Trust Assessment on Streaming Data: a Real Time Predictive Approach

Tao Peng, Sana Sellami, and Omar Boucelma

Aix-Marseille Univ, Université de Toulon, CNRS, LIS,
Laboratoire d'Informatique et Systèmes, UMR 7020, Marseille, France
{tao.peng,sana.sellami,omar.boucelma}@univ-amu.fr

Abstract. IoT data, that most often carry a temporal dimension, can be exploited from an analysis perspective or from a forecasting one. In this paper, we propose a predictive approach to address the problem of *data trustworthiness* in a data stream generated by a Smart Home application. We describe an online *Ensemble Regression* model that performs prediction in assigning a trust score to a target temporal value in real-time. Experiments conducted with data retrieved from the UCI ML repository demonstrate the performance of the model, while assessing data accuracy.

Keywords: data trustworthiness · smart home · data stream.

1 Introduction

Among the large spectrum of IoT applications, time-series data generated by a set of sensors and actuators are integrated to form a data stream. Smart Homes are probably the trendiest domain where data stream can be exploited in different ways such as remote control of home appliances, or even securing a house, assuming the data is reliable. Unfortunately, like any data gathered from hardware devices, sensor data stream may rise quality issues such as inaccuracy or incompleteness [21], leading to difficulties in a decision making process. Within this landscape, trusting the data is a key issue for helping stakeholders involved in such process.

Trust can be handled through the concept of *Data Trustworthiness* (DT) for which there is no unified definition in the literature: for example, [16] considers that DT assessment should be consistent with quality dimensions such as accuracy, timeliness and completeness; [18, 28] highlights accuracy as a DT evaluation while [1] emphasizes on subjectivity and accuracy.

In this paper, we consider accuracy as the main quality dimension for assessing data trustworthiness in a Smart Home application, assuming that data arrives on time and the data is complete. Data accuracy, which refers to the correctness of sensor measurements [21], has been recognized as the most important dimension in several papers [1, 16, 18, 28]. It is worth noticing that accuracy is an objective description while DT is a subjective estimation based on some assumptions (i.e., data follows a specific probability distribution). Considering

the accuracy dimension and subjectivity, we borrow DT definition from [1] that is: *"Data Trustworthiness in IoT Networks is the subjective probability that data observed by a user is consistent with the data at the source"*. Note that this definition is generic enough that leaves the door open to several implementations, depending on the context and on the probability distribution(s) one may adopt.

The remainder of the paper is organized as follows: Section 2 reviews some related work. In Section 3, we describe our approach. Section 4 illustrates the experimental results. Finally, we conclude and present some perspectives in Section 5.

2 Related Work

DT can be assessed by means of **data similarity** such as in [13] where authors propose a pattern-wise method: a target (sensor) value is considered as reliable if it co-occurs more frequently with the value of its neighbor sensor. However, this method is rather suitable for value states (such as 0/1 represents whether it is raining) than for continuous values (such as temperature). Won *et al.* [28] consider that if multiple sensors measure the same value of interest at different indoor locations, the difference between the measured values is proportional to the distance between sensors. DT is inversely proportional to the weighted sum of the difference between test data and neighbor sensor values: the smaller is the distance between sensors, the greater is the weight.

All the above works [13, 28] make the same assumption that *similar/redundant data support each other for gaining trust*. But there aren't always redundant sensors in a smart home: for example, there may be only one humidity sensor per room.

Provenance-based methods rely on different **data lineage dimensions**. In [6], inter-dependency between five items is considered: (a) data similarity, (b) data conflict, (c) path similarity, (d) data deduction, (e) provider reputation. Authors propose an iterative process for computing a trust score: at each iteration, the trustworthiness of data and provider is adjusted according to the above five elements. This work is extended by Wang *et al.* [27] in integrating the user's feedback: data received by the user come with a 'reported trust', and the user will provide its 'adjusted trust' after accepting the data. If the difference between 'reported trust' and 'adjusted trust' is too large, the provider's reputation decreases. Lim *et al.* [22] also extended work of [6] in providing a cyclic trust computation framework suitable for data streams: (a) the more trusted data reported by the sensor, the higher is the (provider's) reputation; (b) data trust depends both on data similarity, provenance similarity and sensor reputation.

The idea behind the provenance-based approach [6, 22, 27] is the same: *the more a data has similar redundant data with different lineages, the more this data is trusted*. However, a Smart Home is often an Ad Hoc network [23] where there is a unique data lineage from the sensor to the gateway [21].

More recent works [1, 15] promote **regression based methods**. In [15] a static city weather data set is analysed: authors propose a method that estimates

the value of a target sensor by means of the values of its surrounding sensors. If the residual between the estimation and the real value exceeds a predefined threshold, then the (target) value is considered as untrusted, the residual being the difference between the predicted value and the real value. Adams *et al.* [1] revisit the work of [15] in considering that the residual follows a Gaussian distribution. A Cumulative Distribution Function takes the residual as input and outputs a trust score: if this score exceeds a threshold, the received data is trusted. Work in [1] shows that Linear Regression outperforms Random Forest Regression, Gradient Boosted Machine and Multi-Layer Perceptrons.

