learning operation over 5 column datasets and observes column values for the possible stuck-at victim. For this purpose, we use Hidden Markov Model (HMM) [16] that trains a dataset through self-learning mechanism and requires an input range of values i.e. hidden states indicating the possibility of stuck-at fault. The HMM model requires few parameters i.e. hidden states $Z = \{z_1, z_2\}$, transition probability over a condition $A = a_{ij} = \{P[q_{t+1} = x_j | q_t = x_j]\}$, observation states $X = \{x_1, x_2, x_3, x_4\}$ and emission probability $B = b_{ij}$. After configuring model parameters, we find observation $O = \{B_1, B_2, B_3, B_4, B_n\}$ over hidden statues i.e. "stuck-at" and "Not Stuck-at" as shown in Fig. 5.

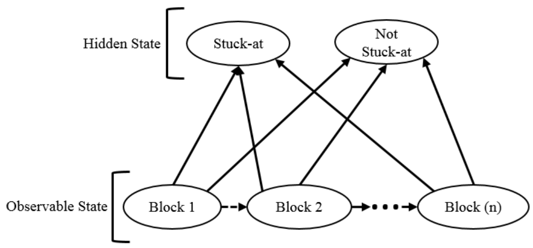

Fig. 5. HMM Observable and Hidden states

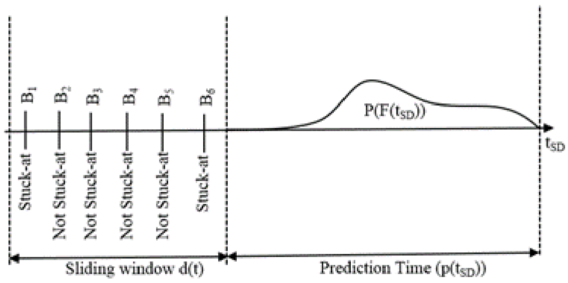

Fig. 6. Block stuck-at fault analysis

The training evaluation can be observed through,

$$\lambda = (\pi, A, B) \qquad (3)$$

Where π refers to initial state probability of transition matrix, B represents a matrix of emission $b_j(Y_t)$ and A is the matrix of transition that calculates transition probability from one state to another. After receiving observations O and hidden states, we perform prediction cycle that generates sliding window at time Δt and returns block status with time t_{SD} as shown in Fig. 6.

Thus, we find that table dataset is understood with the workload of 5 columns of table dataset. Furthermore, we train model through Expectation-Maximization (EM) algorithm [17], which finds likelihood elements in maximizing parameters and similarities until best fit result appears for table dataset. Therefore, at first, we calculate observation sequence probability (O_t, O_{t+1}, O_{t+2}) and use EM algorithm in two steps. The first step refers to the calculation of expected likelihood from current estimation and followed by the second step that calculates parameters of maximizing expectation for stuck-at status likelihood as shown in Algorithm-1.

Algorithm-1. Stuck-at fault predictor

Initialize O = {B_1, B_2, B_3, B_4, B_5, B_6}
S = {Stuck-at, not stuck-at}
M=2
N=6
Initialize Transprobability(A_{ij}), EmissionProbability(B_{ij})
Y = {Y_1, Y_2, Y_3, Y_4}
Initialize Probability(StatePath)
Update B_{ij},A_{ij}
Probability(Path) = B_{ij} * Probability(StatePath)
    For Each state S_j do
        Probability(StatePath)[j,i] ← $\max_k$ (Probability(StatePath)[k,i-1].A_{kj}.B_{jyi})
        Probability(Path)[j,i] ← arg $\max_k$ (Probability(Path)[k,i-1].A_{kj}.B_{jyi})
    End For
    C_i ← arg $\max_k$ (Probability(StatePath))
    Z_i ← S_i
    For I ← T-1, …, 1 do
        C_i ← Probability(Path)[Z_i,i]
    Z_i ← C_i
    End For
Produce (Time,Z)

2.3.2. Block stuck-at Prediction

SMSFA uses Viterbi algorithm [18] to analyze stuck-at fault and calculate hidden states. At first, the algorithm shows optimal state sequence and declares hidden states of the model $\lambda=(\pi,A,B)$ with observations $O = \{B_1, B_2, B_3, B_4, B_n\}$ and generates sequence of states S_{states}= {S_1,S_2,…, S_n} as,

$$S_{optimal} = argmax_s P(Sstates; O; \lambda) \qquad (4)$$

Where $S_{optimal}$ represents optimal state sequence. The Viterbi algorithm permits $S_{optimal}$ to fetch optimal paths at each step '*t*' that ends over *n* states. At *t+1*, *S* increases optimal path with an update in *n* state. At *t+2*, *S* reaches maxima job-likelihood and the optimal path gets updated with *n* state and predicts the hidden state i.e. '*stuck-at' or 'not stuck-at'* from collected observations O of bundle blocks as shown in Fig. 7.

