Wireless Pers Commun
https://doi.org/10.1007/s11277-018-5739-9

Stuck-at Fault Analytics of IoT Devices Using Knowledge-based Data Processing Strategy in Smart Grid

Isma Farah Siddiqui[1] · Nawab Muhammad Faseeh Qureshi[2] · Muhammad Akram Shaikh[1] · Bhawani Shankar Chowdhry[3] · Asad Abbas[4] · Ali Kashif Bashir[5] · Scott Uk-Jin Lee[4]

Abstract Smart grid addresses traditional electricity generation issues by integrating ambient intelligence in actions of connected devices and production processing units. The grid infrastructure uses sensory IoT devices such as smart meter that records electric energy consumption and production information into the end units and stores sensor data through semantic technology in the central grid repository. The grid uses sensor data for various analytics such as production analysis of distribution units and health checkup of involved IoT devices and also observes functional profile of IoT equipment that includes

✉ Scott Uk-Jin Lee
scottlee@hanyang.ac.kr

Isma Farah Siddiqui
isma.farah@faculty.muet.edu.pk

Nawab Muhammad Faseeh Qureshi
faseeh@skku.edu

Muhammad Akram Shaikh
muhammad.akram@faculty.muet.edu.pk

Bhawani Shankar Chowdhry
dean.feece@admin.muet.edu.pk

Asad Abbas
asadabbas@hanyang.ac.kr

Ali Kashif Bashir
dr.alikashif.b@ieee.org

1 Department of Software Engineering, Mehran University of Engineering and Technology, Jamshoro, Pakistan

2 Department of Computer Education, Sungkyunkwan University, Seoul, South Korea

3 Faculty of Electrical, Electronics, and Computer Engineering, Mehran University of Engineering and Technology, Jamshoro, Pakistan

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

5 Faculty of Science and Technology, University of the Faroe Islands, Tórshavn, Faroe Islands

Published online: 17 April 2018

service time, remaining lifespan, power consumption along with its functional error percentile. In a typical grid infrastructure, AMI meters process continuous streaming of data with Nand flash memory that stores dataset in the form of charges such as 0 and 1 in memory cell. Although, a flash memory is tested through rigorous testing profile but the grid environment impacts its cell endurance capacity diversely. Thus, a cell gets stuck-at fault before the end of endurance and can not be used to override a new tuple into it. In this paper, we perform a knowledge-based analytics to observe these stuck-at faults by detecting the abnormal variation among stored data tuples and predicts the going-to-be stuck-at cells of AMI meter. The simulation results show that the proposed approach rigorously maintain a knowledge-based track of AMI devices' data production with an average error percentile of 0.06% in scanning blocks and performed prediction analytics according to the scanning percentile functional health and presents a work-flow to balance the load among healthy and unhealthy IoT devices in smart grid.

Keywords Wireless IoT smart meter · Smart grid · HBase · Stuck-at · Hadoop

1 Introduction

Smart grid is an automated power plant that uses intelligent techniques to maximize power generation and optimize resource consumption in parallel distributed environment. It is a gigantic infrastructure of inter-connected devices that produce sensory data and store into a semantic-enabled reservoir than than traditional power grid [1]. The grid layout consists of components such as semantic-enabled data reservoir, power production and management system, electricity capacitors, automated power lines, distributed control infrastructure, renewable energy resources, intelligent power consumption system, information technology infrastructure, grid security framework and IoT devices [2], as shown in Fig. 1.

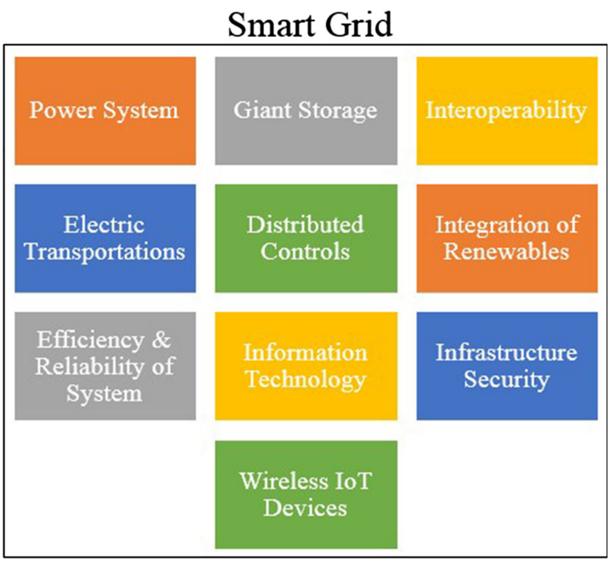

Fig. 1 Smart grid layout

Smart grid generates sensor data through wireless IoT devices such as smart meter [3]. The meter contains a portable data management system that processes sensor data into semantic tuples and produces batches of datasets such as bundles, which are then stored inside semantic-enabled reservoirs [4]. The internal mechanism of smart meter stores transformed semantic tuples in a single row of Static Random-Access Memory (SRAM) having 8 individual columns of 32 bytes each over 256 KB block and manages a data chunk D_i = 52428 rows per transaction [5]. The dataset D_i is then compiled in the form of a bundle using Nand flash memory having size equals to 1 MB. Thus, it generates 4 bundles of sensory data per device unit [6].