These works [1, 15] share the idea that *a small residual (i.e., the model made a good prediction) leads to a high trust score.*

We found the approach described in [1] appealing although it does not take into account data stream characteristics (timeliness, non-stationarity, etc.). Especially, due to seasonal changes, or changes in user habits, the underlying distribution parameters (e.g., means, variance, correlation) of smart home data usually changes over time, which is called the non-stationarity feature of the data stream [26]. Non-stationarity of the data stream leads to a significant degradation of the performance of the prediction/classification model, which is known as concept drift. Although the work of [1] does not take into account the non-stationarity of data stream and the concept drift, we believe it is a good start assuming we could transpose it to target (IoT) data streams.

In the next section, we describe DTOM, a Data Trustworthiness Online Method to evaluate a trust score of (a batch of) data in a real-time data stream. DTOM is based on the work [1] but differs by the following points: (1) DTOM is based on an Online Ensemble Regression model which is suitable for the analysis of online streams; (2) DTOM has a heuristic update strategy: Updated using the data from the top 50% of trust rankings per batch, and (3) DTOM has been evaluated with various real inaccurate data ratios while [1] use a (simulated) inaccurate fixed data ratio.

3 Data Trustworthiness Online Model: DTOM

In this section, we first provide a problem statement as well as algorithmic details, then we describe the Online Ensemble Regression methods we adopted.

3.1 Problem Statement

Given f sensors, each sensor generates a value within a fixed period of time. A value $d_{f',t'}$ is emitted by a sensor f', at time t'. If $d_{f',t'}$ has quality (accuracy dimension) issue, its accuracy level $da_{f',t'}$ is 0, otherwise, it is 1. Our model will give an estimated DT $dts_{f',t'} \in [0,1]$ (denoted as a Trust Score) by Equation 1 from [1]. Estimation of $dts_{f',t'}$ is the solution to problem minimizing $|da_{f',t'} - dts_{f',t'}|$. So, the problem of assessing DT can be considered as a **Prediction** problem.

$$dts_{f',t'} = F(dr_{f',t'}, \mu, \sigma) = \begin{cases} \frac{2}{\sigma\sqrt{2\pi}} \int_{-\infty}^{dr_{f',t'}} EXP\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx, \text{if } dr_{f',t'} < \mu \\ \frac{2}{\sigma\sqrt{2\pi}} \int_{0}^{dr_{f',t'}} EXP\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx, \text{if } dr_{f',t'} \geq \mu \end{cases} \quad (1)$$

3.2 Design and Implementation

DTOM approach consists of three processes: initialization (offline phase), assessment (online phase) and update (online phase).

Initialization: Given a sensor f', its historic data is noted as $Y_{f'}$, and its reference data (gathered from other sensors) is noted as $X_{f'}$. $Y_{f'}$ and $X_{f'}$ are used to initialize the ensemble Regression model (line 1, Algo. 1). We calculate the estimation $\hat{d}_{f',t'}$ of each historical data $d_{f',t'}$. Then, we calculate the residual $dr_{f',t'}$ between $\hat{d}_{f',t'}$ and $d_{f',t'}$ (line 2). The average (resp. standard deviation) of the residual is denoted μ (resp. σ) (line 3, 4).

Algorithm 1 DTOM Initialization

Input: historic data of a sensor f', $Y_{f'}$; reference data of $Y_{f'}$, $X_{f'}$;
Output: the ensemble regression model *Reg*; the average of residuals, μ; the standard deviation of residuals, σ.
1: an ensemble regression model *reg* is initialized with $Y_{f'}$ and $X_{f'}$.
2: *setResiduals* ← the training error (residual) of *Reg* with $Y_{f'}$ and $X_{f'}$
3: μ ← average of *setResidual*
4: σ ← Standard deviation of *setResidual*
5: **Return** μ, σ, *Reg*;

Assessment: One data $d_{f',t'}$ arrives at a processing device (e.g., gateway) as defined in Algorithm 2 (lines 2 - 5). The ensemble regression generates an estimation $\hat{d}_{f',t'}$ (line 3) and gets the corresponding value of residual (line 4). The corresponding trust score $dts_{f',t'}$ is calculated by equation 1 from [1] (line 5).

Update: the new data $d_{f',t'}$ is also buffered, with its reference data $ref_{f',t'}$ and its trust score $dts_{f',t'}$ (line 6 in Algorithm 2). When the buffer is full (lines 7 - 18), the data from the top 50% of trust rankings in the buffer is used to update the regression (line 8), and the buffer is cleared (line 9).

3.3 Online Ensemble Regression

Online Ensemble Regression methods are suitable to our context especially for handling concept drifts [8, 10, 20]. Online Ensemble Regression is a set of individual regression models whose predictions are combined to predict new incoming

Algorithm 2 DTOM Assessment and Update

Input: New data from sensor f' at time t', $d_{f',t'}$; the reference data of $d_{f',t'}$, $ref_{f',t'}$; ensemble regression model, *Reg*, *Reg.predict* is the prediction function of *Reg*; the residual follows a Gaussian Distribution, $N\left(\mu, \sigma^2\right)$; A buffer is used to store the data in each batch, and the upper limit of its capacity is also equal to the batch size, noted as $buffer Size$.

Output: trust score of $d_{f',t'}$, $dts_{f',t'} \in [0, 1]$.

