HMM-Based Fast Detection of False Data Injections in Advanced Metering Infrastructure

Beibei Li[†], Rongxing Lu[‡], and Gaoxi Xiao[†]
[†]School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798
[‡]Faculty of Computer Science, University of New Brunswick, Fredericton, Canada E3B 5A3, Canada
Email: bli012@e.ntu.edu.sg; rlu1@unb.ca; egxxiao@ntu.edu.sg

Abstract—**Smart grids not only provide "intelligence" to the next generation power systems, but also potentially introduce vital security and privacy issues. Particularly, as a core part of the smart grids, advanced metering infrastructure (AMI) is suffering widespread disputes in terms of security and privacy concerns. This paper proposes a novel hidden Markov model (HMM) based method to detect false data injection attacks in AMI. In this method, a global-state HMM of the whole-house appliances is built and trained by sufficient historical meter data in an offline mode. Then, a new fast Viterbi algorithm is devised to decode the hidden states of the HMM. The decoded states are then verified via the partial sub-meter data in an online mode, by which false data can be detected. The effectiveness and efficiency of our method are verified by a public dataset AMPds with one-year real-time meter data.**

Keywords—**Smart grids, advanced metering infrastructure (AMI), security and privacy, false data injection, hidden Markov model (HMM).**

I. Introduction

The smart grid has been visioned to be a revolutionary alternative of traditional power systems with the expectation to achieve more efficient, reliable, and accurate power delivery [1], [2]. Advanced metering infrastructure (AMI) is an innovative architecture of a smart grid that enables automated and two-way communications between the smart meters and the utility [3]. The goal of an AMI is to provide the utility with near real-time power consumption data for pricing and billing purposes, and also allow customers to make informed choices about energy usage to improve energy efficiency and reduce energy budgets. Smart meters, the core intelligent electronic devices at the customer side in AMI, are capable of monitoring and accurately recording the energy usage data of the household appliances in real time bases.

While, tight integration of the power system with information and communications technology not only enables realization of the above-mentioned promising benefits, but also inevitably introduces significant security and privacy threats to the infrastructure [4]–[6]. One of the most significant security threats in AMI is false data injection (FDI) [7], where falsified energy usage data are reported to the utility. The falsified data can originate in two ways: the genuine energy usage data may be forged by compromised smart meters in the data collection stage; or the data may be tampered by man-in-the-middle attackers in the data transmission stage. Small-scale FDIs may reduce the energy efficiency, mess electricity billings, and impair the customer management; large-scale FDIs may mislead the demand response and load forecasting, and compromise distribution automation.

In recent years, many researchers have presented their work focusing on FDI attacks [5], [7]–[9]. However, most of the proposed methods, though applicable and effective, concentrate on FDI attacks occuring in the wide area networks of smart grids (e.g. wide area measurement and control system with phasor measurement units) [5], [8]. While, only a small number of studies focus on the home area networks and local area networks (e.g. AMI with smart meters) [7], [9]. In addition, for those focused on the AMIs, they either do not target on detection of FDIs or cannot effectively and quickly detect the FDIs. For example, Khanna *et al.* introduced an optimized FDI attack in AMI and showed the economic impacts they may cause but without any detection method proposed. Liu *et al.* proposed a collaborative intrusion detection mechanism against FDI in AMI. Unfortunately, their focus is on detecting malicious intrusions on smart meter itself, which cannot be used to effectively detect injected false data at the utility side.

As we see, it is still challenging to fight against the FDIs in AMI. In this paper, we are motivated to propose a hidden Markov model (HMM) based method to detect these FDIs. In this method, a global-state HMM is built to model the energy usage pattern of a house. Following the HMM, real-time aggregate meter data is then decoded by a fast Viterbi algorithm to estimate the true states of household appliances. The decoded states are finally verified by a small fraction of detailed meter data to detect the FDIs. We regard the contributions of this work to be three-fold:

- First, we propose a novel HMM-based method which can detect the FDIs in AMI with a high detection rate and efficiency. This can considerably enhance the security of AMI as well as the smart grid.
- Second, a fast Viterbi algorithm combined with the compressed sparse column (CSC) storage algorithm is devised. This algorithm can not only decode the local states of each household appliance in a quick and accurate mode, but also remarkably reduce the data storage space.
- Third, to our best knowledge, we are the pioneers to use HMM to detect the FDI attacks in AMI. This work sets the groundwork for future studies that employ the HMM to detect FDI attacks in cyber-physical systems.