Nand 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 rate of 100,000 erases per block [7]. Thus, when cell endurance tends to a threshold limit $T_{Limit} \leq 5\%$, stuck-at fault appears over in-use block of flash memory [8] and stores faulty data chunk bundle in the semantic-enabled reservoir [9]. As a result, smart grid produces a percentile of inaccuracy in performing data analytics and maintains an error in grid dataset that increases with the passage of time and severely affects the overall performance of smart grid [10].

Smart meter manages block memory through flash translation layer (FTL) and performs erase block procedure on demand [11]. In the above discussed scenario, FTL returns erase cycle error that could not be addressed through error correction code (ECC) and results in shifting the block pointer to new data block without removing data charge over stuck-at fault block [12]. Thus, it becomes difficult to log stuck-at affected block and resolve the problem through direct or indirect solutions.

To resolve this issue, we propose Smart Meter Stuck-at Fault Analyzer (SMSFA) that performs fault analysis of in-use data blocks. The presented approach detects abnormal patterns of in-use blocks and predicts stuck-at fault through knowledge-based approach of smart meter. The significant contributions of the proposed strategy are:

- A semantic tuple scanner that detects abnormal patterns within the data chunk of in-use data block.
- An analyzer that processes tuple data chunk bundle in HBase.
- An effective training and prediction model to detect stuck-at fault through in-use block dataset.

The remaining paper is organized as follows. Section 2 discusses related work. Section 3 briefly explains the proposed approach (SMSFA). Section 4 presents experimental environment and results. And, Sect. 5 depicts conclusion along with future works.

2 Related Work

Stuck-at fault approach was introduced by Williams, T.W et al. in their research of functional defect levels [13] and later SD Millman et al. discussed its memory perspective in the Complementary metaloxidesemiconductor (CMOS). From then, many researchers explained their contributions related to the identification and recovery process of stuck-at faults with respect to RAM, SRAM, DRAM, ROM memories [14–27]. In the era of nand flash, stuck-at fault came in the limelight due to limited cell endurance issue [28]. Afterwards, many researchers presented their contributions to either overcome the stuck-at fault or reduce the possibility of a cell getting into stuck-at situation [29–42], however, to the best of our knowledge, none of them presented stuck-at fault analysis through knowledge-base perspective.

In this paper, we present a training and prediction model of analyzing stuck-at fault from sensory dataset that effectively addresses the issue by detecting the abnormal data chunk patterns of in-use memory blocks and predicts going-to-be stuck-at block before it gets near to the threshold percentile at the end of cell endurance.

3 Smart Meter Stuck-at Fault Analyzer (SMSFA)

In this section, we discuss proposed approach (SMSFA) that performs block status analysis over individual in-use blocks and get abnormal pattern of column data. The fetched patterns are then stored into an error analyzer for training and prediction analysis so that stuck-at fault could be carried out in a smart meter. SMSFA consists of three components i.e. (1) Data chunk bundle scanner, (2) HBase error analyzer, and (3) Stuck-at trainer and fault predictor.

3.1 Data Chunk Bundle Scanner

Data chunk bundle scanner is categorized in two components i.e. (i) Block scanner, (ii) Error sample bundler.

3.1.1 Block Scanner

Block scanner is a functional unit to perform scan analysis over in-use data block. It receives block B_i of smart meter SM_i having 5 columns of row data i.e. *SMID*, *unitlocation*, *temperaturevalue*, *loadvalue* and *timestamp*. The column of block B_i can be expressed as,

$$B_i = (SM_m, Loc_i, T_n, L_n, Time) \tag{1}$$

SMSFA filters out abnormality using modified TSPLFC [43] algorithm over block B_i with two enhancements i.e. (i) scanning 5 columns simultaneously using synchronous threads and (ii) concurrent identification of an element beyond abnormal range of block B_i.

For this purpose, we scan block B_i = $[0m-1, 0n-1, 0k-1, 0z-1, 0c-1]$ through abnormal patterns M_m = $[0w-1]$, $Li = [0r-1]$, N_n = $[0y-1]$, G_n = $[0g-1]$ and D_i = $[0v-1]$. 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 until array set ($SM_m[0]$, $SM_m[m-1]$ do not find $M_m[i+1, w-1]$ and array set ($Loc_i[0]$, $Loc_i[n-1]$ do not find $L_i[i+1, r-1]$, array set $T_n[0]$, $T_n[k-1]$ do not find $N_n[i+1, y-1]$, array set $L_n[0]$, $L_n[z-1]$ do not find $G_n[i+1, g-1]$ and array set $Time_i[0]$, $Time_i[c-1]$ do not find $D_i[i+1, v-1]$ and terminate its execution through returning index value as seen from Fig. 2.