$myBuffer \leftarrow \phi$ // The buffer cache is empty
if new data $d_{f',t'}$ is generated **then**
    $\hat{d}_{f',t'} \leftarrow Reg.predict(ref_{f',t'})$ // generates an estimation of $d_{f',t'}$
    $dr_{f',t'} \leftarrow d_{f',t'} - \hat{d}_{f',t'}$ // get the residual
    $dts_{f',t'} = F(dr_{f',t'}, \mu, \sigma)$ // as Equation 1
    $myBuffer \leftarrow myBuffer \cup (d_{f',t'}, ref_{f',t'}, dts_{f',t'})$ //new data, its reference data and its trust score are buffered
    if $|myBuffer| \geq buffer Size$ **then** // when the buffer is full
        $Reg \leftarrow Reg$ update with the data from the top 50% of trust rankings in $myBuffer$
        $myBuffer \leftarrow \phi$ //the buffer is cleared
    end if
end if
Return $dts_{f',t'}$;

instances in real time. There are several online regression models in the literature [2–4, 7, 12, 14, 17, 19, 24, 25, 29].

Online ensemble regression methods may adopt the following strategies to accommodate concept drift (the strategies chosen for each model are shown in Table 1):

- M1) Modification of basic models' weights: The better the performance of the basic model in the latest data, its voting weight increases, otherwise the weight decreases.
- M2) Modification of basic models' parameters: If the basic model is updatable, its parameters are adjusted with new data.
- M3) Modification of basic models' parameters: If the loss of the entire ensemble regressions exceeds a threshold, new basic models are added to improve performance.
- M4) Modification of basic models' parameters: Poorly performing or too old basic models are removed to reduce the computational burden.
- A1) Selecting instances: Incorrectly predicted data is used to update the model because it may represent the trend of data changes.
- A2) Weighting instances: Incorrectly predicted data gets more weight that affect the model update.

As illustrated in Table 1, Online Ensemble Regression methods can be updated in 1) using a single piece of data (denoted "simple") or 2) multiple pieces (denoted "batch"). In terms of "explanatory", Online Ensemble Regression methods can be divided into two categories [25]: 1) implicit, online regression

model does not use detection techniques of a concept drift, but is continuously updated with new data; 2) explicit, the update mechanism is triggered only when the concept drift is confirmed by the concept drift detection module. Some Online Ensemble Regression methods use a sliding window, while others don't (see table 1). Most methods limit the number of base models except for [29].

In order to choose an Ensemble method, we adopted the following criteria (models that meet the criteria are marked with * in Table 1):

- **No re-accessible historic data** is one of the main differences between data streams and static data [20]. Data is accessed only once and then discarded to limit memory and storage space usages [9].
- **Batch-by-Batch update** has better stability than instance-by-instance [3, 8] and is less sensitive to inaccurate data [3].
- **Implicit method** is more suitable than the explicit one (such as concept detection) in noisy data streams [20], because the latter may cause too many false alarms [8, 20].
- **Limited number of basic models** reduce the storage burden [20].

As illustrated in Table 1, **AddExp** [19] and **B-NNRW** [3] are the methods that meet our criteria. **AddExp** uses a loss bound to obtain the error model, and adjusts the expert's weights according to their actual losses (M1). Each expert updates upon new arrival data (M2). If the overall performance (loss bound) is below (above) a predefined threshold, a new expert is added (M3). The pruning strategy is weakest-first or oldest-first (M4).

Note that the original version of AddExp was designed to update instance-by-instance, but AddExp can be easily extended to update Batch-by-Batch [25]. The original AddExp does not reveal which instances should be taken for training a new basic model [5]. However, for "Batch-by-Batch update", it can train / initialize a new basic model by Boosting / Bagging the instances in the current batch (A1,A2). One limitation of AddExp is that its predictions are in [0, 1] interval. Another limitation is that it depends on a number of hyper-parameters, as follows: 1) factor of decreasing weight β: the weight of basic model is updated as $\omega_{t+1,i} = \omega_{t,i}\beta^{|\xi_{t,i}-y_t|}$; 2) loss required to add new expert τ: if $|\hat{y}_t - y_t| > \tau$, a new expert is added; 3) factor of new expert weight γ: the weight of the new basic model is equal to $\gamma \sum_{i=1}^{N_t} \omega_{t,i} |\xi_{t,i} - y_t|$. Where, $\omega_{t,i}$ is the weight of basic model i in time t; y_t is the dependent variable; $\xi_{t,i}$ is the estimation of y_t by basic model i; N_t is number of overall experts; $\hat{y}_t$ is the estimation of AddExp (over all basic models).