II. Models and Design Goals

In this section, we show the system model, threat model as well as the design goals. The symbols used in this paper is summarized in Table I.

TABLE I
NOMENCLATURE

Symbol	Description
K	number of appliances in a house
X^k	local state of appliance k
X	global state of all appliances
M^k	number of local states of appliance k
M	number of global states of all appliances
N	number of observations of the global-state HMM
$\boldsymbol{\pi}$	row vector ($1 \times M$) of initial transition probabilities
$\mathbf{A}$	transition matrix ($M \times M$) of the global-state HMM
$\mathbf{B}$	emission matrix ($M \times N$) of the global-state HMM
$\mathbf{P}$	matrix ($M \times N$) storing the posterior probabilities
Nrow	number of rows of a matrix
Ncol	number of columns of a matrix
Nz	number of non-zero elements of a matrix
R	number of correctly decoded states
L	length of reported sub-meter data
η_{th}	threshold of decoding accuracy

A. System Model

Figure 1 shows our system model - the AMI architecture. AMI is an integrated system comprising household smart meters, data transmission networks, and the utility. A smart meter periodically collects the power consumption data of each appliance in a house at, for example, one-minute intervals or five-minute intervals. Then, the data transmission networks including broadband over power line (BPL), power line communications (PLC), fixed radio frequency (RF), and public networks (e.g., landline and cellular), aggregate and transmit the data to the utility data center. These transmitted data can either be sum-meter data or sub-meter data, or both. The sum-meter data is the sum of energy consumption data of all the appliances, while the sub-meter data is the separate energy consumption data of each appliance, e.g. HVAC [10]. In the utility data center, these collected data are further analyzed for real-time pricing, billing, energy management, and offering the customers with informed decisions to save energy budgets.

B. Threat Model

In this work, we consider FDIs as the main security threats. FDI in AMI refers that falsified meter data is received by the utility data center. The adversaries inject false meter data with possible purposes of stealing energy, causing economical losses of other customers or the utility, creating disorders to energy management and forecasting, and even power outages.

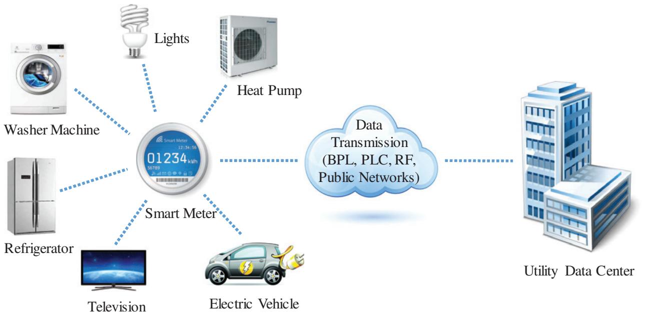

Fig. 1. The household smart meter and AMI architecture in a smart grid.

The received false data may come from two ways. In the data collection stage, if the smart meters are compromised by the adversaries, they can directly report forged meter data to the utility. In addition, in data transmission, man-in-the-middle attackers can intercept into the communication networks to tamper the contents of information packages. In this way, the original meter data can also be falsified.

C. Design Goals

The main design goal of this work is to address the above-mentioned threatening issues and guarantee the utility's data analysis with genuine meter data. Specifically,

- we aim to propose an HMM-based method that can effectively detect the potential false meter data to enhance the security level of smart grids;
- we also plan to design a fast Viterbi algorithm to quickly decode the HMM and use a CSC storage algorithm to reduce the storage space, both of which are beneficial to achieving real-time fast detection;
- our method is also expected to provide additional privacy preservation by quickly and accurately detecting the compromised smart meters; thus preventing subsequent meter data from leaking to the outside world.

III. Proposed Method

In this section, we present the details of our proposed method.

Figure 2 shows the flow diagram of the proposed method. We can see from the diagram that the method is composed of three steps: offline model training, fast decoding, and false data detection. Specifically, in the offline model training step, an HMM is established using sufficient historical meter data including all sub-meter data and the aggregated sum-meter data. Then, in the second step, a fast Viterbi algorithm is designed to decode the real local states of each appliance by using the obtained HMM and real-time sum-meter data. In the last step, we conduct online false data detection to identify falsified meter data. The technical details are elaborated in the following subsections.

A. Offline HMM Training

In this subsection, we discuss on the offline HMM training that includes defining the global-state HMM, quantizing

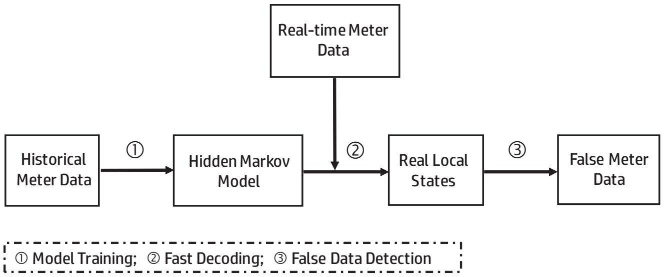

Fig. 2. The flow diagram of our proposed method.

the local states and global states, and estimating the model parameters of HMM.

1) Global-State HMM: In a house deployed with a smart meter and K smart appliances (e.g. television, washer, heat pump, etc.), the power consumption data of each appliance is periodically recorded by the smart meter. It is modeled that each appliance has M^k local states X^k, where $k \in \{1, 2, \cdots, K\}$. Let us take the washer as an example. A washer may be at any of the following 4 states: washing, rinsing, spinning, and off. Thus, the washer has $M^k = 4$ local states. The whole-house state X, also called the global state, is composed of all the appliances' local states, i.e. $X = \{X^1, X^2, \cdots, X^K\}$. The total number of the global states is, therefore, $M = \prod_1^K M^k$.

We model the state transition process of a house as an HMM [11]: $\boldsymbol{\lambda} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$, where $\boldsymbol{\pi} = \{\pi_0^1, \pi_0^2, \cdots, \pi_0^M\}$ is a row vector of initial prior probabilities of length M, $\mathbf{A}$ is an $M \times M$ transition matrix, and $\mathbf{B}$ is an $M \times N$ emission matrix. The entries of $\mathbf{A}$ and $\mathbf{B}$ are defined as $A[i, j] = p(X_t = j | X_{t-1} = i)$ and $B[j, n] = p(Y_t = n | X_t = j)$, where X_t is the global state and Y_t is the power observation at time instant t.

2) State Quantization: To obtain the exact model, we need to determine the parameters (i.e., $\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}$) of the HMM. A widely accepted way is to calculate or estimate these parameters using the abundant historical meter data.

One significant step to obtain these parameters, which is also a big challenge, is to determine the number of the local states of each appliance as well as the number of global states. Table II lists the number of distinct current readings of some appliances in AMPds dataset (version one) [12]. If all the shown numbers of states are utilized to build up the HMM, there will be a considerably huge state space resulting in heavy computation and storage burden. Therefore, it is of great importance to find a way to reduce the number of distinct readings. As an example shown in Fig. 3, although there are altogether 87 distinct readings of the clothes dryer, they mostly fluctuate and concentrate on only three different values. In this case, we can quantize all the current readings into three states: {S1,S2,S3}. Inspired by this example, our method of state quantization is: (1) plot the meter readings as Fig. 3; (2) eliminate those current values that have significantly low frequencies of occurrence (less than a threshold, e.g., 5%); (3) identify the peaks of current values as the quantized states.

TABLE II
NUMBER OF DISTINCT CURRENT READINGS

ID	Load Name	Distinct Current Readings
B1E	North Bedroom	14
B2E	South Bedroom	19
BME	Basement Plugs and Lights	52
CDE	Clothes Dryer	87
CWE	Clothes Washer	123
DWE	Dishwasher	48
FGE	Kitchen Fridge	133
GRE	Garage	71
HPE	Heat Pump	192
OFE	Home Office	72
TVE	Entertainment: TV, PVR, AMP	42

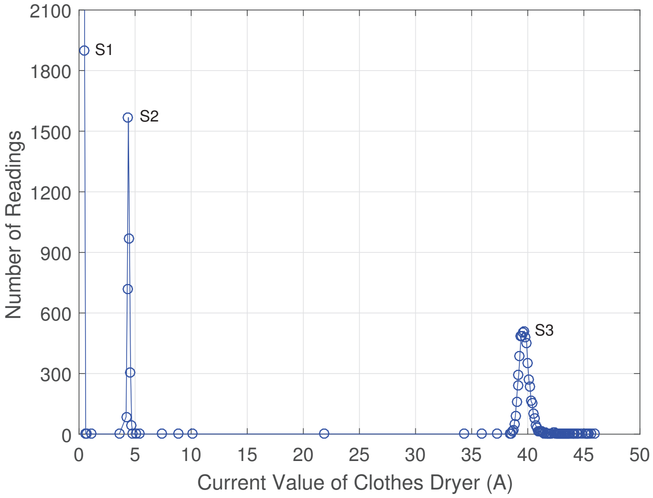

Fig. 3. The current values and corresponding number of readings of a clothes dryer.

By quantizing the current readings, the number of local states can be effectively reduced, which as a result, can also significantly reduce the number of global states.

3) Parameters Estimation: In order to build up an HMM that best models the user's electricity usage pattern, we take advantages of the historical meter data to estimate the model parameters. According to the above-mentioned state quantization method, we quantize all the current readings of the historical meter data and combine them together as global states. Following this, by processing these historical meter data, we can easily obtain the probability mass function of each global state used for initializing $\boldsymbol{\pi}(i), i \in \{1, 2, \cdots, K\}$, the transition probabilities between each pair of global states $A[i, j], i, j \in \{1, 2, \cdots, K\}$, and the emission probabilities from any global state to possible current observations $B[j, n], j \in \{1, 2, \cdots, K\}$ and $n \in \{1, 2, \cdots, N\}$. In this way, all the model parameters are determined. Note that, the parameter estimation can be finished in only several seconds for the whole AMPds dataset, so the cost of the offline training is not a big problem.

B. Sparse Matrix and Fast Viterbi

With the model parameters in hand, we are able to decode the hidden states by using a Viterbi algorithm [13]. However, though we have significantly reduced the state space, the sizes of matrices $\mathbf{A}$ and $\mathbf{B}$ are still very large. Bulky storage space and considerable computation costs are required for the Viterbi algorithm to solve the HMM problem.

Fortunately, it is found that matrices $\mathbf{A}$ and $\mathbf{B}$ are large but reasonably sparse as they have numerous zero entries [14]. Thus, it is possible for us to employ a compression algorithm to avoid storing these numerous zero values in memory, and reduce zero-value multiplications of the Viterbi algorithm. In our implementations, we make use of the CSC algorithm [15] to store $\mathbf{A}$ and $\mathbf{B}$, and propose a CSC-based fast Viterbi algorithm to solve the HMM problem. The existing CSC algorithm is presented in Algorithm 1 to ensure the integrity of our method, and our proposed fast Viterbi algorithm is shown in Algorithm 2.

Algorithm 1 CSC Algorithm [15]

```
procedure SPARSE_COMPRESS($\mathbf{M}$)
    Compute $Nrow$, $Ncol$, and $Nz$
    Initialize arrays $Val$, $RowIndex \leftarrow [0, 0, \cdots]$ with length $Nz$,
  and array $ArrIndex \leftarrow [1, 0, 0, \cdots]$ with length $Nz + 1$
    Initialize counter $count \leftarrow 1$
    for $col = 1$ to $Ncol$ do
        $countcol \leftarrow count$
        for $row = 1$ to $Nrow$ do
            if $\mathbf{M}[row, col] \neq 0$ then
                $Val(count) \leftarrow \mathbf{M}[row, col]$
                $RowIndex(count) \leftarrow row$
                $count \leftarrow count + 1$
            end if
        end for
        $countcol \leftarrow count - countcol$
        $ArrIndex(col + 1) \leftarrow ArrIndex(col) + countcol$
    end for
end procedure
```

In the CSC algorithm, a sparse matrix $\mathbf{M}$ is compressed and non-zero entries are stored in three arrays: *Val* stores the entry values, *ArrIndex* stores the indexes of columns, and *RowIndex* stores the row indexes of nonzero elements. Zero entries are ignored in this storage algorithm.

Algorithm 2 describes the proposed fast Viterbi algorithm. First, A function RETRIEVE$(\mathbf{M}, i)$ is defined to retrieve the non-zero elements of a given column (see lines 2-8). Given an observation Y_t corresponding to global state X_n, value pairs of non-zero elements in compressed emission matrix $\mathbf{B}$ are retrieved. These value pairs contain the possibilities as well as possible local states that can emit observation X_n. For each pair (j, p_{jn}) in $\mathbf{B}$, we can retrieve all the possible value pairs from the compressed $\mathbf{A}$ which contains the possibilities of all possible state transitions. The final step is to find the maximum posterior probability $\mathbf{P}_t[j]$ and the corresponding state index j.

Algorithm 2 Fast Viterbi Algorithm

```
 1: procedure FAST_VITERBI($\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}, \mathbf{P}, Y_t$)
 2:     function RETRIEVE($\mathbf{M}, i$)
 3:         $V \leftarrow [\ ]$
 4:         for $k = \mathbf{M}.ArrIndex(i)$ to $\mathbf{M}.ArrIndex(i+1) - 1$ do
 5:             $V.append \leftarrow (RowIndex(k), Val(k))$
 6:         end for
 7:         return $V$
 8:     end function
 9:     Initialize $\mathbf{P}_0[i] \leftarrow \boldsymbol{\pi}[i]$, for $i = 1, 2, \cdots K$
10:     Find the index $n$ of global state $X_n$ corresponding to $Y_t$
11:     for $(j, p_{jn}) \in$ RETRIEVE($\mathbf{B}, n$) do
12:         $S \leftarrow$ RETRIEVE($\mathbf{A}, j$)
13:         $\mathbf{P}_t[j] \leftarrow \max_{(i, p_{ij}) \in S} (\mathbf{P}_{t-1}[i] \cdot p_{ij} \cdot p_{jn})$
14:     end for
15:     return $\operatorname{argmax}_j(\mathbf{P}_t[j])$
16: end procedure
```

C. Online False Data Detection

As mentioned in our threat model, the compromised smart meters as well as communication channel interceptions may cause false meter data be fed to the utility. In this subsection, we devise an algorithm to detect the false meter data (see Algorithm 3).

Algorithm 3 False Data Detection

```
procedure DETECT($Y_t, Dsub$)
    for each smart meter, the utility data center do
        Collect $Y_t$ and $Dsub$ (with length $L$)
        Decodes $X_t$ using FAST_VITERBI algorithm with $Y_t$
        for $i = 1$ to $L$ do
            Compare $Dsub(i)$ with decoded state $X_t^k \in X$
        end for
        Count the number $R_t$ of correctly decoded states
        Compute the average decoding accuracy $\eta_t = \frac{R_t}{L}$
        if $\eta_t < \eta_{th}$ then
            False meter data detected
        else
            No false meter data
        end if
    end for
end procedure
```

In this detection algorithm, the utility data center collects the real-time sum-meter data Y_t and a fraction of sub-meter data *Dsub* with length L. Then, the global state $X_t = \{X_t^1, X_t^2, \cdots, X_t^K\}$ comprising all the local states is decoded by the proposed fast Viterbi algorithm. After that, the decoded local states are verified by the "true" states stored in *Dsub*, and the average decoding accuracy η_t is then computed. The final step is the judgement: if η_t is less than an experienced threshold η_{th}, false meter data is detected; otherwise no false meter data is detected. Note that our scheme can identify false meter data as long as a certain fraction of decoded states mismatch with the reported states. In other words, either the sum- or sub-meter data is false, or both are false, our method is effective in identifying the anomalies. Further, since a smart meter does not have the memory capacity and computational capability to train an HMM itself, even though both false sum-

and sub-meter data are simultaneously reported, they cannot perfectly feed into our HMM to avoid false data detection.

IV. Numerical Results and Discussions

In this section, we conduct intensive experiments to show the advantages of the CSC storage algorithm, proposed fast Viterbi as well as the detection algorithms. In our experiments, a public dataset AMPds [12] that is popular for household load research is fully utilized. AMPds contains one-year electricity measurements of 11 loads (see Table II) at one minute intervals. Each type of load contains 524,543 readings throughout the year. Experiments are conducted on a Windows machine with a 3.7 GHz Intel Xeon processor.

A. Matrix Sparsity and Storage Space with CSC Algorithm

In this subsection, we show the sparsity of transition matrix $\mathbf{A}$ and emission matrix $\mathbf{B}$, as well as the compression efficiency of the CSC storage algorithm. Table III shows the sparsity values of matrices $\mathbf{A}$ and $\mathbf{B}$ under different numbers of loads. Clear from this table that for both matrices $\mathbf{A}$ and $\mathbf{B}$, the sparsity grows with the number of loads. Interestingly, the sparsity of both the two matrices are significantly high (above 90%) when the number of loads is greater than 4. Since the number of loads in a house is usually above 10, so the sparsity value of matrices $\mathbf{A}$ and $\mathbf{B}$ would be considerably high at over 98%.

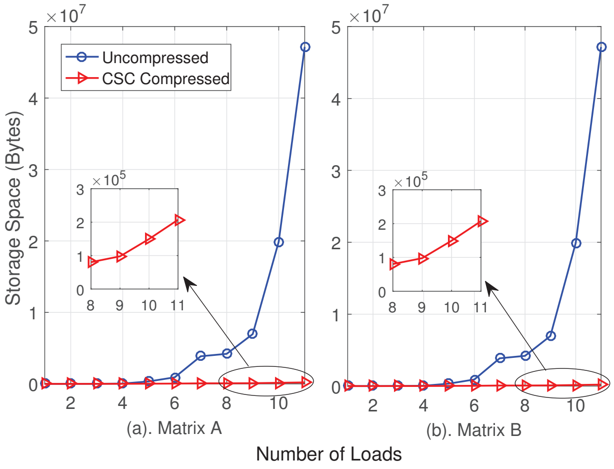

Fig. 4. The storage space comparison of CSC-compressed and uncompressed sparse matrices $\mathbf{A}$ and matrix $\mathbf{B}$.

To visually show the advantages of employing the CSC storage algorithm, the storage space of matrices $\mathbf{A}$ and $\mathbf{B}$ before and after the compression by the CSC algorithm are plotted in Fig 4. We can see from the figure that for both matrices $\mathbf{A}$ and $\mathbf{B}$, the storage space is dramatically reduced by adopting the CSC storage algorithm.

B. Decoding Efficiency and Accuracy of Viterbi Algorithms

The decoding efficiency and accuracy of the proposed HMM-based method are presented in this subsection. Figure 5 compares the elapsed time of decoding the training dataset under different numbers of loads for the proposed fast Viterbi algorithm and traditional Viterbi algorithm [13], respectively. As is shown, the elapsed time for the traditional Viterbi algorithm increases exponentially with the number of loads, while the time for the proposed fast Viterbi algorithm increases in a rather slow manner. Clearly, the proposed fast Viterbi algorithm outperforms the traditional Viterbi algorithm in terms of the elapsed time which also represents the performance of computation cost. Note that, we use offline decoding in this part to show the superiority of the proposed fast Viterbi algorithm over the traditional in [13]. While, real-world false data detection is conducted on an online decoding basis as shown in Algorithm 3. Only one-step FAST_VITERBI algorithm is needed, which is fairly fast and light-weight.