3.1.2 Error Sample Bundler

The Error Sample Bundler (ESB) is a function that writes N_n, G_n, M_m, L_i and D_i identified array set of errors into a DFS file. At first, ESB method fetches row of N_n, G_n, M_m, L_i and D_i for storing into ESB_{Buffer}. Therefore, a single buffer of in-use data row can be expressed as,

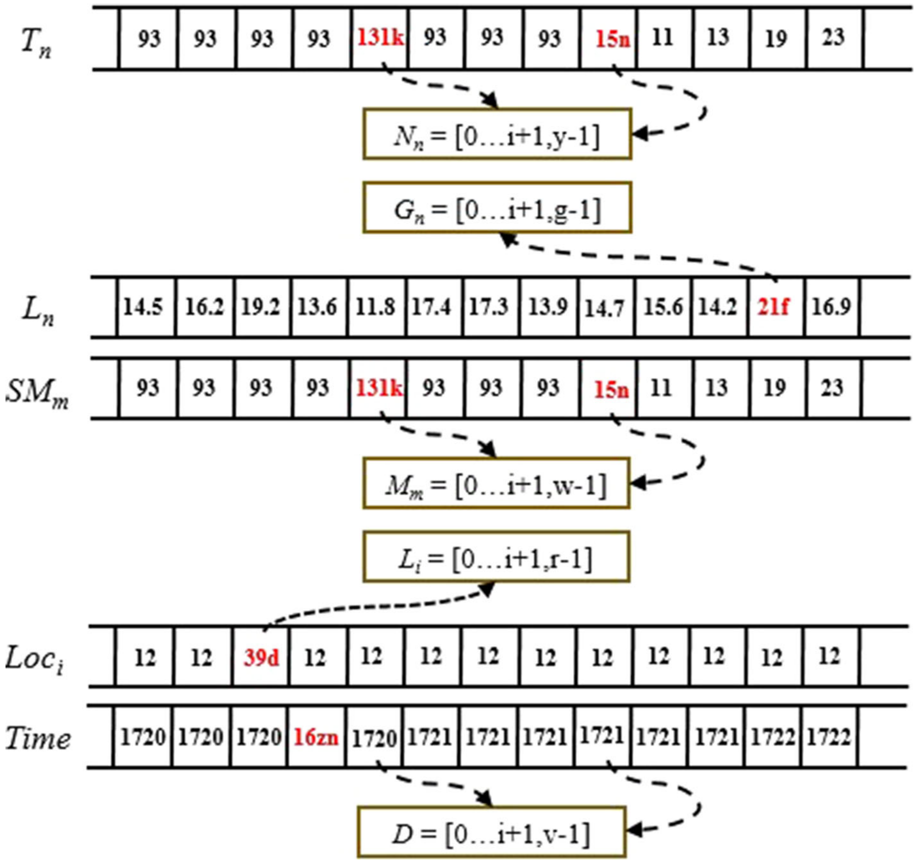

Fig. 2 In-use data block scanning through modified TSPLFC

$$ESB_{Buffer_i} = \sum_{R_i \le B_i}^{Columns\epsilon String(i)} (N_n, G_n, M_m, L_i, D_i) \quad (2)$$

DFS uses socket to fetch single row of ESB_{Buffer_i} through index $Index_i$ method and applies *textoutputformat* [44] that writes entries into a text file ESB_{File_i} as shown in Fig. 3.

3.2 HBase Error Analyzer

The proposed approach (SMSFA) uses semantic tuple repository Hbase for storing file content of text file ESB_{File_i}. Hbase is a project of Apache Foundation that works in coordination with Hadoop ecosystem [45] and performs statistical analytics over obtained error dataset. SMSFA collects the error data chunk into a table $SMError_{Table}$ of Hbase repository and helps to performs error analytics over fetched data as shown in Fig. 4.

SM ID	Location	Load	Temp	Time

Fig. 3 Error sample Bundler dataset format

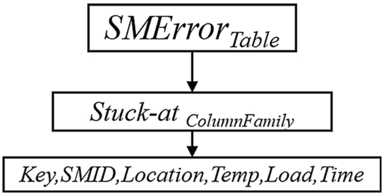

Fig. 4 HBase smart meter error repository

3.3 Stuck-at Trainer and Fault Predictor

SMSFA predicts stuck-at fault in two phases i.e. (1) Training of $SMError_{Table}$ dataset and (2) Stuck-at fault prediction.

3.3.1 *Training of $SMError_{Table}$ Dataset*

The training phase performs a learning operation over 5 column datasets and observes columns' values for the possible stuck-at victim. For this purpose, we use Hidden Markov Model (HMM) [46] 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 such as hidden states $Z = \{z_1, z_2\}$, transition probability over a condition $A = a_{ij} = \left\{P \mid q_{(t+1)} = x_j \mid q_t = x_j\right\}$, 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.

The training $SMError_{Table}$ dataset can be observed through,

$$\lambda = (\pi, A, B) \tag{3}$$

Where *pi* 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 in-use block status with time t_{SD} as shown in Fig. 6.