B-NNRW, a Boosting/Bagging ensemble method is based on NNRW (A1,A2), a Neural Network with Random Weights where the weights between the input layer and the hidden layer are fixed. NNRW does not update and adopts a linear assumption. Therefore, B-NNRW also adopts the linear assumption and adjusts its weights according to their loss in the last batch of data (M1). Pruning (M3) and adding (M4) basic models are also used to maintain the performance of the whole system. B-NNRW also relies on some hyper parameters such as 1) the pruning rate q: only Q models with the lowest error are eligible to participate in

method	description	new data size	explanatory	Sliding window	adaption	# basic model
***AddExp** [19]	Additive expert ensembles regression	*both	*implicit	*no	M1, M2, M3, M4	*limited
ILLSA [17]	Incremental Local Learning Soft Sensing Algorithm	*batch	*implicit	yes	M1, M3, M4	*limited
OWE [25]	On-line Weighted ensembles regression	simple	*implicit	yes	M1, M3, M4, A1	*limited
R-FIMT-DD [14]	Ensemble of Incremental Hoeffding-based trees	simple	explicit	*no	M2, A1	*limited
AMRules [2]	Ensemble of randomized adaptive model rules	simple	explicit	*no	M3, M4	*limited
DOER [24]	Dynamic and Online Ensemble Regression	simple	*implicit	yes	M1, M2, M3, M4	*limited
VHPRE [4]	Vertical and Horizontal Partitioning for Data Stream Regression Ensemble	simple	explicit	*no	M1, M3, M4	*limited
ARF-Reg [12]	Adaptive Random Forest (ARF) for regression	simple	explicit	*no	M2, M3, M4	*limited
Online-DNNE [7]	Neural network ensembles with random weights based	simple	*implicit	*no	M2	*limited
***B-NNRW** [3]	Neural network ensembles with random weights + Bagging / Boosting	*batch	*implicit	*no	M1, M3, M4, A1, A2	*limited
Learn.++ R2C [29]	Learn++.NSE [8] + R2C	*batch	*implicit	*no	M1	not limited

Table 1. Online Regression Methods with their Characteristics

the final prediction, $Q = q * M$, M being the maximum number of basic models ; 2) the replacement rate r: the number of new added models is $r * M$.

4 Experimentation

4.1 Experimental Dataset

We conducted our experiments with the Appliances Energy Prediction dataset retrieved from the UCI Machine Learning Repository data portal [1] consisting of the following attributes: energy assumption, humidity and temperature. For illustration purposes, we focus on the *RH2 sensor* which is a humidity sensor in a living room area.

Dataset volume and velocity: One humidity sensor and one temperature sensor are installed in each room and outside the building (18 sensors in total). Data were averaged for 10 minutes period and gathered during 4.5 months (from 11/01/2016 to 05/27/2016) resulting in a total 12 MB CSV file with 19735 instances.

Simulated untrusted data (SUTD): Variance Fault (Gaussian noise) is one of several types of faults that can be injected into a data stream (randomly selected original data) to represent untrustworthy data [11]. [1] shows that the detection of Variance Fault is more difficult than others, such as Stuck Fault (replaces the true data value with a constant value), Offset (adds an a constant value to the true data value). The percentage of noisy data injected into original data (OD) varies from 5% to 65% (by step of 5%) for each experiment. Based on [1], we define SUTD as follows: $SUTD = OD + N(0, \delta')$, where N is a Gaussian distribution and δ' is the Standard Deviation of RH2 sensor data. Due to space limitations, Fig. 1 (b-e) shows RH2 data for the first 24 hours, respectively without and with 5%, 35% and 65% of noise.

RH2 sensor data: Fig. 1(a) displays RH2 data with some concept drifts detected by Page-Hinckley Test [2]. We observe that from January to March, data changes are relatively flat compared to April, May. Correspondingly, the concept drift from January to March is less visible than for April, May. This smart home sensor data with non-stationary nature (concept drift) will be used to test whether DTOM can handle the concept drift to correctly assess DT in the non-stationary data stream.

Reference Data: For RH2 sensor, the reference dataset is the data sent from other 17 humidity / temperature sensors (not including energy assumption data), and these sensors always generate correct data. Due to space limitations, Fig. 2 (a-d) shows an excerpt of RH6, T6, RH5, T5 sensor data with their statistical description.

Root Mean Square Error (RMSE) (Equation 2) is a known measure that we use to assess OD trust. The lower the RMSE value of ODs / SUTDs, the more accurately their trust scores are estimated.

[1] UCI https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction

[2] Details about Page-Hinkley method for concept drift detection are available at https://scikit-multiflow.github.io/scikit-multiflow/

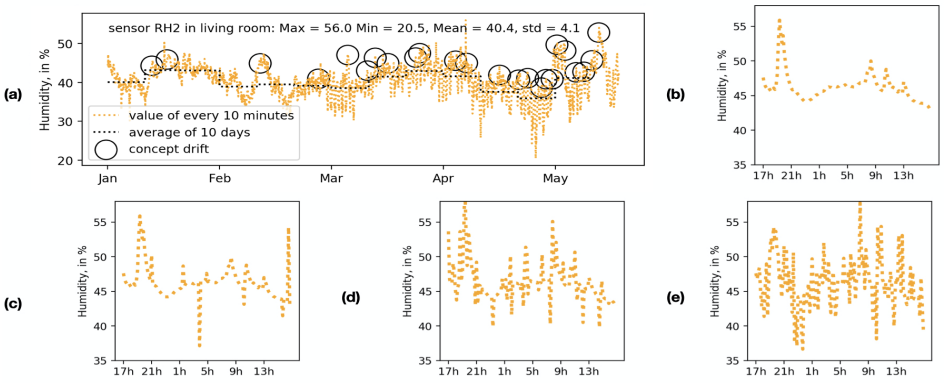

Fig. 1. (a) RH2 data in living room, from January to May, 2016. Some concept drifts are detected by Page-Hinckley Test, illustrated in circles. The first 24 hours data in RH2 (b) without SUTD; with (c) 5% of SUTD; (d) 35% of SUTD; (e) 65% of SUTD.

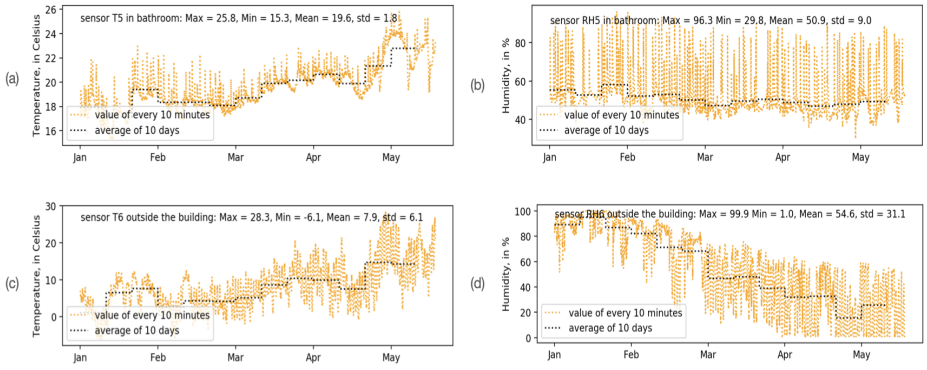

Fig. 2. (a) Temperature sensor RH5 and (b) Humidity sensor T5 in bathroom; (c) Temperature sensor RH6 and (d) Humidity sensor T6 outside the building.

$$RMSE = \begin{cases} \sqrt{\frac{1}{|ODs|}\sum_{d_{f',t'} \in ODs} (dts_{f',t'} - 1)^2}, \text{ for ODs} \\ \sqrt{\frac{1}{|STUDs|}\sum_{d_{f',t'} \in STUDs} (dts_{f',t'} - 0)^2}, \text{ for STUDs} \end{cases} \quad (2)$$

Balanced-Accuracy (BACC). To further evaluate DTOM, data are classified either as trustworthy or untrustworthy according to a threshold tth. Let us set up: an OD is seen as a **true positive (TP)** if it is correctly classified as 'trustworthy' and a **false negative (FN)** otherwise, and that a SUTD is seen as a **true negative (TN)** if it is correctly identified as 'untrustworthy' and a **false positive (FP)** otherwise. In this case, $BACC$ (equation 3) indicates whether the overall data is well classified and takes into account the unbalanced nature of the dataset.

$$BACC = (Sensibility + Specificity)/2$$
$$\text{where } Sensibility = \frac{\#TP}{\#TP + \#FN} \text{ and } Specificity = \frac{\#TN}{\#TN + \#FP} \quad (3)$$

Trust Score We can also directly observe the trust score of ODs / SUTDs to determine whether they are correctly scored when the concept drift occurs. The expected trust score for any OD is 1. Therefore, in the case of concept drift, the higher (more accurate) of ODs' trust score, DTOM adapts better to the concept drift. Similarly, the expected trust score for any SUTD is 0. In the case of concept drift, the lower (more accurate) SUTDs' trust score, DTOM adapts better to the concept drift.

4.2 Evaluation

In order to evaluate DTOM, we implemented AddExp and B-NNRW. We also compared DTOM with linear regression (a static model), to show how DTOM behaves in presence of concept drift. For any regressor, the first 5% data are used for initialization. Trust threshold tth is determined by maximizing BACC. For any Online Ensemble Regression: the maximum number of basic models is 25; instances are weighted by Boosting; buffer size is 100. The super-parameters for each regressor are as follows:

- *AddExp*: factor of decreasing weight $\beta = 0.5$, factor of new expert weight $\gamma = 0.1$, loss required to add new expert $\tau = 0.05$ (See definitions in Section 3.3). These super-parameter settings are the optimal values after tuning, i.e., the same settings suggested in [19]. Basic regression models are SGD-Regressor and Passive-Aggressive-Regressor[3]. Pruning strategy is the worst first [3].

[3] available in Sklearn: https://scikit-learn.org/stable/

– *B-NNRW*: Number of hidden nodes of NNRW is 16; the pruning rate $p = 0.9$ (optimal value between 1.0 and 0.7); the replacement rate $r = 0.1$ (optimal value between 0.0 and 0.3) (See definitions in Section 3.3).
– *Linear Regression*: the first 5% data are taken for initialization; the trust threshold *tth* is determined by maximizing BACC, but without update.

4.3 Results

In this subsection, we will discuss the numbers obtained for RMSE (trust score's accuracy) and BACC (OD / SUTDs' classification) for all the above methods.