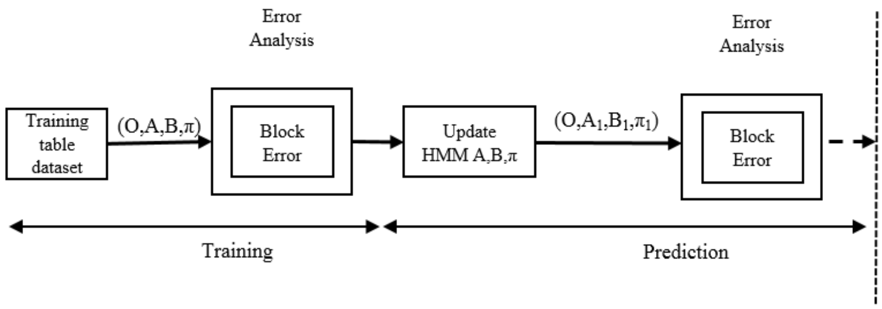

Fig. 7. Block error stuck-at prediction in Smart Meters

3. Experimental Evaluation

We evaluate SMSFA model through a configuration of the cluster as shown in Table 1.

Table 1. Cluster Configuration.

Machine	Specifications	No. of VM	
Intel Xeon E5-2600 v2	8 CPUs, 32GB memory, 1T Disk and 128 GB SSD	3	1 Master Node, 2 Datanodes
Intel core i5	4 Core, 16GB memory, 1T Disk and 128 GB SSD	2	2 Datanodes
Hadoop	Hadoop-2.7.2 (stable)		
Virtual Machine Management	VirtualBox 5.0.16		

The cluster includes Intel Xeon with 8 CPUs, storage media i.e. Hard disk drive 1TB, 128 GB Samsung SSD and 32GB memory. Similarly, we also use Intel core i5 with 4 Core, storage media i.e. Hard disk drive 1TB, 128 GB Samsung SSD and 16GB memory. For the virtual environment, we used VirtualBox 5.0.16 for installing 5 virtual machines on discussed cluster configurations as shown in Table 2.

Table 2. Virtual Machine Configuration Over Hadoop Cluster

Node	CPU	Memory	Disk	Configuration
Master Node	6	16 GB	HDD & SSD	Intel Xeon
Slave1	2	4GB	HDD & SSD	Intel Xeon
Slave2	2	4GB	HDD & SSD	Intel Core i5
Slave3	2	4GB	HDD & SSD	Intel Core i5
Slave4	2	4GB	HDD & SSD	Intel Core i5

The dataset used to process experimental work includes 100 smart meter block bundles of one month (2 GB dataset) [19].

3.1 Experimental Results

The experiments conducted to evaluate our scheme are (i) Block scanner (ii) Block stuck-at prediction.

3.1.1. Block scanner

To evaluate the efficiency of block scanner, we perform scan analysis over dataset columns i.e. smart meter id, location, temperature, load and time. We find that dataset having 100 smart meters' data bundle scan 0.08% error in smart meter id column, 0.03% error in location column, 0.09% error in temperature, 0.07% error in load column and 0.03% error in time column in '28371', '27142', '29432', '30993' and '25539' seconds respectively, as shown in Fig. 8.

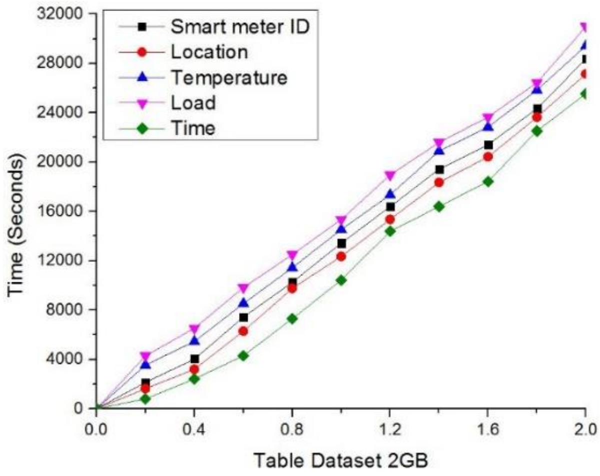

Fig. 8. Block scanner in table dataset

3.1.2. Block stuck-at Prediction

After successful training of table dataset, we perform prediction simulation to find possible stuck-at fault in bundle blocks of smart meters. SMSFA performed the simulation for three hours and evaluated that first hour prediction declares '528' bundle blocks with a status of "stuck-at" out of '15450' bundle blocks. Furthermore, the second hour of simulation predicts '831' bundle blocks with a status of "stuck-at" out '19530' bundle blocks. Moreover, the third hour of simulation predicts '1077' bundle blocks with a status of "stuck-at" out of '20750' bundle blocks as shown in Fig. 9.