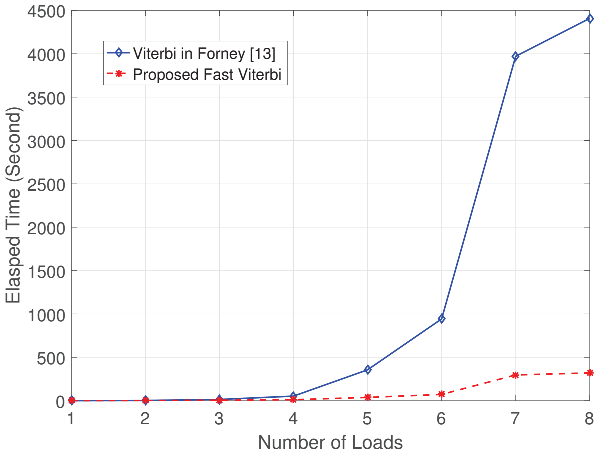

Fig. 5. The elapsed time comparison of offline decoding of the training dataset between the traditional Viterbi and fast Viterbi algorithms.

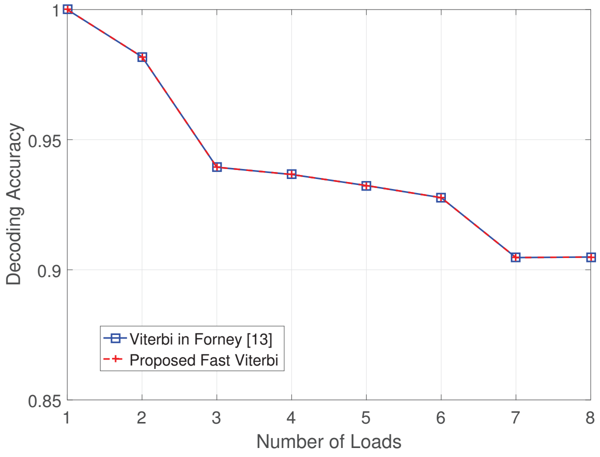

Fig. 6. The decoding accuracy by using HMM.

Figure 6 shows the decoding accuracy of the proposed method under traditional and fast Viterbi algorithms, respectively. It is clear from the figure that both traditional and fast Viterbi algorithms achieve the same and fairly high accuracy of above 90%. We also see that as the number of loads increases, the decoding accuracy decreases and in a slow mode. This is

TABLE III
SPARSITY VALUES OF TRANSITION AND EMISSION MATRICES $\mathbf{A}$ AND $\mathbf{B}$ IN THE HMM

Number of Loads	1	2	3	4	5	6	7	8	9	10	11
Sparsity of Matrix $\mathbf{A}$	22.22%	59.50%	78.55%	90.03%	96.39%	97.95%	99.06%	99.11%	99.35%	99.66%	99.80%
Sparsity of Matrix $\mathbf{B}$	66.67%	87.17%	90.34%	94.97%	96.33%	97.16%	97.86%	97.89%	98.57%	98.78%	98.86%

because that the more loads being included in the HMM, the higher complexity of the customer behavior pattern is, which leads to more decoding errors. Another important finding is that compared with the traditional Viterbi algorithm, the proposed fast Viterbi algorithm will not change the decoding accuracy but only save the decoding time.

C. Detection Rate of the Proposed Method

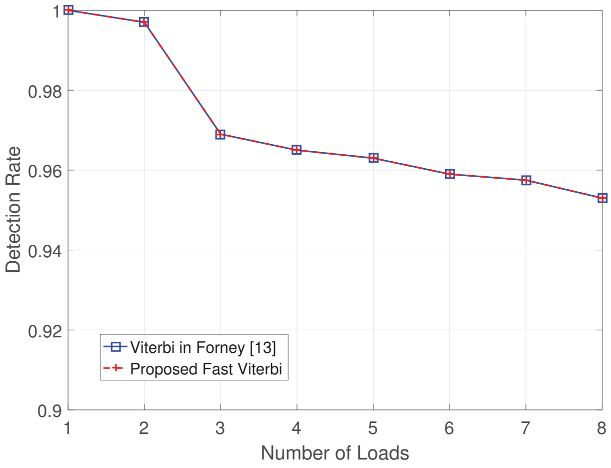

Fig. 7. The detection rate by using HMM.