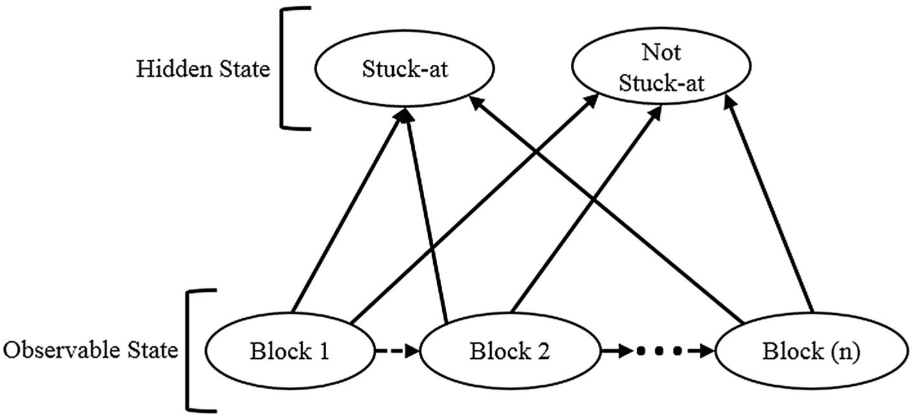

Fig. 5 HMM observable and hidden states

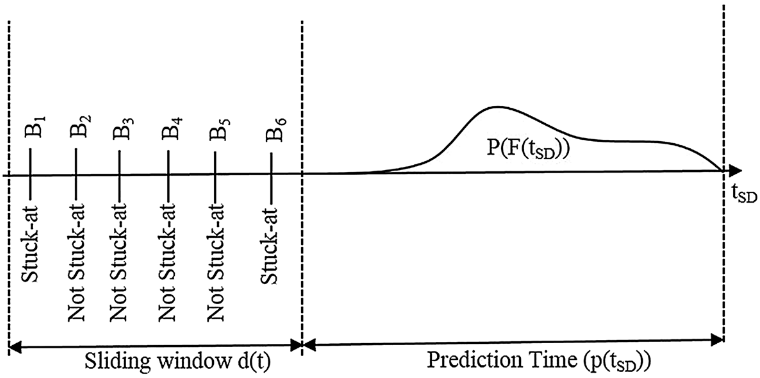

Fig. 6 Block stuck-at fault analysis

Algorithm 1 Stuck-at fault predictor using $SMError_{Table}$

Initialize $O = \{B_1, B_2, B_3, B_4, B_5, B_6\}$
$S = \{Stuck - at, notStuck - at\}$
$M = 2$
$N = 6$
Initialize $Transprobability\,(A_{ij})\,, Emissionprobability\,(B_{ij})$
$Y = \{Y_1, Y_2, Y_3, Y_4\}$
Initialize $Probability\,(StatePath)$
Update B_{ij}, A_{ij}
procedure
    $Probability\,(Path) = B_{ij} \times Probability\,(StatePath)$
    for Each state S_j **do**
        $Probability\,(StatePath)\,[j, i] \leftarrow max_k\,\left(Probability\,(StatePath)\,[k, i-1] * A_{kj} * B_{jyi}\right)$
        $Probability\,(Path)\,[j, i] \leftarrow argmax_k\,\left(Probability\,(Path)\,[k, i-1] * A_{kj} * B_{jyi}\right)$
    $C_i \leftarrow argmax_k\,(Probability\,(StatePath))$
    $Z_i \leftarrow S_i$
    for $I \leftarrow T-1, ..., 1$ **do**
        $C_i \leftarrow Probability\,(Path)\,[Z_i, i]$
        $Z_i \leftarrow C_i$
    Produce $(Time, Z)$

In this way, we process $SMError_{Table}$ dataset and train HMM model through Expectation-Maximization (EM) algorithm [47], which finds likelihood elements in maximizing parameters and similarities until best fit result appears for $SMError_{Table}$ dataset. The training phase consists of two steps such as, initially, we calculate observation sequence probability (O_t, O_{t+1}, O_{t+2}) and applies *EM* algorithm in two steps. The first step calculates expected likelihood from current estimation followed by the second step that calculates parameters of maximizing expectation for stuck-at status likelihood as observed in Algorithm-1.

3.3.2 Stuck-at Fault Prediction

SMSFA uses Viterbi algorithm [48] that analyzes stuck-at fault and calculates hidden states. At first, the algorithm shows optimal state sequence and declares hidden states of the model 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,

$$S_{Optimal} = arg \times max_s \times (S_{States}; O; \lambda) \tag{4}$$

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 such as $stuck-at$ or $not\ stuck-at$ from collected observations O of in-use block bundle as shown in Fig. 7.

4 Experimental Evaluation

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

The cluster includes Intel Xeon with 8 CPUs, storage media i.e. Hard disk drive 1TB, 128 GB Samsung SSD and 32 GB memory. Similarly, we also use Intel core i5 with 4 Core, storage media i.e. Hard disk drive 1TB, 128 GB Samsung SSD and 16 GB 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.

The dataset used to process experimental work includes 100 smart meter block bundles of one month (2 GB dataset) [49–53].

4.1 Experimental Results

The experiments conducted to evaluate our scheme are (1) In-use block scanner (2) Stuck-at block training and (3) Stuck-at block prediction.

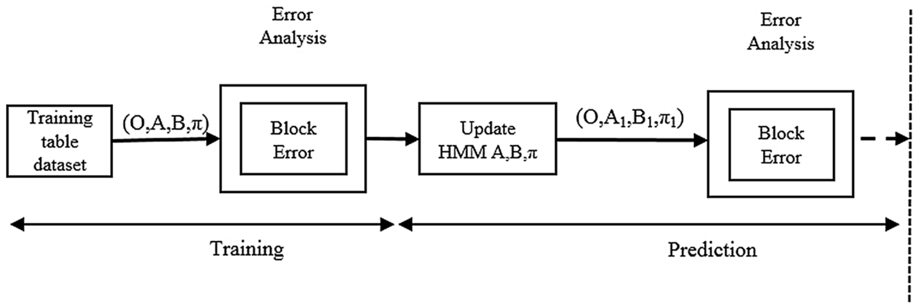

Fig. 7 In-use data block stuck-at error prediction process

Table 1 Cluster configuration

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

Table 2 Virtual machine configuration over Hadoop cluster

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

4.1.1 In-Use Block Scanner

To evaluate the efficiency of block scanner, we perform scan analysis over dataset columns such as 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.

4.1.2 Stuck-at Block Training

The training phase includes processing of Expectation-Maximization (EM) algorithm over $SMError_{Table}$ in the Hbase repository. At first, this algorithm buffers observations % percentile of all bundle columns and perform state probability through maximization phase that matches the neighborhood range against original error bundle. In this way, we observe the 98.2% identical error sampling of bundles close to real ones that were collected through scanning.

4.1.3 Stuck-at Block 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

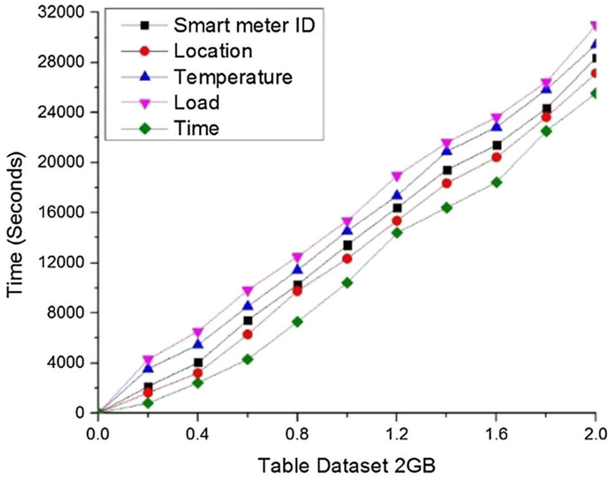

Fig. 8 Block scanner in table dataset

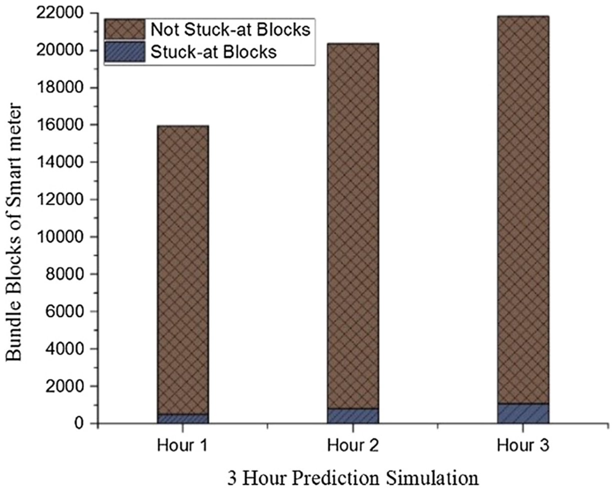

Fig. 9 Stuck-at fault prediction statistics

hours and evaluated that first hour prediction declares '528' bundle blocks with a status of "stuck-at" out of '15,450' bundle blocks. Furthermore, the second hour of simulation predicts '831' bundle blocks with a status of "stuck-at" out '19,530' bundle blocks. Moreover, the third hour of simulation predicts '1077' bundle blocks with a status of "stuck-at" out of '20,750' bundle blocks as shown in Fig. 9.

5 Conclusion

This paper proposes a novel knowledge-based stuck-at fault analyzer that scans in-use data bundle for abnormal patterns of sensory column values. The proposed approach stores scanned error dataset into HBase repository and performs prediction analysis for training and evaluating in-use stuck-at data blocks in smart meters. The experimental evaluation shows that SMSFA effectively identifies stuck-at fault bundles in large sensory dataset of smart meters and resolves the issue of inaccurate data analytics due to stuck-at smart meter data block in the smart grid. In future works, we will focus on sentinel data analytics using in-use smart meter data blocks into smart grid.

Acknowledgements This work was supported by the National Research Foundation of Korea through the Korean government (MSIP) under Grant NRF-2016R1C1B2008624.

References

1. Tuballa, M. L., & Abundo, M. L. (2016). A review of the development of smart grid technologies. *Renewable and Sustainable Energy Reviews*, *59*, 710–725.
2. Bera, S., Misra, S., & Rodrigues, J. J. P. C. (2015). Cloud computing applications for smart grid: A survey. *IEEE Transactions on Parallel and Distributed Systems*, *26*(5), 1477–1494.
3. Stojkoska, B. L. R., & Trivodaliev, K. V. (2016). A review of Internet of things for smart home: Challenges and solutions. *Journal of Cleaner Production*.
4. Chren, S., Rossi, B., & Pitner, T. (2016). Smart grids deployments within EU projects: The role of smart meters. In *2016 Smart cities symposium Prague (SCSP)*.