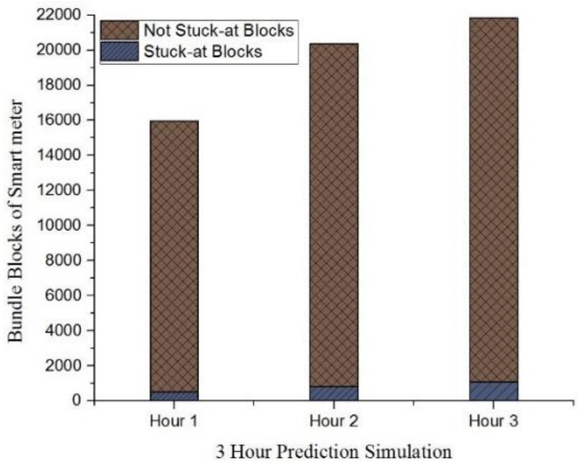

Fig. 9. Smart Meter prediction for hot socket issue

4. Conclusion

This paper proposes a novel knowledge-based stuck-at fault analyzer that scans bundle information for abnormal generation of column values, stores scanned dataset into HBase repository and performs prediction analysis to evaluate stuck-at blocks in smart meters. The experimental evaluation shows that SMSFA effectively identifies stuck-at fault bundles in large sensor dataset of smart meters and decreases the possibility of false data analytics in the smart grid.

In future, we will focus on sentinel management of smart meters in the smart grid.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korean government (MSIP) (No. NRF-2016R1C1B2008624)

References

[1] M. L. Tuballa and M. L. Abundo, "A review of the development of smart grid technologies," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 710–725, Jun. 2016.

[2] S. Bera, S. Misra, and J. J. P. C. Rodrigues, "Cloud computing applications for smart grid: A survey" *IEEE Transactions on Parallel and Distributed Systems*, 26(5), pp. 1477–1494, May 2015.

[3] B. L. Risteska Stojkoska and K. V. Trivodaliev, "A review of Internet of things for smart home: Challenges and solutions," *Journal of Cleaner Production*, Oct. 2016.

[4] S. Chren, B. Rossi, and T. Pitner, "Smart grids deployments within EU projects: The role of smart meters," *2016 Smart Cities Symposium Prague (SCSP)*, May 2016.

[5] G. KG, "Toshiba Smart meter MCUs", *Glyn.de*, 2017. [Online]. Available: http://www.glyn.de. Last accessed: 27- Apr- 2017.

[6] He, Jun, et al. "The Unwritten Contract of Solid State Drives." *Proceedings of the Twelfth European Conference on Computer Systems*. ACM, 2017.

[7] Compagnoni, Christian Monzio, et al. "Reviewing the evolution of the NAND Flash technology." *Proceedings of the IEEE* (2017).

[8] Chaudhry, Adnan Aslam, Cai Kui, and Yong Liang Guan. "Mitigating Stuck Cell Failures in MLC NAND Flash Memory via Inferred Erasure Decoding." *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems* (2017).

[9] V. C. Gungor et al., "Smart grid technologies: Communication technologies and standards," *IEEE Transactions on Industrial Informatics*, 7(4), pp. 529–539, Nov. 2011.

[10] V. C. Gungor et al., "A survey on smart grid potential applications and communication requirements," *IEEE Transactions on Industrial Informatics*, 9(1), pp. 28–42, Feb. 2013.

[11] D. Alahakoon and X. Yu, "Smart electricity meter data intelligence for future energy systems: A survey," *IEEE Transactions on Industrial Informatics*, 12(1), pp. 425–436, Feb. 2016.

[12] King, Jennifer, and Christopher Perry. "Smart Buildings: Using Smart Technology to Save Energy in Existing Buildings." (2017).

[13] M. Zubair, F. Wahab, I. Hussain, and J. Zaffar, "Improved text scanning approach for exact string matching," *Proceedings of International Conference on Information and Emerging Technologies*, Jun. 2010.

[14] A. S. Foundation, "Text output format API," 2016. [Online]. Available: https://hadoop.apache.org/docs/r2.7.2/api/org/apache/hadoop/mapreduce/lib/output/TextOutputFormat.html. Last accessed: Apr. 27, 2017.

[15] "Welcome to Apache™ Hadoop®," 2014. [Online]. Available: http://hadoop.apache.org/. Last accessed: Apr. 27, 2017.

[16] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks," *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1), pp. 9-42, 2001.

[17] Ajit Singh, "EM Algorithm", 2005.

[18] G. D. Forney, "The Viterbi algorithm," in *Proceedings of the IEEE*, 61(3), pp. 268-278, 1973.

[19] C.-L. N. Revolution, "Smart Meter Dataset," Customer-Led Network Revolution, 2016. [Online]. Available: http://www.networkrevolution.co.uk/project-library/dataset-tc1a-basic-profiling-domestic-smart-meter-customers/. Last accessed: Apr. 27, 2017.