The detection rate of our method is plotted in Fig. 7. We train the HMM using the first 450,000 meter readings and insert 1,000 false meter data in random positions of the rest 745,543 meter readings. The detection rate is computed as the percent of detected false meter data over the total number of inserted false meter data. L is set as 3 when the number of loads exceeds 3; and set as 1, otherwise. The threshold of decoding accuracy η_{th} is designed as 0.8 here. As the results shown in Fig. 7, our method can achieve a very high detection rate to identify the injected false meter data under the above-mentioned parameter settings.

Future research direction will further extend and consolidate this work by evaluating the effects on the detection rate under different values of η_{th}, L, as well as various strategies of false data generation. In addition, how to update the HMM when the household appliance profile changes deserves further investigation. In this work, we assume that the profile does not change. It is reasonable for most cases, but not applicable for all the cases.

V. CONCLUSIONS

In this paper, we proposed a novel HMM-based method to quickly and accurately detect false data in smart grid. We have built a global-state HMM and trained it by historical meter data. In addition, a CSC-based fast Viterbi algorithm is also devised which can significantly mitigate the storage space and achieve fast decoding. The experiments on real dataset AMPds fully demonstrate the compression efficiency, decoding efficiency, decoding accuracy, and detection rate of our method.

REFERENCES

[1] R. X. Lu, X. H. Liang, X. Li, X. D. Lin, and X. M. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.

[2] R. Deng, G. Xiao, R. Lu, and J. Chen, "Fast distributed demand response with spatially and temporally coupled constraints in smart grid," *IEEE Trans. Ind. Inf.*, vol. 11, no. 6, pp. 1597–1606, Dec. 2015.

[3] P. Yi, T. Zhu, Q. Zhang, Y. Wu, and J. Li, "A denial of service attack in advanced metering infrastructure network," in *Proc. of the IEEE International Conference on Communications (ICC)*, Sydney, Australia, June 2014, pp. 1029–1034.

[4] J. Wei, D. Kundur, T. Zourntos, and K. L. Butler-Purry, "A flocking-based paradigm for hierarchical cyber-physical smart grid modeling and control," *IEEE Trans. Smart Grid*, vol. 5, no. 6, pp. 2687–2700, Nov. 2014.

[5] B. Li, R. Lu, W. Wang, and K.-K. R. Choo, "DDOA: A dirichlet-based detection scheme for opportunistic attacks in smart grid cyber-physical system," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2415–2425, Nov. 2016.

[6] R. Berthier, W. H. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: Requirements and architectural directions," in *Proc. of the First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Maryland, USA, Oct. 2010, pp. 350–355.

[7] X. Liu, P. Zhu, Y. Zhang, and K. Chen, "A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2435–2443, Sep. 2015.

[8] S. Li, Y. Yılmaz, and X. Wang, "Quickest detection of false data injection attack in wide-area smart grids," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2725–2735, Dec. 2015.

[9] K. Khanna, B. K. Panigrahi, and A. Joshi, "Data integrity attack in smart grid: optimised attack to gain momentary economic profit," *IET Generation, Transmission & Distribution*, vol. 10, no. 16, pp. 4032–4039, Dec. 2016.

[10] L. Mane, "Submetering: A practical approach," *GE ESL Magazine*, pp. 1–13, Summer 2005.

[11] S. R. Eddy, "Hidden Markov models," *Curr. Opin. Struct. Biol.*, vol. 6, no. 3, pp. 361–365, June 1996.

[12] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Baji, "AMPds: A public dataset for load disaggregation and eco-feedback research," in *Proc. of the IEEE Electrical Power & Energy Conference (EPEC)*, NS, Canada, Aug. 2013, pp. 1–6.

[13] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[14] S. Makonin, F. Popowich, I. V. Baji, B. Gill, and L. Bartram, "Exploiting HMM sparsity to perform online real-time nonintrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 7, no. 6, Nov. 2016.

[15] I. S. Duff, R. G. Grimes, and J. G. Lewis, "Sparse matrix test problems," *ACM Trans. on Mathematical Software (TOMS)*, vol. 15, no. 1, pp. 1–14, Mar. 1989.