A HIDDEN MARKOV MODEL TO PREDICT HOT SOCKET ISSUE IN SMART GRID

1ISMA FARAH SIDDIQUI, 2ASAD ABBAS, 3SCOTT UK-JIN LEE
1,2,3 Department of Computer Science and Engineering, Hanyang University ERICA, Ansan, South Korea

*Corresponding Author
E-mail: 1isma2012@hanyang.ac.kr, 2asadabbas@hanyang.ac.kr, 3scottlee@hanyang.ac.kr

ABSTRACT

Smart meters collect sensor data at distribution ends of smart grid. The collection process performs non-stop data bundling and results in ‘hot socket’ issue due to high resistance. This results an abnormal generation of dataset and overall severely affect the operational aspects of smart grid. In this paper, we present a model for Smart Meter Abnormal Data Identification (SMADI) over the communication bridge of Smart grid repository and distribution end units, to redirect abnormal samples to HBase error repository using Message propagation strategy. SMADI predicts possible hot socket smart meter node through HMM and generates a sequence of possible hot socket smart meters over time interval. The simulation results show that SMADI precisely collect error samples and reduce complexity of performing data analytics over giant data repository of a smart grid. Our model predicts hot socket smart meter nodes efficiently and prevent computation cost of performing error analytics over smart grid repository.

Keywords: IoT, Smart meter, Smart grid, HBase, Hot Socket.

1. INTRODUCTION

The concept of smart grid has enabled many state-of-the-art ideas to prevail in the technological world. It is a giant automated paradigm that enables to deliver efficient electrical operations around a metropolitan hub [1]. The smart grid consists of power system, giant storage, intelligent agents, electric transportations, interoperability, integration of renewables, distributed controls, information technology, efficiency and reliability of system, communication technology, IoT devices and infrastructure security [2] as seen from Figure-1.

Recently, IoT devices have replaced traditional devices in the smart grid [3]. Smart meters have become an efficient sensor product to analyze and collect distributed units processing information [4]. Recently, smart grid has enhanced communication and processing protocols along with storage repository technology [5]. The dataset of smart meters is collected over a mutual IoT data container [6] and transform from smart meter format i.e. ‘CSV’ to smart grid repository format ‘RDF’ [7]. The transformed dataset is then forwarded to the transmission channel of smart grid [8]. The transmission channel compiles the RDF dataset to smart meter data collector unit [9] and...
stored over smart grid repository [10]. The smart grid core performs multiple type of analytics over stored dataset [11] and uses repository to perform functional processing of smart grid [12] as seen from Figure-2.

Smart meters process sensor datasets round the clock [13] and results in decrease of lifespan [14]. As a result, smart meters are affected with ‘hot socket’ problem [15] and generate abnormal temperature and load point dataset. The researcher presented two models to identify temperature and load point faults in previous study. However, related work could not perform abnormality analytics over temperature and load point dataset and prediction of exact status of smart meter through functional dataset. Moreover, we know that, ‘hot socket’ problem is a hardware based issue [16] and smart meter become useless after having it. Therefore, we present an error analytical solution to the said problem.

We propose a software solution to avoid above mentioned ‘hot socket’ problem. The proposed SMADI scheme performs scan of abnormal temperature and load points of smart meter and fetch them to the HBase repository. After that, SMErro r module fetch error dataset for training the model and predict fault analysis before a smart meter reached over dead end.

The main contributions of the proposed scheme are:

- An enhanced smart meter temperature and load point scanner.
- An efficient data points bundler.
- A column family based storage of error data in the HBase.
- A novel SMEr ro r module over message propagation model.
- An efficient data trainer of SMErro r module.
- A precise prediction model for possible protection of ‘Hot Socket’ problem.

The remaining paper is organized as follows. Section II briefly explains proposed approach SMADI. Section III explains experimental environment and SMADI scheme results. Finally, section IV shows conclusion with significant contributions and future research directions.

2. SMART METER ABNORMAL DATA IDENTIFICATION (SMADI)

The proposed model is categorized in two phases i.e. (i) Error data analytics and (ii) Hot socket prediction. When a smart meter generates a dataset of CSV file format [17], it is transformed into RDF format and stored over Smart meter data collector (SMDC). SMADI collect the data sample and perform scanning to collect error samples in dataset. The resultant collection is bundled in the form of Hadoop Archive (HAR) [18] and stored over Hbase [19] repository. Furthermore, SMADI train HBase dataset and predict hot socket smart meters as seen from Figure-3.

2.1 Error Data Analytics

Error data analytics (EDA) is categorized into three phases i.e. (a) Sample scanner, (ii) Sample Bundler and (iii) HBase storage.