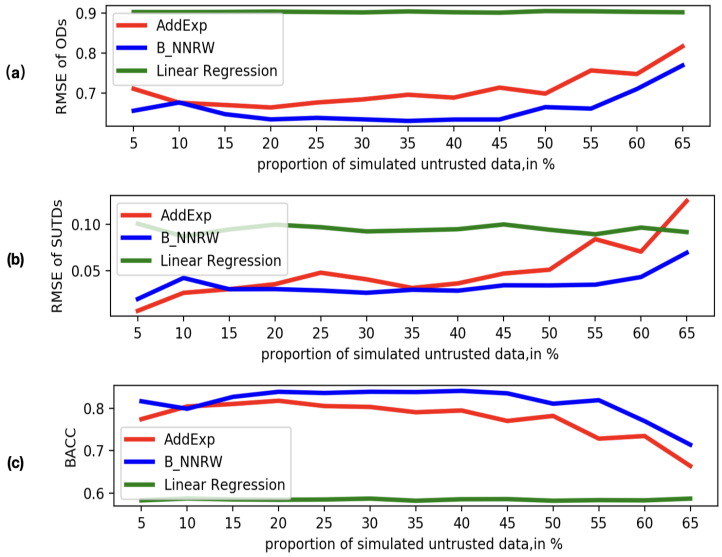

Fig. 3. (AddExp vs. B-NNRW vs. Linear Regression) performance with different SUTD ratios(5% - 65%) in RH2 data: (a) RMSE of ODs' trust score; (b) RMSE of STUDs' trust score; (c) BACC of overall data.

RMSE of ODs : As depicted in Fig. 3(a), the linear regression curve is close to 0.9 for all different SUTD ratios. This means that with linear regression, OD is always wrongly evaluated with a relatively low trust score. The reason is that the residuals between ODs and their prediction are unexpectedly too large. A further explanation is that linear regression without updates cannot maintain predictive power in non-stationary data streams, due to concept drift.

Note that for all SUTD ratios, B-NNRW curve is always lower, and therefore better than AddExp. One possible explanation is that, for non-stationary data stream, the prediction ability of B-NNRW is better than AddExp one.

For a SUTD ratio in the 5%-45% range , both B-NNRW and AddExp curves are stable. In other words, B-NNRW and AddExp maintain their performance as data quality declines. The reason is that 1) DTOM has successfully filtered out most of low-quality data that is not used to update the Ensemble Regression models, 2) B-NNRW and AddExp both have a certain tolerance for inaccurate data.

However, when the SUTD ratio exceeds 50%, B-NNRW and AddExp curves increase. This is because they both are updated by using the data from the top 50% of trust rankings in each batch. If the SUTD ratio is close to or higher than 50%, SUTDs inevitably interfere with its update process.

RMSE of SUTDs : Fig. 3(b) shows that, for all SUTD ratios, SUTDs' RMSE of Linear Regression is close to 0.1.

Fig. 3(b) also shows that AddExp and B-NNRW performances are stable when the SUTD ratio is within a 5% to 50% range, and B-NNRW ratio (which is close to 0.02) is slightly lower (better) than AddExp (close to 0.03). Both AddExp and B-NNRW behave better than linear regression.

When the proportion of SUTD is greater than 50%, AddExp curve increases significantly. This means that AddExp loses performance: it is even worse than Linear Regression. However, B-NNRW curve increases more slowly than AddExp. One possible explanation is: 1) AddExp loses its predictive ability due to updating with some SUTDs; 2) B-NNRW has a higher tolerance than AdddExp for SUTD, and its prediction ability is less negatively impacted.

BACC : Fig. 3(c) shows that, in all cases, Linear Regression BACC is stably close to 0.55, which is lower than others. This means that nearly half of the data is correctly classified.

For all SUTD ratio range values, we have shown that B-NNRW performs better than AddExp, in comparison with ODs' RMSE and of STUDs' RMSE. Therefore, BACC of B-NNRW is always higher (better) than AddExp. This means that, a higher percentage of data is correctly classified with B-NNRW than with AddExp.

From the 50%-65% SUTDs ratio, we showed that B-NNRW and AddExp have lost performance in both ODs' RMSE and STUDs' RMSE, due to unavoidable update with SUTDs. Therefore, as the SUTD ratio increases from 50%, both B-NNRW and AddExp BACC values decrease.

Trust Score of ODs. We have shown that 1) when SUTDs ratio does not exceed 50%, the performance of B-NNRW and AddExp are stable; 2) with any SUTDs ratio (5% - 65%), the performance of Linear Regression is stable.

For illustration purposes, we arbitrarily choose a ratio of 25% SUTDs from 0% to 50%, and illustrate the ODs' trust score generated by all regressors, as illustrated in Fig. 4 (a). Before concept drift No.1, the linear regression curve is even higher (better) than B-NNRW and AddExp ones. However, after concept drift No.1, the linear regression curve is always close to 0.1, which is far from the expected value of 1 for ODs. The reason is that the concept drift affects the