5. G. KG, Toshiba Smart meter MCUs, Glyn.de, 2017. [Online]. Available: http://www.glyn.de.Lastaccessed. 27 April 2017.
6. He, J. et al. (2017). The unwritten contract of solid state drives. In *Proceedings of the twelfth European conference on computer systems*. ACM.
7. Compagnoni, Christian Monzio. et al. (2017). Reviewing the evolution of the NAND Flash technology. In *Proceedings of the IEEE*.
8. Chaudhry, A. A., Kui, C., & Guan, Y. L. (2017). Mitigating stuck cell failures in MLC NAND flash memory via inferred erasure decoding. *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*.
9. Gungor, V. C., et al. (2011). Smart grid technologies: Communication technologies and standards. *IEEE Transactions on Industrial Informatics*, *7*(4), 529–539.
10. Gungor, V. C., et al. (2013). A survey on smart grid potential applications and communication requirements. *IEEE Transactions on Industrial Informatics*, *9*(1), 28–42.
11. Alahakoon, D., & Yu, X. (2016). Smart electricity meter data intelligence for future energy systems: A survey. *IEEE Transactions on Industrial Informatics*, *12*(1), 425–436.
12. King, J., & Perry, C. (2017). Smart buildings: Using smart technology to save energy in existing buildings.
13. Williams, T. W., & Brown, N. C. (1981). Defect level as a function of fault coverage. *IEEE Transactions on Computers*, *30*(12), 987–988.
14. Millman, S. D., McCluskey, E. J., Acken, J. M. (1990). Diagnosing CMOS bridging faults with stuck-at fault dictionaries. In *Test conference, 1990. Proceedings, International. IEEE*.
15. Dekker, R., Beenker, F., & Thijssen, L. (1990). A realistic fault model and test algorithms for static random access memories. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *9*(6), 567–572.
16. McCluskey, E. J., Tseng, C.-W. (2000). Stuck-fault tests vs. actual defects. Test conference. *Proceedings. International* (p. 2000). IEEE.
17. Lima, F., Carro, L., & Reis, R. (2003). Designing fault tolerant systems into SRAM-based FPGAs. In *Proceedings of the 40th annual design automation conference*. ACM.
18. Van De Goor, A. J.., & Al-Ars, Z. (2000) Functional memory faults: A formal notation and a taxonomy. In *VLSI test symposium, 2000. Proceedings. 18th IEEE. IEEE*.
19. Sachdev, M., & Verstraelen, M. (1993). Development of a fault model and test algorithms for embedded DRAMs. In *Test conference, 1993. Proceedings., International. IEEE*.
20. Wiscombe, P. C. (1993). A comparison of stuck-at fault coverage and I/sub DDQ/testing on defect levels. In *Test conference, 1993. Proceedings, International*. IEEE.
21. Nagvajara, P., & Karpovsky, M. G. (1991). Built-in self-diagnostic read-only-memories. In *Test conference, 1991, Proceedings, international*. IEEE.
22. Fan, X., et al. (2005). A novel stuck-at based method for transistor stuck-open fault diagnosis. In *Test conference, 2005. Proceedings. ITC 2005. IEEE International*. IEEE.
23. Mikitjuk, V. G., V. N. Yarmolik, Van De Goor, A. J. (1996). Ram testing algorithms for detection multiple linked faults. In *European design and test conference, 1996. ED&TC 96. Proceedings*. IEEE.
24. Soden, J. M., et al. (1992). IDDQ testing: A review. *Journal of Electronic Testing*, *3*(4), 291–303.
25. Gai, S., Mezzalama, M., & Prinetto, P. (1983). A review of fault models for LSI/VLSI devices. *Software & Microsystems*, *2*(2), 44–53.
26. Kim, H. et al. (2001). Design of dual-duplex system and evaluation of RAM. In *Intelligent transportation systems, 2001. Proceedings. 2001 IEEE*. IEEE.
27. Corsi, A., & Morandi, C. (1983). A review of RAM testing methodologies. *Microelectronics Journal*, *14*(2), 55–71.
28. Kuo, T.-W., et al. (2011). An efficient fault detection algorithm for NAND flash memory. *ACM SIGAPP Applied Computing Review*, *11*(2), 8–16.
29. Chaudhry, A. A., Kui, C., & Guan Y. L.. (2017). Mitigating stuck cell failures in MLC NAND flash memory via inferred erasure decoding. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems.*
30. Cooke, J. (2007). The inconvenient truths of NAND flash memory. Flash Memory Summit.
31. Kgil, T., Roberts, D., Mudge, T. (2008) Improving NAND flash based disk caches. In *Computer Architecture, 2008. ISCA'08. 35th International Symposium on. IEEE*.
32. Grupp, L. M., Davis, J. D., Swanson, S. (2012). The bleak future of NAND flash memory. In *Proceedings of the 10th USENIX conference on file and storage technologies*. USENIX Association.
33. Kgil, T., & Mudge, T. (2006). FlashCache: A NAND flash memory file cache for low power web servers. In *Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems*. ACM.
34. Jimenez, X., Novo, D., Ienne, P. (2014). Wear unleveling: Improving NAND flash lifetime by balancing page endurance. FAST. Vol. 14..
35. Bez, R., et al. (2003). Introduction to flash memory. *Proceedings of the IEEE*, *91*(4), 489–502.