2.1.1 Sample Scanner

The Sample scanner fetches abnormal element from dataset sample having set of load points \( L_p \) and temperature points \( T_p \). The collection of smart meter data elements can be represented as,

\[
Data_{Set} = (T_p, L_p)
\]

A modified text string algorithm TSPLFC [20] is used to scan abnormal elements over \( Data_{Set} \). There are two changes in current algorithm i.e. (i) Scanning large buffer of data at a time and (ii) Simultaneous identification of dataset elements \( L_p \) and \( T_p \).

Suppose we process \( Data_{Set} = [0...m-1,0...n-1] \) over an abnormal pattern \( A_p = [0...s-1] \). The modified TSPLFC algorithm compare \( T_p \) and \( L_p \) over pattern of \( A_p \) and fetch result set in array of \( F_p \). The scanning process continues till \( L_p[0] \) and \( L_p[m-1] \) do not find \( A_p[i+1...s-1] \) and terminate the algorithm execution by returning an index value. Similarly, \( T_p[0] \) and \( T_p[n-1] \) do not find \( A_p[i+1...s-1] \) and generate an array of scanned elements \( S_p \) to
bundle resultant sample for bundle processing as seen from Figure-4.

![Figure 4: Modified TSPLFC Based Sample Scanning](image)

### 2.1.2 Sample Bundler

The sample bundler (SB) is a paradigm to write $A_p$ dataset over a DFS file. At first, SB collect $A_p$ result set to $SB\_Buffer$. A single sample of buffer collection can be represented as,

$$SB\_Buffer = \sum (Smart\_Meter, A_p(T_p), A_p(T_p))$$

The DFS write socket fetch $SB\_Buffer$ text over a single index $SB\_Index$ and $textoutputformat$ [21] write entries over file format $SB\_File$ as seen from Figure-5.

<table>
<thead>
<tr>
<th>SM ID</th>
<th>Location</th>
<th>$A_p(T_p)$</th>
<th>$A_p(T_p)$</th>
<th>Time</th>
</tr>
</thead>
</table>

![Figure 5: Sample Bundler data format](image)

### 2.1.3 Sample HBase Storage

The Sample HBase Storage (SHS) is a Hadoop [22] repository to store $SB\_File$ for error analytics. For this purpose, we created a table $SMError\_table$ over $SB\_File$ format. The $SMError\_table$ configuration can be observed through Figure-6.

![Figure 6: Hbase Smart Meter Error Repository](image)

### 2.2 Hot socket prediction

Hot socket prediction is categorized into three phases i.e. (i) SMError Module, (ii) Data trainer and (iii) HotSocket Prediction.

#### 2.2.1 SMError Module

SMError Module (SMM) is a paradigm to collect $SMError\_table$ data conditionally for hot socket prediction. SMM is derived through Belief propagation method [23] that helps to fetch conditional data elements from HBase repository. In order to perform inference on belief propagation, we choose Message propagation model [24] that fetch a message $m$ of variable container $i$ having value $\epsilon_i$ with a belief $b_i(\epsilon_i)$ and propagate from source container $a$ to destination container $i$ and presents likeliness of random variable $X_i$ where $\epsilon_i \in X_i$ by [25],

$$message_{m_a \rightarrow i}(\epsilon_i)$$

The information message of Key ($k$), SMID ($S$), Location ($L$), $A_p(T_p)$, $Tm_i$ and time ($Tm$) to $Temperature_{ColumnFamily}$ container, belief of $T$ container can be represented as,

$$b_i(\epsilon_{T}) \propto \prod m(k_i S_i L_i(T_p)_{Tm_i}) \in N(Temp)$$

We normalize eq (9) with constant $z$ and the belief of $T_i$ can be represented as,

$$b_i(\epsilon_{T_i}) = \frac{1}{Z} \prod m(k_i S_i L_i(T_p)_{Tm_i})$$

Similarly, $Load_{ColumnFamily}(L)$ belief with Key ($k$), SMID ($S$), Location ($L$), $A_p L_p$ ($L_p$) and time ($Tm$) can be obtained,

$$b_i(\epsilon_{L_i}) = \frac{1}{Z} \prod m(k_i S_i L_i L_p(T_p)_{Tm_i})$$

The belief of $T_i$ and $L_i$ can be observed from Figure-7(a) and Figure-7(b)
In order to forward container messages \( T_i \) and \( L_i \) to SMErr\( \text{or} \) module, we calculate joint belief of \( T_i \) and \( L_i \). The joint belief is represented as a logical container \( L \) and can be represented as,

\[
b_l(\chi_L) = b_A(\chi_A)
\]

(12)

where \( \chi_A = \{ \chi_{T_i}, \chi_{L_i}, T_i, L_i \in N(A) \} \) and \( \chi_L \) is the domain space linked with logical container \( L \) as seen from Figure-8.

