Securing Advanced Metering Infrastructure Using Intrusion Detection System with Data Stream Mining

Mustafa Amir Faisal[1], Zeyar Aung[1], John R. Williams[2], and Abel Sanchez[2]

[1] Computing and Information Science Program,
Masdar Institute of Science and Technology, Abu Dhabi 54224, United Arab Emirates
{mfaisal,zaung}@masdar.ac.ae
[2] Engineering Systems Division, Massachusetts Institute of Technology (MIT),
Cambridge, Massachusetts 02139, United States of America
{jrw,doval}@mit.edu

Abstract. Advanced metering infrastructure (AMI) is an imperative component of the smart grid, as it is responsible for collecting, measuring, analyzing energy usage data, and transmitting these data to the data concentrator and then to a central system in the utility side. Therefore, the security of AMI is one of the most demanding issues in the smart grid implementation. In this paper, we propose an intrusion detection system (IDS) architecture for AMI which will act as a complimentary with other security measures. This IDS architecture consists of three local IDSs placed in smart meters, data concentrators, and central system (AMI headend). For detecting anomaly, we use data stream mining approach on the public KDD CUP 1999 data set for analysis the requirement of the three components in AMI. From our result and analysis, it shows stream data mining technique shows promising potential for solving security issues in AMI.

Keywords: Data stream mining, Advanced metering infrastructure (AMI), Smart grid, Intrusion detection system (IDS).

1 Introduction

Smart grid (SG) is the integration of modern information technologies for two-way communication, updating users about their consuming behavior, monitoring electric grid health system, controlling home appliances and other smart grid components remotely, etc. with present power system. To provide these facilities SG needs to introduce many devices as well as applications and thus communicating, monitoring, and controlling them may require new protocols and standards. This modernized and complex electric grid system is also exposed to augmented security threats like any other complex systems. Because of its inter-operability nature, SG exposes many security vulnerabilities. If proper initiatives are not taken, then there can be a huge catastrophic impact on the whole system and thus to the society. For this reason, NIST (National Institute of Standards and Technology) and FERC (Federal Energy Regulatory Commission) identify cyber

M. Chau et al. (Eds.): PAISI 2012, LNCS 7299, pp. 96–111, 2012.
© Springer-Verlag Berlin Heidelberg 2012

security as one of the vital areas for further exploration [12]. In this paper we focus on security issues and solutions for AMI.

AMI can be conceived as an attachment for providing bidirectional communication between user domain to utility domain [2]. This sophisticated infrastructure forms a high speed media to exchange information flow between these domains. The principle functionalities of AMI encompasses bidirectional communication and power measurement facilities, assisting adaptive power pricing and demand side management, self-healing ability, and providing interfaces for other systems [20]. However, AMI exposes to various security threats like privacy breach, monetary gain, energy theft, and other malicious activities. As AMI is directly related to the revenue earning, customer power consumption and privacy, the utmost important is to secure its infrastructure.

IDS is a monitoring system to detect any unwanted entity into a system (like AMI in our context). IDS can be signature-based, specification-based, and anomaly-based [4]. A signature-based IDS builds a back list of attacks. It is not suitable for AMI because new types of attacks are growing frequently since AMI is an emerging system. On the other hand, a specification-based IDS can be promising solution for AMI as mentioned in [4,3]. However, building a specification for AMI networks is neither easy nor cost effective. As AMI evolves, fresh specifications are to be added. Hence, changing specifications in all key sensors would be expensive and cumbersome. In this paper, we choose to employ anomaly-based IDS using data mining. However, instead of considering conventional static mining techniques, we opt to stream mining, precisely evolving data stream mining, as we believe it to be a more realistic approach in real-world network monitoring and intrusion detecting.

In this paper, we regard IDS as a second line security solution after firewall, cryptography, authorization techniques etc. which are first line security measures. However, [9] stresses that only these first line security steps will not be sufficient for securing AMI because such kinds measures do not concern the detection of anomalies if they occur. Our IDS architecture is based on AMI architecture prepared by OPENMeter [24], a project deployed by several European countries to reduce gap between the state-of-art technologies and AMI's requirements. Data stream mining for each component (smart meter, data concentrator, and headend) in AMI is analyzed with MOA (Massive Online Analysis) [21,7].

The contributions of this paper are that: (1) we discuss security issues in AMI and conduct a literature review of IDS in general as well as for AMI; (2) we propose an IDS architecture which would enhance the security and reliability of the AMI network; and (3) we make an analysis of the individual IDS for each AMI component.

The rest of the paper is organized as follows: in Section 2, a brief overview of AMI is provided. The related works for AMI security issues, IDS with stream data mining, and IDS in AMI are discussed in Section 3. Section 4 is focused on our proposed IDS architecture for AMI in detail. Section 5 describes the experimental design for accuracy analysis of IDS in each AMI component in proposed architecture. We discuss the results and analysis of our experiment in section 6. A comparison with those of the existing works as well as an evaluation of our architecture are presented in Section 7. Finally, we conclude our paper in Section 8 with our future research plan.

2 AMI Overview

AMI is mainly composed of smart meters, data concentrators, communication networks, and central system (AMI headend). However, these components are located in various networks [4] and different realms (i.e., private and public realms) [27]. An overview of AMI components and networks (highlighted with dotted lines) within the bigger context of electric power distribution, consumption, and renewable energy generation/storage are shown in Figure 1.