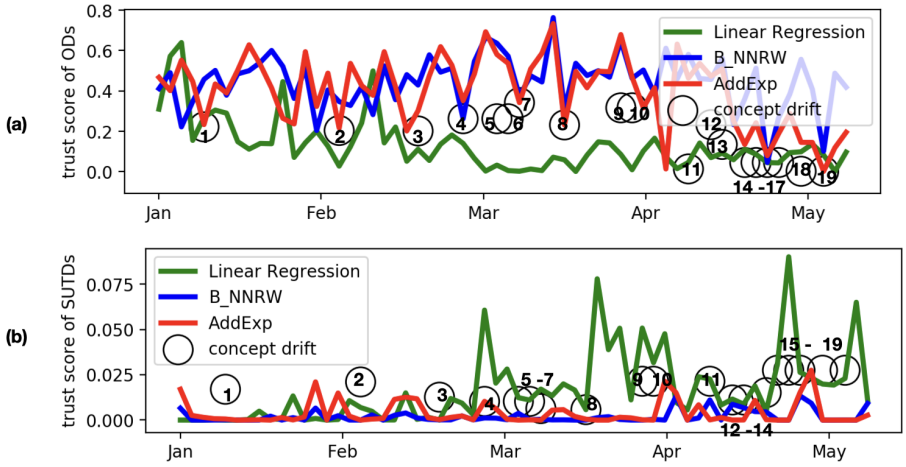

Fig. 4. (AddExp vs. B-NNRW vs. Linear Regression) performance with RH2 data (25% SUTD ratios), in case of concept drift: (a) trust score of ODs; (b) trust score of SUTDs. Up to 19 concept drifts were detected by Page-Hinckley Test (illustrated by numbers and circles).

performance of Linear Regression because this method does not handle concept drifts.

By observing Fig. 4 (a), both of B-NNRW or AddExp follow a downward trend due to the concept drift. The decline of the curve means that performance is reduced. However, after the performance degradation, the curves of both B-NNRW and AddExp tend to return to the previous level. This ability comes from the update process of Online Ensemble Regression, which enables DTOM to deal with the concept drift.

In comparing B-NNRW with AddExp curves, we note that 1) when there is no concept drift, the curves of both are closed; 2) when the concept drift occurs, the curve of B-NNRW declines slightly than the AddExp one: this is illustrated in Fig. 4 (a) concept drifts 1 - 3, 5 - 8, 10 - 19). This means that B-NNRW can adapt to changes in the data stream more quickly than AddExp, and outperforms AddExp (the same result is shown in Fig. 3 (a) with 25% STUDs ratio).

Trust Score of SUTDs We still choose the 25% SUTDs ratio to illustrate the SUTDs' trust score generated by all regressors (see Fig. 4 (b)).

Before concept drift No.1, the linear regression curve is lower (better) than B-NNRW and AddExp. However, after concept drift No.1 and before the concept drift No.2, the linear regression's curve increase. After the concept drift No.2 , the linear regression curve is slightly higher than B-NNRW and AddExp. After the concept drift No.3, the linear regression curve is significantly higher (worst) than B-NNRW and AddExp. The reason is same as ODs' trust score of linear regression: we do not have an update mechanism to deal with a concept drift.

By observing Fig. 4 (b), most of the concept drifts lead to a relatively slight increase in the curves of B-NNRW and AddExp (degraded performance). However, thanks to the update capacity of the online ensemble regression, upon performance decline, the curves of B-NNRW and AddExp tend to return to the previous level (close to 0).

Our experiments show : 1)when concept drift occurs relatively at low frequency (concept drifts No. 1 - 8 , in Jan., Feb. and Mar.), B-NNRW has a slight advantage over AddExp (B-NNRW 0.001 vs. AddExp 0.003 in mean); 2) When the frequency of concept drift occurs at a higher frequency (concept drifts No. 10 - 19, in Apr. and May), the curves of both increase (worst), but B-NNRW keeps its advantage over AddExp (B-NNRW 0.003 vs. AddExp 0.004 in mean). Hence, the overall performance of B-NNRW is better than AddExp (the same result has been shown in Fig. 3 (b) with 25% STUDs ratio).

5 Conclusion

In this article we described DTOM, an online model-based method for assessing data trustworthiness in smart home (IoT) data streams. DTOM extends the work of [1] in using Online Ensemble Regression, and in adopting a heuristic update strategy: batch-by-batch, with the data from the top 50% of trust rankings in each batch. DTOM has been implemented with B-NNRW and AddExp and experimental results have been conducted with a real dataset.

The first outcome of the experimentation is that B-NNRW ensures Data Trustworthiness for a vast majority of data in a non-stationary data stream, while outperforming other regressors. The second outcome relates to DTOM performance degradation when the SUTD ratio exceeds 50%, because SUTDs will inevitably interfere with the regressor update process.

The work described in this paper is a first step towards developing efficient real-time predictive methods for a data stream, i.e., the proposal of learning methods that (1) can handle the drifts, and (2) cover a comprehensive set of practical applications. However, the proposed methods have some limitations. Indeed, our work is based on the assumption that the initialization phase has a high-trust dataset. If a low-trust dataset is used during the initialization phase, it is possible that 1) the distribution parameters of residuals may be incorrectly estimated; 2) the parameters of the online Ensemble Regression model also may be erroneous. Clearly there is room for improving these methods. One possible research direction is that that our proposed method requires only a small amount of high-trust data for initialization. This amount may be provided by domain experts at limited cost.

Bibliography

[1] Adams, S., Beling, P.A., Greenspan, S., Velez-Rojas, M., Mankovski, S.: Model-based trust assessment for internet of things networks. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). pp. 1838–1843. IEEE (2018)

[2] Almeida, E., Ferreira, C., Gama, J.: Adaptive model rules from data streams. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 480–492. Springer (2013)

[3] de Almeida, R., Goh, Y.M., Monfared, R., Steiner, M.T.A., West, A.: An ensemble based on neural networks with random weights for online data stream regression. Soft Computing pp. 1–21 (2019)

[4] Barddal, J.P.: Vertical and horizontal partitioning in data stream regression ensembles. In: 2019 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2019)

[5] Barddal, J.P., Gomes, H.M., Enembreck, F.: Advances on concept drift detection in regression tasks using social networks theory. International Journal of Natural Computing Research (IJNCR) **5**(1), 26–41 (2015)

[6] Dai, C., Lin, D., Bertino, E., Kantarcioglu, M.: An approach to evaluate data trustworthiness based on data provenance. In: Workshop on Secure Data Management. pp. 82–98. Springer (2008)

[7] Ding, J., Wang, H., Li, C., Chai, T., Wang, J.: An online learning neural network ensembles with random weights for regression of sequential data stream. Soft Computing **21**(20), 5919–5937 (2017)

[8] Elwell, R., Polikar, R.: Incremental learning of variable rate concept drift. In: International workshop on multiple classifier systems. pp. 142–151. Springer (2009)

[9] rez Gallego, S.R.x., Krawczyk, B., García, S., Woźniak, M., Herrera, F.: A survey on data preprocessing for data stream mining: Current status and future directions. Neurocomputing (2017)

[10] Gama, J.: Knowledge discovery from data streams. CRC Press (2010)

[11] Ganeriwal, S., Balzano, L.K., Srivastava, M.B.: Reputation-based framework for high integrity sensor networks. ACM Transactions on Sensor Networks (TOSN) **4**(3), 1–37 (2008)

[12] Gomes, H.M., Barddal, J.P., Ferreira, L.E.B., Bifet, A.: Adaptive random forests for data stream regression. In: ESANN (2018)

[13] Gwadera, R., Riahi, M., Aberer, K.: Pattern-wise trust assessment of sensor data. In: 2014 IEEE 15th International Conference on Mobile Data Management. vol. 1, pp. 127–136. IEEE (2014)

[14] Ikonomovska, E., Gama, J., Džeroski, S.: Online tree-based ensembles and option trees for regression on evolving data streams. Neurocomputing (2015)

[15] Javed, N., Wolf, T.: Automated sensor verification using outlier detection in the internet of things. In: 2012 32nd International Conference on Distributed Computing Systems Workshops. pp. 291–296. IEEE (2012)
[16] Jayasinghe, U., Otebolaku, A., Um, T.W., Lee, G.M.: Data centric trust evaluation and prediction framework for iot. In: 2017 ITU Kaleidoscope: Challenges for a Data-Driven Society (ITU K). pp. 1–7. IEEE (2017)
[17] Kadlec, P., Gabrys, B.: Local learning-based adaptive soft sensor for catalyst activation prediction. AIChE Journal **57**(5), 1288–1301 (2011)
[18] Karthik, N., Ananthanarayana, V.: Data trust model for event detection in wireless sensor networks using data correlation techniques. In: 2017 fourth international conference on signal processing, communication and networking (ICSCN). pp. 1–5. IEEE (2017)
[19] Kolter, J.Z., Maloof, M.A.: Using additive expert ensembles to cope with concept drift. In: Proceedings of the 22nd international conference on Machine learning. pp. 449–456 (2005)
[20] Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: A survey. Information Fusion **37**, 132–156 (2017)
[21] Leonardi, A., Ziekow, H., Strohbach, M., Kikiras, P.: Dealing with data quality in smart home environments—lessons learned from a smart grid pilot. Journal of Sensor and Actuator Networks **5**(1), 5 (2016)
[22] Lim, H.S., Moon, Y.S., Bertino, E.: Provenance-based trustworthiness assessment in sensor networks. In: Proceedings of the Seventh International Workshop on Data Management for Sensor Networks. pp. 2–7 (2010)
[23] Lin, H., Bergmann, N.W.: Iot privacy and security challenges for smart home environments. Information **7**(3), 44 (2016)
[24] Soares, S.G., Applications, R.A.E.S.w., 2015: A dynamic and on-line ensemble regression for changing environments. Elsevier (2015)
[25] Soares, S.G., Araújo, R.: An on-line weighted ensemble of regressor models to handle concept drifts. Engineering Applications of Artificial Intelligence **37**, 392–406 (2015)
[26] Tran, L., Fan, L., Shahabi, C.: Outlier detection in non-stationary data streams. In: Proceedings of the 31st International Conference on Scientific and Statistical Database Management. pp. 25–36. ACM (2019)
[27] Wang, X., Govindan, K., Mohapatra, P.: Provenance-based information trustworthiness evaluation in multi-hop networks. In: 2010 IEEE Global Telecommunications Conference GLOBECOM 2010. pp. 1–5. IEEE (2010)
[28] Won, J., Bertino, E.: Distance-based trustworthiness assessment for sensors in wireless sensor networks. In: International conference on network and system security. pp. 18–31. Springer (2015)
[29] Xiao, J., Xiao, Z., Wang, D., Bai, J., Havyarimana, V., Zeng, F.: Short-term traffic volume prediction by ensemble learning in concept drifting environments. Knowledge-Based Systems (2018)