The domain space \( \chi_L \) equals to \( \{ (\chi_{T_i}, \chi_{L_i}) \mid f_A(\chi_{T_i}, \chi_{L_i}) = 1, \chi_{T_i} \in \chi_{T_i}, \chi_{L_i} \in \chi_{L_i} \} \) and factor \( f_A \) is bipartite string between joint containers. The joint belief of container \( L \) can be obtained as,

\[
b_L(\chi_L) = \frac{1}{Z} \prod_{(T_i,L_i) \in N(L)} m_{(T_i,L_i)}(\chi_L)
\]

(13)

After applying factor \( f_A \) to SMErr\( \text{or} \) module, we receive a close form solution as,

\[
m_{(T_i,L_i)}(\chi_L) = \sum_{(T_i,L_i) \in N(L)} f_A(\chi_A) \prod_{(T_i,L_i) \in N(L)} m_{(T_i,L_i)}(\chi_L)
\]

(14)

Where \( m_{(T_i,L_i)}(\chi_L) \) represents collection of temperature and load messages in SMErr\( \text{or} \) module equivalent to factor \( F_A \) filtered messages over \( \text{Load} \) and \( \text{Temperature} \) ColumnFamily. Moreover, Figure-9 shows the collection process of smart meter temperature and load points into SMErr\( \text{or} \) Module.

2.2.2 Data Trainer

The Data trainer (DT) train temperature and load points present in SMErr\( \text{or} \) module. After that it predict whether a smart meter is tending toward hot socket issue or at a stable position. For this reason, we use Hidden Markov Model (HMM) [26] that provide a self-learning mechanism to train dataset. Therefore, we insert information related to ‘Faulty’ and ‘Not Faulty’ from SMErr\( \text{or} \) module. After that, we create a transition matrix that contain the probability of hidden states. By default, HMM have hidden states as \( Z = \{ z_1, z_2 \} \), transition probability at a condition \( A = a_{ij} = \{ \mathcal{P}(q_{t+1} = q_{t} = x_i) \} \), observation state \( X = \{ x_1, x_2, x_3, x_4 \} \) and emission probability \( B = b_{ij} \). After, we apply HMM to our DT, we observe \( O = \{ \text{SM}_1, \text{SM}_2, \text{SM}_3, \text{SM}_4, \text{SM}_5 \} \) over hidden states i.e. Faulty and Not Faulty as seen from figure-10.

According to the definition, we get,

\[
\lambda = (\pi, A, B)
\]

(15)

where \( B \) is matrix of emission that produce \( b_{ij}(Y_i) \), \( \pi \) is initial state probability of transition matrix and \( A \) is the matrix of transition which have probability of transitioning from one state to another. After having \( O \) and hidden states, we calculate prediction cycle sliding window of time \( \Delta t \) and observe that smart meters depict status and time \( t_{SD} \) as seen from figure-11.
Finally, DT model is trained with smart meter load and temperature points through SMEError module. We train model through Expectation-Maximization (EM) algorithm [27] that find likelihood elements over maximizing parameters with maximum similarities until best fit appear to DT model.

At first, we calculate observable sequence probability \( O_t, O_{t+1}, O_{t+2} \) and uses EM algorithm in two steps. The first step calculates expected likelihood from current estimation and second step calculate parameters maximizing expected fault status likelihood as observed from Algorithm-1.

\[
\begin{align*}
S_{optimal} &= \arg\max_{S_{states}} P(S_{states}; O; \lambda) \quad (16)
\end{align*}
\]

where \( S_{optimal} \) reflects the optimal state sequence. Viterbi algorithm allow \( S_{optimal} \) to retrieve possible optimal paths at each step \( t \) that end at \( n \) states. At \( t+1 \), \( S \) hold to increase and optimal path for \( n \) is updated. At \( t+2 \), \( S \) comes at maxima job-likelihood and optimal path for \( n \) is updated and predicts the hidden state i.e. ‘faulty’ or ‘not faulty’ from collected observations \( O \) of smart meters as seen from Figure-12.

**3. EXPERIMENTAL EVALUATION**

In this section, we evaluate our proposed model over a configuration of cluster as seen from Table-1.

**Table 1: Cluster Configuration.**

<table>
<thead>
<tr>
<th>Machine</th>
<th>Specifications</th>
<th>No. of VM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Xeon E5-2600 v2</td>
<td>8 CPUs, 32GB memory, 1 TB HDD &amp; SSD</td>
<td>3</td>
</tr>
<tr>
<td>Intel core i5</td>
<td>4 Core, 16GB memory, 1 TB HDD &amp; SSD</td>
<td>2</td>
</tr>
<tr>
<td>Hadoop</td>
<td>Hadoop-2.7.2 (table)</td>
<td>2</td>
</tr>
<tr>
<td>Virtual Machine Management</td>
<td>Virtualbox 5.0.16</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2: Virtual Machine Configuration Over Hadoop Cluster.**