3 Related Work

3.1 Security Issues in AMI

Due to unique requirements (e.g., real time, confidentiality, integrity, availability, etc.) and constraints (e.g., topology, bandwidth, computational power, and memory size), AMI is vulnerable for various attacks in every sector. Cleveland [9] focuses on the security requirements and threats in AMI by pointing confidentiality, integrity, availability, non-repudiation in AMI. Moreover, that paper also mentions the unique constraints of each unit in AMI which should take into consideration in designing security solution for AMI. As [4] mentions, like many other systems, AMI requires three stages solutions: first one for prevention which encompasses secure protocol, authorization and authentication techniques, firewall, etc. Second solution is detection which includes IDS, etc. and finally, mitigation or resilience, i.e., recovery activities after the attack. Besides this, authors in [4] builds a threat model for AMI where they identify various attackers, attack techniques, and their consequences.

McLaughlin *et al.* [22] and Cleveland [9] mention, along with physical tampering, a smart meter in AMI is more vulnerable than a pre-AMI analog meter. The reason is the current smart meters, run by software, can be compromised easily. Energy theft is an immense concern in smart metering, the paper emphasizes that. Shein [27] focuses on weakness of present meters and their physical vulnerabilities. Data collectors are vulnerable because of their physical locations as well as communication networks may suffer direct attacks for lacking of required separation of traffic and access. Openness of AMI networks to external, unsecured environments is also a big threat [9]. Though the AMI headend, located in utility office, is more secure when compared with other AMI components, the data and control commands are more accessible [9,27]. Besides this, a dissatisfied employee can be a significant threat against this cental system [9].

3.2 Stream Data Mining for Intrusion Detection System

Using stream data mining for IDS in computer network is comparatively new. Chu *et al.* [8] propose an architecture for network IDS using single-pass technique including both anomaly and signature-based mechanisms. In that architecture, IDS is split into two modules: passive module which monitors data stream using signature-based detection and active module which uses anomaly-based detection mechanism. For implementing this architecture, they propose a single pass

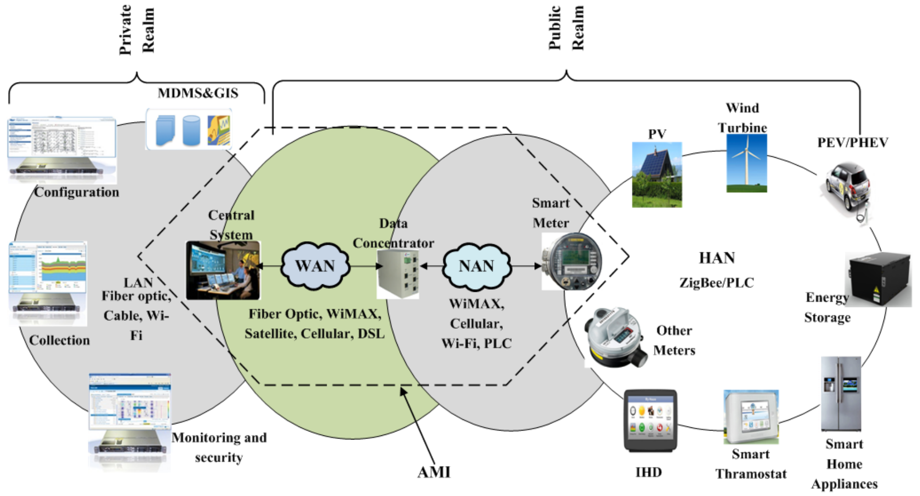

Fig. 1. Overview of AMI components and networks. (AMI = Advanced Metering Infrastructure; DSL = Digital Subscriber Line; GIS = Geographic Information System; HAN = Home Area Network; IHD = In Home Display; LAN = Local Area Network; MDMS = Meter Data Management System; NAN = Neighborhood Area Network; PEV = Plug-in Electric Vehicle; PHEV = Plug-in Hybrid Electric Vehicle; PLC = Power Line Communication; PV = Photovoltaic.)

algorithm named FP-Stream. In [23], Oh *et al.* proposes a clustering method for anomaly detection which considers the number of clusters is unknown and a cluster can be divided into two clusters and a couple of clusters can be merged into a single cluster according to object distribution in the data stream. To augment the accuracy and efficiency, Li *et al.* [19] presents an model based on sequence mining for NIDS (network IDS). In this model, the authors use multidimensional item set to outline network events. Moreover, sliding window and sequence mining algorithms are utilized for collecting network data stream and detecting intrusion respectively. Using fuzzy logic, Khan [17] introduces an automated annotation for obtaining results from data stream clustering. The principle feature is interpreting the nature of clusters, anomaly or normal, without human help. Recently, Zhang and Huang [29] propose TFSE (Time-table-joined Frequent Serial Episodes) for extracting required patterns and rules for intrusion detection from time series stream.

3.3 Intrusion Detection System in AMI

As smart grid and AMI are relative new concepts, few research works are accomplished regarding IDS in AMI. Berthier *et al.* in [4] discuss components, technologies, type of IDSs. They also propose an AMI monitoring architecture using a distributed scheme where most data processing will be done by sensors located in meter network. A centralize component coordinates the sensors' tasks and collecting upper level alerts. Resource requirements for this IDS are network configuration, protocol specifications, system and network security policies, and statistical profiles. The authors mention their believe that specification-based IDS would be best approach for AMI and the reasons are: (1) specification-based IDS has better accuracy level over signature-based IDS; (2) lack of empirical data to build a blacklist of signatures; and (3) limited number of protocols and applications will be monitored in AMI and for this specification would be cost-effective as AMI is a controlled environment. Berthier and Sanders in [3] extend their work where the sensors, characterized with specification-based intrusion detection mechanism, placed in key access points to monitor network, transport, and application layers in the OSI model. Using state machine, studying constraints, and requirements, they build four rules to detect traffic modification, injection, replay, compromised meters, preventing large set of meters from a malicious node, and DoS (Denial of Service) attacks. Moreover, they emulate AMI environment where these four rules test the performance. At the application-layer, a formal verification of the specifications and monitoring operations are conducted.

Zhang *et al.* [30] propose a distributed IDS using a multi-layer network architecture for complete smart grid including AMI and SCADA. Three IDSs would be placed in HAN, NAN, and WAN (wide area network). For detecting intrusion, they use Support Vector Machine (SVM), two clonal selection algorithms named CLONALG and AIRS2Parallel, which are derived from artificial immune system (AIS).

4 Proposed Architecture

For each component of AMI, we believe that it is more practical to assume the data as a stream. That means, the data is sequentially continuous, much larger in size than the device's memory, and most importantly mining algorithm can trace data for a very limited number of times. For this reason, we characterize the nature of data in each component of AMI in Table 1. At present, industrial meters have very limited amount of memory for necessary updates [27]. In our opinion, a smart meter's memory and processing capacity should be enhanced in order to provide it with enough security protection (as well as other analytical capabilities). The security of smart meter is very essential because it is responsible for remote access of various smart home appliances, pricing, energy usage recording, etc. Smart meter will be installed in houses of ordinary people as well as in crucial places like banks, hospitals, educational institutes, parliaments, and presidential houses. To solve this security issue, we propose a security entity, named 'security box', which can be integrated within or outside a smart meter. A possible design of smart meter is provided in Figure 2(a) based on the one presented in [10]. Although we show 'security box' as a simple meter IDS (M-IDS), in the future, it can also include other security mechanisms like encryption, authentication, firewall, etc. The IDS will be placed in three AMI components' premises, in smart meter, data concentrator, and central system (headend). The configuration of IDS for the smart meter is shown in Figure 2(b). Similarly, IDSs in the data concentrator and the headend have the same configuration albeit with different amount of resources.

Table 1. Characteristics of smart meter, data concentrator, and AMI headend

Smart meter	**Data Concentrator**	**AMI Headend**
Similarity		
Data is continuous in each of them.		
Differences		
Data is small in amount as data sources are customer's HAN (home area network) and its associated devices (like other meters).	Data is comparatively in high volume as it has to handle data from about a few hundred to tens of thousands of smart meters [4].	Data is in huge volume as it has to tackle data from about several million smart meters [4].
The resources like memory (in kilobyte range), processing capacity etc. are very restrictive.	The resources are higher (in megabyte range [11]).	The resources are very high because they are usually power server.
Data speed is comparatively low because of non-frequent requests at the smart meter.	Data speed is high as it aggregates good number of smart meter data.	Data speed is very high as it has to handle huge amount of meter data, event data, command, etc.

One noticeable thing, our proposed IDS architecture follows a sequential process. Communication data from various sources are inserted in Acceptor Module. Pre-processing unit is responsible to generate data according to the predefined attributes by monitoring the communication data and this generated data acts as input for stream mining module. Stream mining module run the stream mining algorithm over the generated data set by data maker module. Decision maker unit decides whether it should trigger an alarm or not. This module also keeps record for the corresponding information for attacks. The overall IDS architecture is shown in Figure 3. For inter-communication, IDSs can use separate network which is also mentioned in [4]. Though this dedicated network is expensive, it increases reliability. If IDS uses same AMI communication network and a node is compromised by an attacker, the purpose of IDS will be defeated.

An intrusion detection flow chart from smart meter to central system is depicted in Figure 4. The flow can also be initiated from the headend to the smart meter. However, some devices like O&M modules can be connected to IDS locally.

5 Experimental Setup

5.1 Data Set

We use an improved version [28] of KDD Cup 1999 data set [16]. This data set has 41 features as well as training and testing data set has 22 and 38 distinct type of attacks. However, these attacks can be categorized into 5 broad types which are: (1) normal, (2) denial-of-service (DOS) (e.g. syn flood, land, back, etc.), (3) unauthorized access from a remote machine (R2L) (e.g., guessing password); (4) unauthorized access to local superuser (root) privileges (U2R) (e.g., different 'buffer overflow' attacks), and finally (5) surveillance and other probing (e.g.,

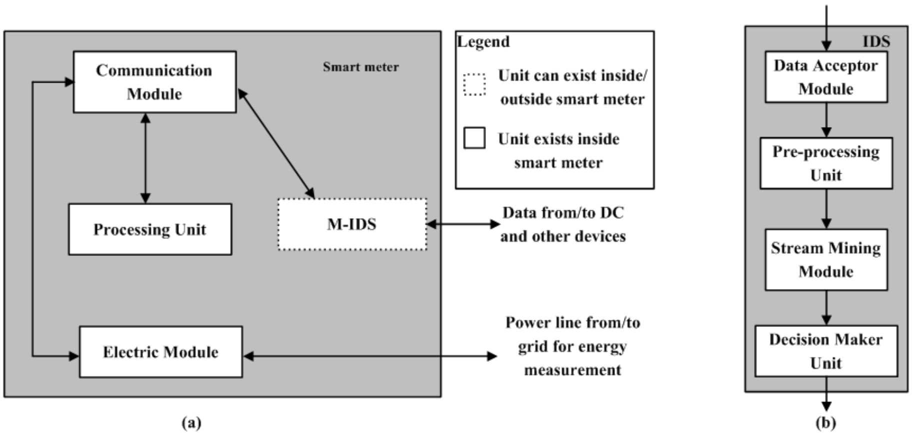

Fig. 2. (a) Smart meter with IDS. (M-IDS = Meter IDS.) (b) IDS for a smart meter.

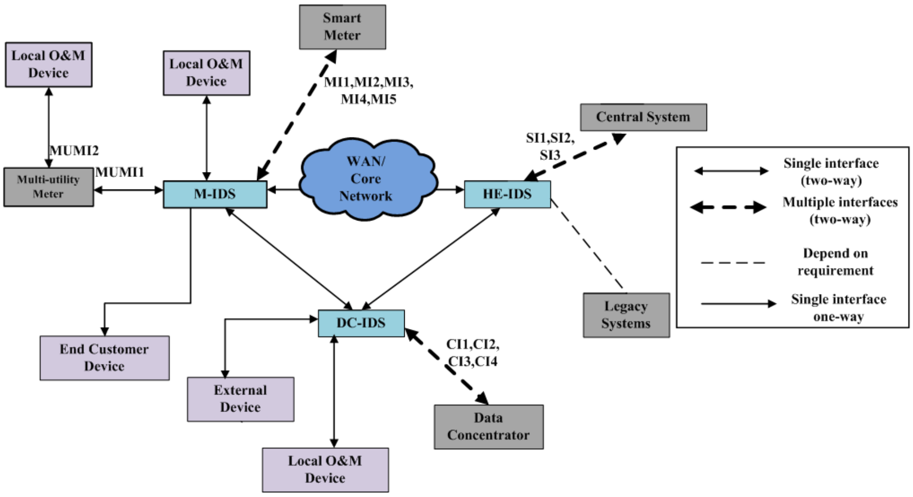

Fig. 3. Architecture of whole IDS in AMI. (CI = Concentrator Interface; DC-IDS = Data Concentrator IDS; HE-IDS = Headend IDS; M-IDS = Meter IDS; MI = Meter Interface; MUMI = Multi-utility Meter Interface; O&M = Operations and Maintenance; SI = Cental System Interface.)

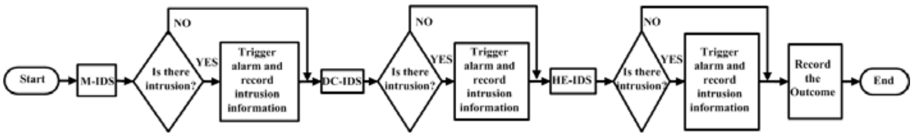
Start
M-IDS
Is there intrusion?
NO
YES
Trigger alarm and record intrusion information
DC-IDS
Is there intrusion?
NO
YES
Trigger alarm and record intrusion information
HE-IDS
Is there intrusion?
NO
YES
Trigger alarm and record intrusion information
Record the Outcome
End

Fig. 4. Intrusion detection procedure from smart meter to AMI headend

port scanning). A tabular form of size of sample and various attack types for both training and testing data set is given in Table 2.

Table 2. Various attacks' sample sizes

Type of sample	Training data set	Testing data set
Normal	67,343	9,711
R2L	942	1,656
U2R	105	1,298
DOS	45,927	7,458
Probing	11,656	2,421
Total	125,973	22,544

5.2 Algorithms Explored

We use evolving data stream mining classifiers in MOA [21,7] as they are capable to cope with concept drift in data streams. There are 16 evolving data stream classifiers in MOA. After an initial trail on those 16 classifiers, the following ensemble learners (with their respective base or component classifiers shown in parentheses) are selected because of their higher accuracy (evaluated with *EvaluatePrequential* [14]) for training data set.

For evaluating the performance of the classifiers, we use *EvaluatePrequential* evaluation approach in MOA. This evaluation technique first tests the classifier and then trains for each example in the data stream. For this method we apply *BasicClassificationPerformanceEvaluator* as an evaluator which measures the performance of classifier from beginning of data stream, instead of instances in a window like by *WindowClassificationPerformanceEvaluator* evaluator.

The four selected classifiers are briefly described as follows:

1. *Leveraging Bagging* (*HoeffdingTreeNB*) [6];
2. *LimAttClassifier* (*LimAttHoeffdingTreeNBAdaptive*) [5];
3. *OzaBagAdwin* (*HoeffdingTreeNB*) [25,26];
4. *Single Classifier Drift* (*HoeffdingTreeNBAdaptive*) [1,13].

The base learners used by these classifiers are variants of *HoeffdingTree* [15]. It is based on the Hoeffding bound which quantifies the number of observations required to estimate necessary statistics within a prescribed precision. Mathematically, Hoeffding bound can be expressed using Equation (1).

$$\in = \sqrt{\frac{R^2 ln(1/\delta)}{2n}} \tag{1}$$

This equation says that the true mean of a random variable of range R will not differ from estimated mean after n independent examples or observations by more than $\in$ with probability $1 - \delta$. Brief descriptions of the four selected ensemble classifiers are given below.

Leveraging Bagging: Mainly two randomization improvement techniques are applied to enhance bagging performance in this classifier. For first improvement, Bifet *et al.* [6] propose to use higher value of λ to compute the Poission distribution's value which would increase the re-sampling weights. For second enhancement, randomization is added at the output of the ensemble using error-correcting output codes. When a new instance x arrives it is assigned to the class with nearest binary code. An error-correcting code can be viewed as a form of voting in which a number of incorrect votes can be corrected. The main motivation for using random code instead of deterministic codes is that each classifier in ensemble will predict a different function which may reduce the correlation effects among the classifiers and thus, the diversity of the ensemble will be increased. Each classifier m and class c are assigned binary value $\mu_m(c)$ in an uniform, independent, and random way. Exact half of the classes are mapped to 0. The output of the classifier for an example is the class which has more votes of its binary mapping classes. To deal with concept drift in data stream, ADWIN, a change detection method for data stream, is used.

LimAttClassifier: This ensemble classifier combines restricted Hoeffding Trees using stacking. A classification model based on an ensemble of restricted decision trees are generated. Each decision tree is built from a unique subset of attributes. The whole model is formed by mixing the log-odds of the predicted class probabilities of these trees using sigmoid perceptrons, with single perceptron for individual class. ADWIN is used for setting perceptrons' learning rate as well as for resetting Hoeffding trees when they no longer perform well. Instead of forming an ensemble classifier in a greedy fashion like in standard boosting approach, LimAttClassifier builds each Hoeffding tree in sequence and assigns related weights as a by-product. Thus, each tree generated in parallel and then these trees are combined using perceptron classifiers by adopting the stacking approach. Adaptive naive Bayes Hoeffding Trees with limited attributes show better performance instead of using individually naive Bayes or majority class for prediction.

OzaBagAdwin: The main idea of this algorithm is to use a sliding window, not fixed a priori, whose size is recomputed in online according to the change rate observed from the data in window itself. The window will grow or shrink keeping pace with change in data stream. For this reason, OzaBagAdwin uses ADWIN2, an enhanced version of ADWIN in term of time- and memory efficiency. This change detection technique holds a window of length W with $O(\log W)$ memory and update time. The classifier provides a performance guarantee by bounding the rates of false positives and false negatives.

Single Classifier Drift: Single Classifier Drift is an evolving classifier with a wrapper on it for handling concept drift in data stream. For this, drift detection method (DDM) [13] or early drift detection method ($EDDM$) [1]. In *DDM*, the number of errors produced by learning model during prediction are controlled. This procedure is done by comparing the statistics of two windows where first one contains all the data and second one contains only the data from the beginning until number of errors increases. For *EDDM*, an enhanced version of *DDM*, the fundamental idea is to consider the distance between two errors classification instead of considering only the number of errors. Increasing the average distance between two errors and improving the prediction are the improvements of this method.

We use the default parameter values provided the GUI version of MOA for most of the parameters for the above mentioned classifiers. However, for a few parameters, we set their values as mentioned in Table 3 for *EvaluatePrequential* task. All experiments are carried out in a PC with Intel Core i7-2600 CPU 3.4GHz, 8.00 GB RAM, and 64-bit Windows 7 Professional.

Table 3. Changed Parameters and their corresponding tuned values for *EvaluatePrequential*

LimAttClassifier		**OzaBagAdwin**	
Parameter	**Value**	**Parameter**	**Value**
`numAttributes` (number of attributes to use per model)	2	`gracePeriod` (The number of instances a leaf should observe between split attempts)	26
		`splitConfidence` (allowable error in split decision)	0.064
LeveragingBag		**SingleClassifierDrift**	
Parameter	**Value**	**Parameter**	**Value**
`gracePeriod` (The number of instances a leaf should observe between split attempts)	27	`binarySplits` (Only allow binary splits)	checked
`splitConfidence` (allowable error in split decision)	0.95	`splitConfidence` (allowable error in split decision)	0.95
		`gracePeriod` (The number of instances a leaf should observe between split attempts)	25
		`tieThreshold` (Threshold below which a split will be forced to break ties)	0.045
For EvaluatePrequential			
Parameter		**Values**	
`evaluator` (Performance evaluation method)		BasicClassificationPerformanceEvaluator	

The complete flow of our experiment is depicted in Figure 5. This procedure repeat for all evolving classifiers to select the best ones.

6 Results and Analysis

For comparing the performance of the algorithms, the criteria of (1) accuracy, (2) Kappa Statistic, (3) FPR (False Positive Rate), and (4) FNR (False Negative Rate) (see in Table 4) as well as (5) time and (6) memory consumption are used (see in Figure 6).

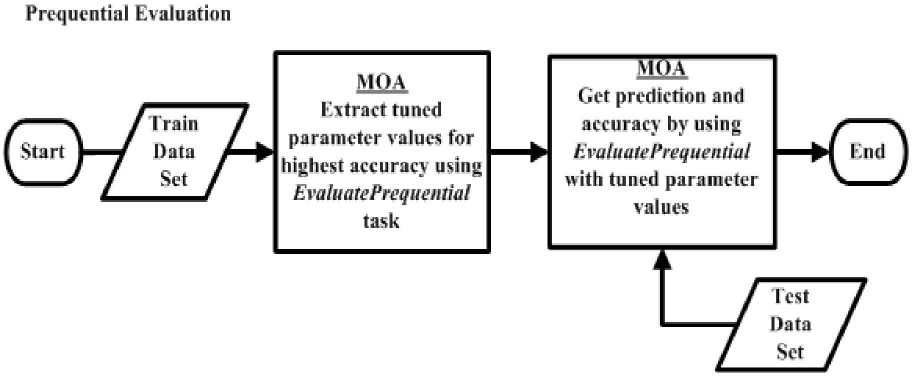
Prequential Evaluation
Start
Train Data Set
MOA
Extract tuned parameter values for highest accuracy using *EvaluatePrequential* task
MOA
Get prediction and accuracy by using *EvaluatePrequential* with tuned parameter values
End
Test Data Set

Fig. 5. Experimental Flow (left to right)

Table 4. Performance comparison for classifiers

Classifier vs. Performance measures	**OzaBag Adwin**	**Leveraging Bag**	**LimAtt Classifier**	**Single Classifier Drift**
Training Data Set				
Accuracy(%)	98.61	99.41	99.49	99.16
Kappa Statistic(%)	97.57	98.97	99.12	98.54
Test Data Set				
Accuracy(%)	96.05	95.65	96.58	93.97
Kappa Statistic(%)	94.21	93.60	95.01	91.1
FPR(%)	2.15	1.6	2.48	2.49
FNR(%)	5.55	6.85	3.39	8.23

A graphical comparison among the models of these classifiers in terms of correct prediction of particular types of attacks is provided in Figure 7.

Table 4, *LimitAttClassifier* shows best performance among the four classifiers, though it has a higher FPR than that of for both *OzaBagAdwin* and *LeveragingBag* classifiers. However, highest number of *Normal* and *R2L* attacks are identified by *SingleClassifierDrift* and *OzaBagAdwin* respectively (seen in Figure 7). On the other hand, the largest amounts of *DOS*, *U2L*, and *Probing* attacks are detected by *LimitAttClassifier*. Though for training data set *SingleClassifierDrift* performs well, it shows lowest accuracy for test data set. However, from Figure 6, we can see both time and memory requirements for *LimitAttClassifier* and *LeveragingBag* are quite high in spite of their good performance for other metrics (like accuracy, etc.). On the other case, *SingleClassifierDrift* shows its limited resource requirements. The later classifier can be potential one for its deployment in smart meter due to resource restrictive nature of this device. In addition, this classifier should be further investigated to improve its accuracy level as well as FPR and FNR. High time requirement specially for *LimitAttClassifier* may

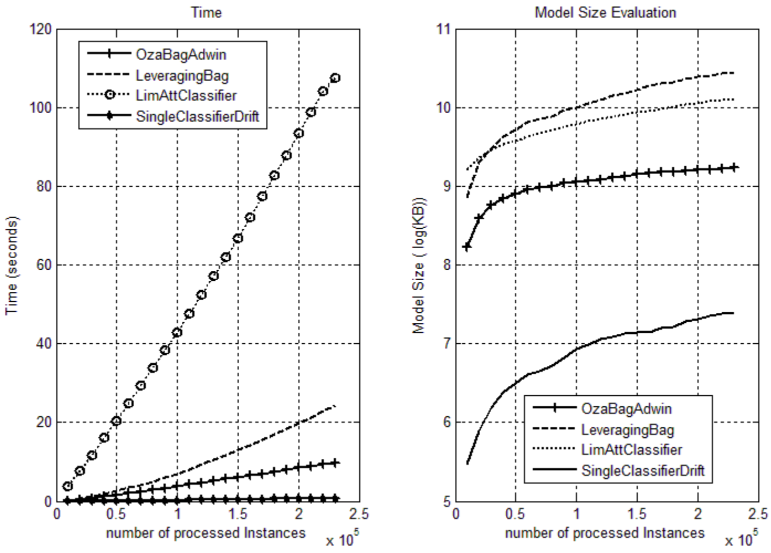

Fig. 6. Time (left) and memory consumption (right) comparison among the classifiers

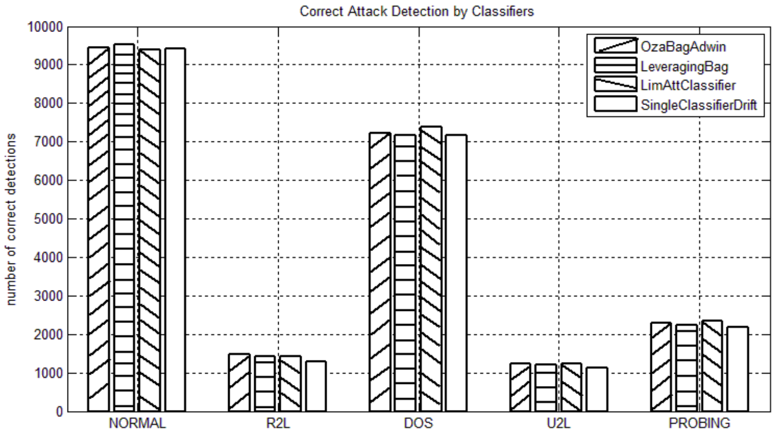

Fig. 7. Comparison among the classifiers for 5 attack types

prohibit its deployment even in other components (data concentrator and AMI head). Further analysis and improvement are also required for *OzaBagAdwin* and *LeveragingBag* to reduce their time requirement to cope with the necessary time restrictive demand for data concentrator and AMI head. Time requirement can be relaxed in smart meter as it will not be queried as frequently like other two components. However, the memory and processing ability of smart meter should be considered for designing the streaming algorithm.

7 Discussions

7.1 Comparison with Existing Works

Our work is related with Berthier *et al.* [4,3] and Zhang *et al.* [30]. However, Berthier *et al.* emphasize on specification based IDS for AMI due to AMI's controlled network and lack of training data for anomaly based IDS. They propose to use sensor based architectural scheme. On the other hand, we propose anomaly based IDS using stream data mining in network layer in OSI model as we believe that building and updating specification for AMI will be expensive eventually. For this we focus on security of individual meter. In stead of placing IDS sensors in key locations, we mention the specific deployment places of IDSs in AMI. Optimizing the sensor locations will also become a signification issue in designing IDS infrastructure.

Our work is closely related with Zhang *et al.* as they use anomaly based IDS. However, they propose IDS architecture for complete SG where we concentrate on AMI security with IDS. Moreover, we emphasize that it is more practical to apply stream mining techniques rather than static techniques in AMI where speed and time are very concerned issues.

Nonetheless, we are not able to directly compare our experimental results with those of the two methods mentioned above because we have a different (and more realistic) assumption that the data is in the form of a stream rather than static.

7.2 Evaluation of Proposed Architecture

Kush *et al.* [18] identified seven requirements in the context of an IDS for smart grids. These requirements are derived from some essential characteristics in SGs environment where IDS will be deployed. Here we try to evaluate our architecture according to those seven requirements (R1 to R7).

R1 (Support of Legacy Protocols): Our IDS architecture is not dependent on particular protocols. Hence, legacy protocols as well as additional ones can be implemented in this architecture. However, particular protocol can be developed for co-ordination among the local IDSs within each component.

R2 (Scalability): As we are mainly focusing the security of every components, like smart meter which can be treated most dynamic component in AMI, our architecture is scalable.

R3 (Support of Legacy Hardware): We show from our result that some stream mining classifiers have the ability to cope with restricted resources. So, our architecture can adapt with existing hardware with some modification like deploying the local IDS inside a particular component. However, the 'security box', we propose, can be installed externally or internally, may not be available currently. Nevertheless, we believe that such a device can be commercialized with existing technologies.

R4 (Standards Compliance): New standards require to be developed for smart meter for our architecture. We strongly believe current extreme resources constrained smart meter does not sufficient for security in both consumer and utility side. Beside this, standards should be made for our proposed 'security box'.

R5 (Adaptiveness): As we introduce M-IDS for each smart meter, any new system or device like electric vehicle, which are attached to the smart grid dynamically can be monitored. However, registering a dynamic device should maintain a secure procedure which will prevent many attacks.

R6 (Being Deterministic): As our architecture considers the traffic flow's continuity and have the ability to monitor with limited resources, dynamic traffic patterns will not reduce the IDS performance.

R7 (Reliability): It is apparent that our architecture will enrich the reliability as we consider security of each and every particular component in AMI.

8 Conclusion and Future Work

In this paper, we propose an architecture for IDS in AMI which is more reliable, dynamic, and considers the real time nature of traffic. Moreover, we simulate the data analysis of IDS in each component of AMI. Our results show, the concurrent stream mining algorithms can meet the restrictive resource requirements like memory in smart meters. In addition, we emphasize dedicated IDS along with other security measures for smart meter. The reason is this security measures will enhance the complete security, reliability, and even prevention the attacks against AMI. Several obvious issues like characteristic of traffic in AMI, coordination among the IDSs, registering dynamic device to smart meter, etc. come in focus from this work. They will help the corresponding stakeholders to pay attention to take necessary steps.

In our future work, we plan to deploy a test bed for our architecture introducing all possible attacks. This will facilitate us for generating real time training data as well as to implement and enhance the system in practice. Moreover, though the current stream mining algorithms already show some promising features like low memory consumption and time requirement for some algorithms, in our future work, we will more concentrate on developing specialized stream mining algorithms to further ameliorate the performance of our proposed IDS scheme. Thus, our works in next phase will alleviate us to understand AMI for better implementation and at the same time to improve the intrusion detection capacity for integral IDS in AMI.

Acknowledgement. This research was sponsored by the Government of Abu Dhabi, United Arab Emirates through its funding of the MIT-Masdar Institute Collaborative Research Project on "Data Mining for Smart Grids" (award number 10CAMA1).

References

1. Baena-García, M., Campo-Avila, J.D., Fidalgo, R., Bifet, A., Gavaldà, R., Morales-Bueno, R.: Early Drift Detection Method. In: 4th International Workshop on Knowledge Discovery from Data Streams (IWKDDS 2006), pp. 77–86 (2006)
2. Bai, X., Meng, J., Zhu, N.: Functional Analysis of Advanced Metering Infrastructure in Smart Grid. In: 2010 International Conference on Power System Technology (POWERCON 2010), pp. 1–4 (2010)
3. Berthier, R., Sanders, W.H.: Specification-based Intrusion Detection for Advanced Metering Infrastructures. In: 17th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2011), Pasadena, California, USA (2011)
4. Berthier, R., Sanders, W.H., Khurana, H.: Intrusion Detection for Advanced Metering Infrastructures: Requirements and Architectural Directions. In: 1st IEEE International Conference on Smart Grid Communications (SmartGridComm 2010), pp. 350–355 (2010)
5. Bifet, A., Frank, E., Holmes, G., Pfahringer, B.: Accurate Ensembles for Data Streams: Combining Restricted HoeffdingTrees using Stacking. In: 2nd Asian Conference on Machine Learning (ACML 2010), pp. 225–240 (2010)
6. Bifet, A., Holmes, G., Pfahringer, B.: Leveraging Bagging for Evolving Data Streams. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS, vol. 6321, pp. 135–150. Springer, Heidelberg (2010)
7. Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., Seidl, T.: MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering. In: JMLR Workshop and Conference Proceedings. Workshop on Applications of Pattern Analysis, vol. 11, pp. 44–50 (2008)
8. Chu, N.C.N., Williams, A., Alhajj, R., et al.: Data Stream Mining Architecture for Network Intrusion Detection. In: 2004 IEEE International Conference on Information Reuse and Integration (IRI 2004), pp. 363–368 (2004)
9. Cleveland, F.M.: Cyber Security Issues for Advanced Metering Infrasttructure (AMI). In: 2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, pp. 1–5 (2008)
10. Costache, M., Tudor, V., Almgren, M., Papatriantafilou, M., Saunders, C.: Remote Control of Smart Meters: Friend or Foe? In: 7th European Conference on Computer Network Defense (EC2ND 2011), Göteborg, Sweden (2011)
11. Data Concentrator in AMI, `http://www.meworks.net/userfile/44670/DataConcentratorforAdvancedMeteringInfrastructure(AMI)_1.pdf`
12. FitzPatrick, G.J., Wollman, D.A.: NIST Interoperability Framework and Action Plans. In: 2010 IEEE Power and Energy Society General Meeting, pp. 1–4 (2010)
13. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with Drift Detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
14. Gama, J., Sebastião, R., Rodrigues, P.: Issues in Evaluation of Stream Learning Algorithms. In: 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009), pp. 329–338 (2009)
15. Hulten, G., Spencer, L., Domingos, P.: Mining Time-changing Data Streams. In: 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001), pp. 97–106 (2001)
16. KDD Cup 1999 Data, `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`
17. Khan, M.U.: Anomaly Detection in Data Streams using Fuzzy Logic. In: 2009 International Conference on Information and Communication Technologies (ICICT 2009), pp. 167–174 (2009)

18. Kush, N., Foo, E., Ahmed, E., Ahmed, I., Clark, A.: Gap Analysis of Intrusion Detection in Smart Grids. In: 2nd International Cyber Resilience Conference (ICR 2011), pp. 38–46 (2011)
19. Li, Q., Zhao, F., Zhao, Y.: A Real-Time Architecture for NIDS Based on Sequence Analysis. In: 4th International Conference on Machine Learning and Cybernetics (ICMLC 2005), vol. 3, pp. 1893–1896 (2005)
20. Lu, Z., Lu, X., Wang, W., et al.: Review and Evaluation of Security Threats on the Communication Networks in the Smart Grid. In: 2010 Military Communications Conference (MILCOM 2010), pp. 1830–1835 (2010)
21. Massive Online Analysis, `http://moa.cs.waikato.ac.nz`
22. McLaughlin, S., Podkuiko, D., McDaniel, P.: Energy Theft in the Advanced Metering Infrastructure. In: Rome, E., Bloomfield, R. (eds.) CRITIS 2009. LNCS, vol. 6027, pp. 176–187. Springer, Heidelberg (2010)
23. Oh, S., Kang, J., Byun, Y., et al.: Intrusion Detection Based on Clustering a Data Stream. In: 3rd ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2005), pp. 220–227 (2005)
24. Open Public Extended Network Metering, `http://www.openmeter.com/`
25. Bifet, A., Gavaldà, R.: Learning from Time-Changing Data with Adaptive Windowing. In: 2007 SIAM International Conference on Data Mining (SDM 2007), Minneapolis, Minnesota, USA (2007)
26. Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavaldà, R.: New Ensemble Methods For Evolving Data Streams. In: 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009), pp. 139–148 (2009)
27. Shein, R.: Security Measures for Advanced Metering Infrastructure Components. In: 2010 Asia-Pacific Power and Energy Engineering Conference (APPEEC 2010), pp. 1–3 (2010)
28. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A Detailed Analysis of the KDD CUP 99 Data Set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2009), pp. 1–6 (2009)
29. Zhang, Q., Huang, W.: Research on Data Mining Technologies Appling Intrusion Detection. In: 2010 IEEE International Conference on Emergency Management and Management Sciences (ICEMMS 2010), pp. 230–233 (2010)
30. Zhang, Y., Wang, L., Sun, W., et al.: Distributed Intrusion Detection System in a Multi-Layer Network Architecture of Smart Grids. IEEE Transactions on Smart Grid 2, 796–808 (2011)