36. Lee, S. et al. (2009). FlexFS: A flexible flash file system for MLC NAND flash memory. In *USENIX annual technical conference*.
37. Desnoyers, P. (2010). Empirical evaluation of NAND flash memory performance. *ACM SIGOPS Operating Systems Review*, *44*(1), 50–54.
38. Bez, R., & Pirovano, A. (2004). Non-volatile memory technologies: Emerging concepts and new materials. *Materials Science in Semiconductor Processing*, *7*(4), 349–355.
39. Cho, S., & Lee, H. (2009). Flip-N-write: A simple deterministic technique to improve PRAM write performance, energy and endurance. In *Microarchitecture, 2009. MICRO-42. 42nd annual IEEE/ACM international symposium on. IEEE*.
40. Mohan, V., et al. (2010). How I learned to stop worrying and love flash endurance. *HotStorage*, *10*, 3–3.
41. Park, M., et al. (2007). The effect of trapped charge distributions on data retention characteristics of NAND flash memory cells. *IEEE Electron Device Letters*, *28*(8), 750–752.
42. Kim, W., et al. (2009). Multi-layered vertical gate NAND flash overcoming stacking limit for terabit density storage. In *VLSI Technology, 2009 Symposium on. IEEE*.
43. Zubair, M., Wahab, F., Hussain, I., Zaffar, J. (2010). Improved text scanning approach for exact string matching. In *Proceedings of international conference on information and emerging technologies*.
44. 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 27 Apr 2017.
45. "Welcome to Apache Hadoop," 2014. [Online]. Available: http://hadoop.apache.org/. Last accessed 27 Apr 2017.
46. Ghahramani, Z. (2001). An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, *15*(1), 9–42.
47. Ajit Singh, EM Algorithm, 2005.
48. Forney, G. D. (1973). The Viterbi algorithm. *Proceedings of the IEEE*, *61*(3), 268–278.
49. 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: 27 Apr 2017.
50. Musaddiq, A., Zikria, Y. B., Hahm, O., Yu, H., Bashir, A. K., & Kim, S. W. (2018). A survey on resource management in IoT operating systems. *IEEE Access*, *6*, 8459–8482.
51. Qureshi, N. M. F., Shin, D. R., Siddiqui, I. F., & Chowdhry, B. S. (2017). Storage-tag-aware scheduler for hadoop cluster. *IEEE Access*, *5*, 13742–13755.
52. Qureshi, N. M. F. & Shin, D. R. (2016). RDP: A storage-tier-aware robust data placement strategy for hadoop in a cloud-based heterogeneous environment. *KSII Transactions on Internet and Information Systems*, *10*(9), 4063–4086.
53. Siddiqui, I. F., Lee, S. U. J., Abbas, A., & Bashir, A. K. (2017). Optimizing lifespan and energy consumption by smart meters in green-cloud-based smart grids. *IEEE Access*, *5*, 20934–20945.

Publisher's Note

Isma Farah Siddiqui received the B.E. degree in software engineering and the M.E. degree in information technology from the Mehran University of Engineering and Technology, Pakistan, in 2006 and 2008, respectively, and the Ph.D. degree from Hanyang University ERICA Campus, South Korea, in 2018, funded by the Higher Education Commission, Pakistan. She received best PhD thesis and researcher award by graduate school of engineering, Hanyang University, For her outstanding research contributions. Since 2006, she has been with the Department of Software Engineering, Mehran UET, where she is currently serving as an Assistant Professor. Her research interests include machine learning and data analytics in areas of smart environment, semantic Web, IoT, green cloud, and big data.

Nawab Muhammad Faseeh Qureshi received Bachelors of Engineering(B.E.) in Software Engineering and Masters of Engineering(M.E.) in Information Technology (I.T.) from Mehran University of Engineering and Technology, Pakistan, respectively. He was awarded with SAMSUNG Scholarship for Ph.D. studies and received his Ph.D. degree in Computer Engineering from Sungkyunkwan University, South Korea. He received Superior Research Award from College of Information and Communication Engineering on account of his research contributions and performance during Ph.D. studies. Currently, he is Assistant Professor in Sungkyunkwan University, South Korea. He is actively involved in working on novel Big Data applications and discusses new trends in large-scale data analytics. He has been an active TPC member of various international conferences. He is a reviewer for renowned Journals such as IEEE Transactions on Industrial Informatics, IEEE Communications Magazine, Future Generation Computer Systems, IEEE Access, Wireless Personal Communication. His research interests include Big Data analytics, context-aware data processing of Internet of Things and Cloud computing.

Muhammad Akram Shaikh is working as Professor in the Department of Software Engineering, and remained Director General in Pakistan Scientific and Technology Information Centre (PASTIC), a subsidiary of Pakistan Science Foundation under Ministry of Science and Technology and Co-Director Institute of Information and Communication Technologies, Mehran University of Engineering and Technology Jamshoro. He has received his B.E. from Mehran University (Pakistan) in 1993, MBA from University of Sindh in 1996, M.Sc. from University of Huddersfield (UK) in 2001, and Ph.D. from Tsinghua University (China) in 2008. He is the life time member of Pakistan Engineering Council. He started his professional career in 1993. He has 23 years of teaching, research and management/admin experience. He is author of more than 25 journal/conference papers of national/international repute. In addition he is also attached as editor/co-editor/reviewer of national/international journals, Session chair/PC member of national/international conferences, member accreditation committee PEC, member national curriculum review committees of HEC, member technical review committee of PSF, member executive committee of PSF, member Board of Trustee (BoT) PSF, member HEC digital library advisory board, and member board of studies of various universities of Pakistan. His research areas of interest include Knowledge Engineering, Scientific and Technological Databases, Information Processing, Data Mining and Data Warehousing, Software Engineering, Automation and Control, Networks, Virtual Reality and Graphics.

Bhawani Shankar Chowdhry is full Professor and the Dean Faculty of Electrical Electronics and Computer Engineering at Mehran University of Engineering and Technology, Jamshoro, Pakistan. He did his Ph.D. from renowned School of Electronics and Computer Science, University of Southampton, UK in 1990. He is having teaching and research experience of more than 30 years. He has the honor of being one of the editor of several books Wireless Networks, Information Processing and Systems, CCIS 20, Emerging Trends and Applications in Information Communication Technologies, CCIS 281, Wireless Sensor Networks for Developing Countries, CCIS 366, Communication Technologies, Information Security and Sustainable Development, CCIS 414, published by Springer Verlag, Germany. He has also been serving as a Guest Editor for Wireless Personal Communications which is Springer International Journal. He has produced more than 13 Ph.Ds. and supervised more than 50 M.Phil./Masters Thesis in the area of ICT. His list of research publication crosses to over 60 in national and international journals, IEEE and ACM proceedings. Also, he has Chaired Technical Sessions in USA, UK, China, UAE, Italy, Sweden, Finland, Switzerland, Pakistan, Denmark, and Belgium. He is member of various professional bodies including: Chairman IEEE Karachi Section, Region10 Asia/Pacific, Fellow IEP, Fellow IEEEP, Senior Member, IEEE Inc. (USA), SM ACM Inc. (USA). He is lead person at MUET of several EU funded Erasmus Mundus Program including Mobility for Life, StrongTies, INTACT, and LEADERS. He has organized several International Conferences including IMTIC08, IMTIC12, IMTIC13, IMTIC15, WSN4DC13, IEEE SCONEST, IEEE PSGWC13, and track chair in Global Wireless Summit (GWS 2014), and Chief Organizer GCWOC 16 and GCWOC 17 in Malaga Spain.

Asad Abbas received his B.S (Information Technology) degree from University of the Punjab Pakistan in 2011. Currently he is student of MS-leading to PhD program in Department of Computer Science and Engineering in Hanyang University ERICA campus at Ansan, South Korea, funded by Higher Education Commission of Pakistan in 2014. His research interests include Software Product Line, Software Requirement Traceability and IoT Applications.

Ali Kashif Bashir received his Ph.D. in Computer Science and Engineering from Korea University, South Korea. Currently he is Associate Professor in Department of Science and technology, University of the Faroe Islands, Faroe Islands, Denmark. His previous appointments includes Graduate School of Information Science and Technology, Osaka University, Japan; Nara National College of Technology, Japan; National Fusion Research Institute, South Korea; Southern Power Co. Ltd., South Korea; and Seoul Metropolitan Government, South Korea. He is a senior member of IEEE (SMIEEE) and active member of IEICE and IEEE Experts in Technology and Policy (ETAP). He is serving as Editor in chief on IEEE Internet Technology Policy Newsletter and IEEE Future Directions Newsletter. He is editorial board member of journals like IEEE Access, Journal of Sensor Networks and Data Communications, etc. He has chaired several conference sessions, gave several invited and keynote talks, and reviewed the technology leading articles for journals like IEEE Transactions on Industrial Informatics, IEEE Communication Magazine, IEEE Communication Letters, IEICE Journals, and conferences like IEEE Infocom, IEEE ICC, IEEE Globecom, IEEE Cloud of Things, etc. His research interests include: cloud computing, NFV/SDN, network virtualization, IoT, computer networks, RFID, Sensor Networks, wireless networks, distributed computing, etc.

Scott Uk-Jin Lee received his B.E degree in software engineering and Ph.D. degree in computer science from the University of Auckland, New Zealand. After the doctoral degree, he worked as post-doctoral research fellow at Commissariat l'nergie atomique et aux nergies alternatives (CEA), France. Dr. Lee has been with the Department of Computer Science and Engineering, Hanyang University, South Korea since 2011. He is serving as the head of Department of Software for Emerging Technology major. He has served as editor, technical chair and committee member for several journals and conferences. He is also a member of the Korean Institute of Information Scientists and Engineers and the Korean Society of Computer and Information. His research interests include Software Engineering, Formal Methods, Web, Software Product Line, Software Requirement Engineering, Software Quality Management, Semi-structured Data, Semantic Web, Web technologies and Security, Software Security and Internet of Things (IoT).