<table>
<thead>
<tr>
<th>Node</th>
<th>CPU</th>
<th>Memory</th>
<th>Disk</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Node</td>
<td>6</td>
<td>16 GB</td>
<td>HDD &amp; SSD</td>
<td>Intel Xeon</td>
</tr>
<tr>
<td>Slave1</td>
<td>2</td>
<td>4 GB</td>
<td>HDD &amp; SSD</td>
<td>Intel Xeon</td>
</tr>
<tr>
<td>Slave2</td>
<td>2</td>
<td>4 GB</td>
<td>HDD &amp; SSD</td>
<td>Intel Core i5</td>
</tr>
<tr>
<td>Slave3</td>
<td>2</td>
<td>4 GB</td>
<td>HDD &amp; SSD</td>
<td>Intel Core i5</td>
</tr>
<tr>
<td>Slave4</td>
<td>2</td>
<td>4 GB</td>
<td>HDD &amp; SSD</td>
<td>Intel Core i5</td>
</tr>
</tbody>
</table>

**3.2 Experimental Dataset**

The dataset used to process experimental work includes (i) 100 smart meter load points of a month (20 GB dataset) (ii) 100 smart meter temperature points of a month (20 GB dataset) [29].
3.3 Experimental Results

The experiments conducted to evaluate our scheme are (i) Temperature and load scanner (ii) Data Bundler (iii) Message Response of SMErr module (iv) SM hot Socket prediction.

3.3.1 Temperature and load scanner

In order to evaluate the efficiency of Sample scanner, we perform scan analysis over two datasets i.e. Off Block and Peak Block. The off block dataset consists of normal load processing while Peak block dataset includes extreme load processing. We evaluate that off block dataset having 100 smart meters’ data samples scan 1.2% error in temperature and 2.5% error in load points in ‘1890’ and ‘2712’ seconds averagely as seen from Figure-13.

Moreover, we evaluate that peak block data having 100 smart meters’ data sample scan 1.7% error in temperature and 2.9% error in load points at ‘2090’ and ‘3105’ seconds averagely as seen from Figure-14.

3.3.2 Data Bundler

After the error analytics complete fetching tempered temperature and load points, SB_read method store the result set over SB_container. The SB_container is capable to store double primitive data of temperature and load points with SMID information. After the End_of_File (EOF) method is invoked at file_length limitation, bundler compile the DFS file as SB_file. We evaluate that data bundler in off load dataset write ‘2207’ entries of temperature over ‘2251’ seconds averagely. Furthermore, we evaluate that data bundler write ‘2811’ entries of load over ‘2876’ seconds averagely as seen from Figure-15.

Furthermore, we evaluate that data bundle in peak load dataset write ‘2417’ entries of temperature over ‘2486’ seconds averagely. Furthermore, we evaluate that data bundler write ‘3105’ entries of load over ‘3148’ seconds averagely as seen from Figure-16.

3.3.3 Message Response of SMErr Module

The purpose of SMErr module is to collect precise load and temperature tempered analysis for hot socket prediction. SMErr send a request_ack message to HBase repository with a query of requisite column data. Hadoop cluster route query message to HBase and respond with a data value of K, S, L, T_p and L_p component. The cluster exchange information at bandwidth 0.5 ≤ Bandwidth ≥ 20 MB/s. The message propagation strategy reduces message overhead from original one and consume 18% less bandwidth than traditional processing. The bandwidth utilization of multiple components can be observed from Figure-17.
3.3.4 Smart Meter Hot Socket Prediction

After training DT model, we run multiple simulations of predictions for possible Hot socket smart meter status i.e. Faulty and Not Faulty. We evaluated proposed scheme for three hours. At first hour of prediction, we predict ‘12’ smart meters could appear possible hot socket issue and ‘88’ smart meters do not present hot socket issue over a data sample of ‘5’ GB dataset. In the second hour of prediction, we predict we predict ‘14’ smart meters could appear possible hot socket issue and ‘86’ smart meters do not present hot socket issue over a data sample of ‘10’ GB dataset. In the last third hour of prediction, we predict we predict ‘18’ smart meters could appear possible hot socket issue and ‘82’ smart meters do not present hot socket issue over a data sample of ‘20’ GB dataset as observed from Figure-18.

4. CONCLUSION

This paper proposes a novel scheme to a state-of-an-art problem ‘hot socket’ in smart meters. The proposed approach scan tempered data samples and store it over a HBase repository. Furthermore, it synchronizes precise data samples over SMError module as a data source and predict hot socket possibility in smart meters. The experimental evaluation shows that SMADI is an efficient approach that predict hot socket issue before it appears in smart meters and reduce complexity of predicting hot socket problem over smart grid repository. In future, we would focus to work over domain integrated issues of smart meters.

ACKNOWLEDGEMENT

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

REFERENCES:


