

# A Gamification Framework for Sensor Data Analytics

Alexandra L'Heureux, Katarina Grolinger, Wilson A. Higashino, Miriam A. M. Capretz  
Department of Electrical and Computer Engineering  
Western University  
London, ON, Canada N6A 5B9  
{alheure2, kgroling, whigashi, mcapretz}@uwo.ca

**Abstract**—The Internet of Things (IoT) enables connected objects to capture, communicate, and collect information over the network through a multitude of sensors, setting the foundation for applications such as smart grids, smart cars, and smart cities. In this context, large scale analytics is needed to extract knowledge and value from the data produced by these sensors. The ability to perform analytics on these data, however, is highly limited by the difficulties of collecting labels. Indeed, the machine learning techniques used to perform analytics rely upon data labels to learn and to validate results. Historically, crowdsourcing platforms have been used to gather labels, yet they cannot be directly used in the IoT because of poor human readability of sensor data. To overcome these limitations, this paper proposes a framework for sensor data analytics which leverages the power of crowdsourcing through gamification to acquire sensor data labels. The framework uses gamification as a socially engaging vehicle and as a way to motivate users to participate in various labelling tasks. To demonstrate the framework proposed, a case study is also presented. Evaluation results show the framework can successfully translate gamification events into sensor data labels.

**Keywords**—Internet of Things; Sensor Data; Gamification; Data Analytics; Machine Learning; Crowdsourcing;

## I. INTRODUCTION

The Internet of Things (IoT) [1] is an ecosystem powered by sensors and microchips, which enables connection and communication among real-world objects, environments, software, and people. Through this network of things, sensors and devices are capturing and exchanging enormous amounts of data and fuelling the Big Data movement.

The functionality of the IoT depends upon four fundamental steps [2]: data acquisition, information extraction, knowledge extraction, and action-taking. Data analytics techniques, through the use of machine learning algorithms, can be used to extract information and knowledge from raw data. Supervised and unsupervised machine learning algorithms, however, rely strongly on data labels for proper functioning.

*Data labels* are defined as a representation of the ground truth or gold standard [3] of a data sample. Supervised machine learning algorithms are entirely dependent upon labels to learn and extract knowledge from data, and their performance is directly related to label quality [4]. On the other hand, unsupervised machine learning algorithms extract patterns or discover similarities from data without prior access to labels

[5]. In this case, however, labels are still important to validate algorithm accuracy.

The lack of labels describing the contextual information surrounding sensor data readings is one of the root challenges for data analytics within the IoT. This is especially true in case of human activity data: the data captured by sensors during performance of human tasks that affect sensor readings. For example, an electricity consumption sensor can capture variations in consumption when someone turns off a light or plugs in a device. Although gathered within the IoT, this type of sensor data is often processed and analyzed by field experts because such data cannot be easily reconciled and interpreted without prior domain and contextual knowledge. For instance, simply by looking at electricity consumption time-series data, it is hard to determine which device was turned on or off. In contrast, other types of data, such as images and social media posts, are human interpretable: an untrained user can correctly identify what the data represent, enabling better data labelling and consequently data analytics.

Crowdsourcing, a solution that leverages the power of crowds to perform tasks at a low cost [6], has been used for labelling. In the machine learning context, these tasks ask a large number of users to identify and label manually specific data such as images or tweets. The Mechanical Turk service [7] is an implementation of crowdsourcing. It enables researchers to post tasks to be performed, and in exchange for their participation, users receive financial compensation.

Such crowdsourcing frameworks are adequate for labelling tasks where humans are more effective than computers, such as identifying images. However, in the case of sensor data typical crowdsourcing frameworks are often ineffective due to poor human readability of the data. Users cannot simply look at sensor data and effectively extract information on what human activity the sensors are measuring.

This paper proposes a framework for sensor data analytics that leverages the power of crowds to enable sensor data labelling through gamification [8]. In the proposed framework, a game is designed to collect the labels needed for data analytics by asking users to perform specific tasks within the game. Each of these tasks is associated with labels, which are then automatically applied to the sensor readings. For example, to label electricity consumption data, a mobile game could ask users to perform tasks such as turning lights on

or off or plugging in different devices. Using this scenario, a case study was successfully conducted and demonstrated the ability of the framework to collect and label data while promoting behaviour changes. However, it could have also been used to motivate users to engage in physical activities by monitoring their movements using mobile sensors. The proposed framework has a flexible design so that it can be used for many types of sensors with which users can interact.

The rest of this paper is organized as follows. Related work and key concepts are introduced in Section II. Section III presents the proposed framework and the design of each of its components. Section IV discusses a case study, and Section V presents an evaluation of the framework. Finally, Section VI concludes the paper.

## II. RELATED WORK

This section presents related work regarding sensor data labelling and gamification.

### A. Sensor Data Labelling

To aid in the process of labelling, sensor data are often annotated with the activity surrounding the data reading. Activity recognition is a way of using sensors to identify activities and provide sensor context awareness to diverse applications [9]. To achieve high accuracy in activity recognition tasks, researchers like Machado *et al.* [10] and Roggen *et al.* [11] used expensive methodologies such as human observers and video labelling respectively. However, these solutions cannot be adapted for automated labelling because of their dependence on third party labellers. Using a different strategy, Cleland *et al.* [12] presented a prompted labelling approach in an attempt to create a more cost-efficient solution. The idea was to detect user activities and then prompt the users to define the activity through a mobile application. Although successful within a test environment, it was found that this technique would not be accurate or reliable in real life settings.

The three methods discussed enable accurate capture of activities surrounding sensor readings while consciously performing cost accuracy trade-offs. However, little consideration is given to the changing nature of the IoT: the methods developed do not account for the possibility of new activities or for rapid changes in labelling requirements. To address these issues, this work uses gamification to improve the interactivity and responsiveness of labelling techniques while remaining cost efficient.

### B. Gamification

The Gamification Summit defines gamification as “the use of game thinking and game mechanics to engage an audience and solve problems” [8]. Many different techniques exist to implement gamification, all of which aim to creating a deep and long-lasting engagement among participants, non-game activities, and supporting organizations [13]. Therefore, a key success criterion of gamification is whether the gaming features of an application enhance users’ internal motivation [13], enabling them to achieve their own goals [14].

Gamification requires four main components [15], which can be implemented in various ways:

- **Mechanics:** the building blocks of a game such as missions, leaderboards, and objectives.
- **Measurements:** the metrics used to quantify users’ progress and success.
- **Behaviour:** the behaviour that the game is attempting to promote. Games are related to intrinsic motivation and must therefore aim at modifying or promoting certain behaviour.
- **Reward:** the compensation given for participating.

Gamification has been integrated with basic questionnaires and quizzes in academic research, and the results have shown that gamification increases participation level [16]. It was also shown that the process does not negatively affect the quality of the captured data, nor is the performance of the test subjects negatively affected [16]. Not only does gamification increase the fun factor of an application, but it also enables more accurate work, motivates behaviour change [17], enhances retention rate, and is cost effective [18]. Therefore, due to its direct relation with intrinsic motivation, gamification is a suitable solution to increase and sustain users’ participation in crowdsourcing tasks.

## III. GAMIFICATION FRAMEWORK

The framework proposed in this research explores the power of crowds to capture and label sensor data through gamification. By leveraging these labels, the framework enables real-time and historical analysis of sensor data and facilitates the use of machine learning algorithms in the IoT context.

Figure 1 illustrates the main concepts of the framework. The central processing unit is built around two separate data streams: *gaming events* and *sensor events*. The gaming event stream is generated by users, who interact with the *Gamification* module by performing actions in a game application. The sensor event stream, on the other hand, is generated by the *Event Detection* module, which accesses the *Sensor Interface* to retrieve sensor data and identify events of interest. These two streams are sent to the *Event Labelling* module, which merges them and searches for correspondences between gaming and sensor events. By doing so, it associates data labels with the sensor readings and enables their use by *Analytics*.

Figure 2 depicts a more detailed view of the framework modules and their components. The following subsections discuss all modules in detail.

### A. Crowdsourcing Dashboard

The *Crowdsourcing Dashboard* is the central point of the framework and therefore interacts with every other module. It provides a means to capture the metadata needed by the framework, enabling the association between the sensor and gaming environments.

This module is composed of two main components: *Sensor Parameter Acquisition* and *Gaming Parameter Acquisition*. The first component captures sensor environment metadata, encompassing all information about the sensors such as their

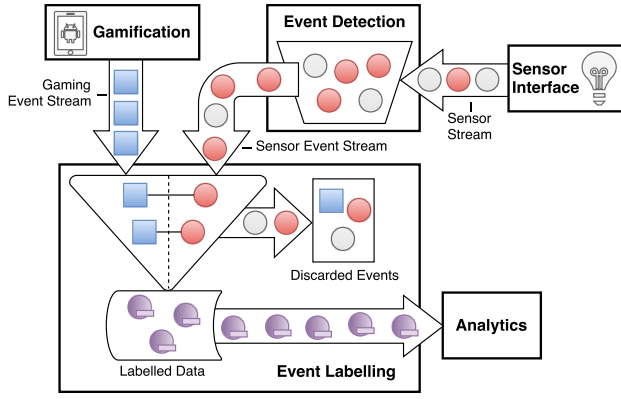


Fig. 1: Overview of the Gamification Framework.

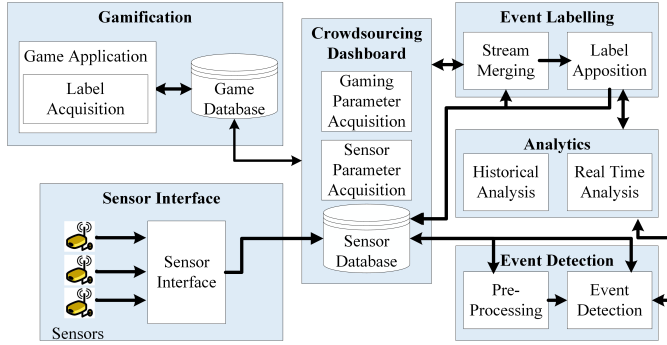


Fig. 2: Component-based view of the Gamification Framework.

category (e.g., HVAC or lighting), type (e.g., voltage or consumption), and physical location. Groups of sensors are also created in this component to bring together sensors that measure the same object.

The *Gaming Parameter Acquisition* component, on the other hand, captures gaming environment metadata. Three main entities are defined in this component: the *actions* users can take in the game (e.g., turn lights off or on), the possible *targets* for these actions (e.g., all light switches), and the rewards granted for completion of actions. Note that a target definition also includes all sensors associated with it, which enables the linkage between the execution of an action and the sensor readings that may be affected.

### B. Sensor Interface

The *Sensor Interface* module communicates with sensor APIs to obtain sensor readings. Because there is no standard for sensor APIs, the *Sensor Interface* module is flexible and can be easily extended to accommodate different formats.

Independently of their format, sensor readings are typically spatial time-series data containing attributes as well as object, time, and location information. The *Sensor Interface* translates these readings to persistent database objects that are stored in the *Sensor Database*.

### C. Gamification

The main goal of using gamification in this study is to motivate users to perform the targeted actions through the use of gaming mechanics. These actions (e.g., turning a light switch off) are physically accomplished and later on are associated with sensor readings, therefore providing the data labels needed for data analytics.

The *Game Application* component implements the gamification aspect of the framework. It is responsible for motivating and engaging users through the accomplishment of tasks and missions. Each of the four key gamification components [15] (mechanics, measurements, behaviour, and rewards) are embedded in the *Game Application* design in a way that enables customization.

In addition, the *Game Application* can also acquire gaming events that are used for data labelling. However, the means of capturing those events varies based on the game requirements. For example, in the case study presented in this paper, the gaming events are captured by scanning QR codes located near light switches. Therefore, each game has its own implementation of the *Label Acquisition* component, which enables the link between the accomplishment of an action and its specific target.

The *Gamification* module also includes a *Game Database*, which stores all events produced by the *Game Application* along with any additional information required for the game. This database is used later as the source for the gaming event stream.

### D. Event Detection

Given the real-time nature of the framework and the high sampling frequency of the sensors, the event detection algorithm must be efficient while maintaining accuracy. Because the vast majority of sensor readings do not record the performance of an action, the algorithm uses the difference between the current sensor reading and the previous one to emphasize the anomalous properties of the current reading. This *reading difference* calculation is defined by Equation 1 and is executed by the *Pre-Processing* component of the *Event Detection* module:

$$\Delta x_t = x_t - x_{t-1} \quad (1)$$

$$\forall \text{ sensors } x$$

where  $x_t$  and  $x_{t-1}$  represents the current and previous readings of sensor  $x$  respectively.

Another factor that must be considered by the algorithm is that an action may relate to multiple sensors; for example when a light is turned off, both the kilowatt and the voltage readings may vary. To associate all these readings with a single event, all sensors measuring the same object must be linked through a sensor group defined through the *Crowdsourcing Dashboard*, as described in subsection III-A.

Algorithm 1 describes the event detection process. For each sensor, the rolling mean of the reading differences is monitored; if a reading difference is greater than the calculated

mean, the reading is found to be anomalous, and an event is deemed to have occurred. Following this detection procedure, all readings originating in the same sensor group are also grabbed, and the detection is terminated for this group. Note that this algorithm is designed to catch all possible events, even if that includes detecting a few false positives. These will be dealt with during the *Event Labelling* step.

```

Data: Sensor Readings
Result: Event Data
foreach sensor of group m do
  read current data;
  check sensor rolling mean;
  if detects then
    get all current readings from m;
    save the current readings into an event;
    start event detection of the next sensor group;
  end
end

```

**Algorithm 1:** Event Detection Algorithm.

### E. Event Labelling

The *Event Labelling* module is responsible for carrying out the event labelling process shown in Figure 2. The module is composed of two main components: *Stream Merging*, which is responsible for aligning and combining the *gaming event* and *sensor event* streams; and the *Label Apposition* component, which matches gaming events with their corresponding sensor events.

As illustrated in Figure 2, the gaming event stream is created by the users playing the game. Given the nature of the framework, not every user input can be assumed to be true because users may by mistake indicate completion of an incorrect action. The sensor data stream, on the other hand, is generated by the *Event Detection* module and may also include a few false positives because its focus is on detecting all events, even if that includes a few incorrectly identified events. Because both the sensor and gaming events streams can contain spurious entries, the label apposition process takes the opposite approach to the event detection process. In this case, it is preferable to miss an event than to have a false positive because otherwise noise will be introduced into the labelled dataset and the validity of the framework will be highly compromised.

To ensure that only hard labels are obtained, a firm set of rules is required. *Label apposition* occurs in three steps:

- 1) Assign a label only if the gaming and sensor events fall within a strict sliding window, which is a time frame equal to the sampling rate of the sensors plus a pre-defined error margin.
- 2) Assign a label to a multi-event only if a corresponding label can be found for the combination of all events that occurred. For instance, when two light switches are turned off simultaneously, two gaming events are generated, yet only one corresponding sensor event

exists. In this case, the algorithm assigns a label to the sensor event only if there is a (single) label representing that both switches have been turned off.

- 3) Once the labels have been assigned, a clustering algorithm is executed to verify whether the readings are significantly different from other readings for the same target. This step is referred to as the *noise removal* process. In practice, clustering is executed using the *k*-means algorithm. All the readings for a specific target are clustered using one or more centres. The number of centres is determined by increasing it until the BSS/TSS ratio [19] is greater than 0.8, indicating good dispersion. The readings that do not appear to belong to the same cluster as the majority of the readings for the same target are then excluded, and those which belong receive the appropriate label. Note that other algorithms could also be used, but that the framework does not depend on this choice.

Algorithm 2 describes the event labelling process. It is composed of the firm set of rules presented above to ensure that the validity of the proposed approach is preserved. The algorithm rejects all events in which it does not have high confidence.

**Data:** Event Data, Gaming Data, sliding window

**Result:** Event Data Labelled

```

while Gaming Data is Available do
  get gaming data;
  get sensor event stream;
  if event was detected then
    if single event found in sliding window then
      Assign label to sensor event;
      Perform clustering to ensure label is accurate;
    else
      if combined event label exists then
        Assign combined label to sensor event;
        Perform clustering to ensure combined
          label is accurate;
      else
        discard the event
      end
    end
    if Assigned to expected cluster then
      Associate event with reading;
    else
      discard the event;
    end
  else
    get next sensor reading;
  end
end

```

**Algorithm 2:** Label Apposition Algorithm.

## F. Analytics

The framework was designed to recognize potential events in real time. Processes were put in place to extract these events, and therefore, as the game progresses, the number of labelled events increases. This enables the framework to learn actively from these labels, becoming more accurate over time.

The *Real Time Analysis* component can be used to process these events in a streaming fashion. This component is not tied to any specific algorithm or analysis implementation, but can be varied based on the targets entered in the *Crowdsourcing Dashboard*.

Furthermore, in addition to real time analysis, the *Historical Analysis* component can be used to perform batch processing within the framework. In this scenario, the newly found knowledge is used to extract additional information from historical sensor data.

## IV. CASE STUDY

The case study presented here was developed in collaboration with Powersmiths [20], a company located in Brampton, Ontario, Canada, that develops meters for measuring various electrical features at a very short interval. Although they had been storing the data captured by these meters for a number of years, they still needed a labelled dataset to obtain actionable insights. This was an excellent opportunity to deploy the proposed framework and enable achievement of some of Powersmiths' business objectives. Indeed, one of the company's goals was to promote sustainability and reduce energy waste, and therefore the mobile game application was developed to engage users in energy-saving challenges. Another company objective was to gain analytical insights from their data. The analytics module was consequently developed to provide real-time and historical analysis relevant to sustainability. The means by which each of these goals was achieved and the implementation details are discussed below.

### A. Game Application

The objective of the game was to encourage sustainable actions among communities of customers by creating a competitive environment where the users who gathered the most points by performing sustainable actions were rewarded. Various gaming mechanisms were incorporated. First, the application used a level hierarchy, each level unlocking more actions, to motivate the players to progress. Second, a leaderboard was implemented so that users could identify where they stood with respect to their community. In addition, the reward artifact was integrated by enabling administrators to assign a point value to each task. By giving this type of control to the administrator, specific tasks could be favoured to ensure that enough data were gathered from the targets.

The game was tested within Powersmiths' own facility and, due to the environmental impact of lighting-related issues, the actions were focussed on turning lights on and off. Using the *Crowdsourcing Dashboard*, administrators created QR codes for each of these actions and affixed the codes close to the target objects. To ensure that actions were recorded in good

faith, a participant who accomplished an action had to scan the QR code physically with a smartphone. The deployment and use of the game not only promoted sustainability, but also enabled the acquisition of a labelled dataset for the analytics module.

### B. Analytics

The real-time analytics component was designed to monitor the lighting and occupancy of the facility; as events were taking place, the status of each area was observed and updated in real time. The idea was to use the acquired data labels to create clusters for each target sensor group. As new sensor events were detected, they were compared against the centroids of each cluster and assigned to the class it was closest to. By joining these newly classified data with room occupancy sensor data, it was possible to establish the level of energy waste in the facility.

The historical analytics component also made use of occupancy sensors and of the framework's ability to classify detected events to extract insights from electrical usage data. Daily historical reports were created to provide a snapshot of facility usage. Within each daily report, every single event detected was listed, and classification was performed to determine which event occurred. Event detection and classification enabled the lighting status of each area to be extracted. Because the occupancy status of any area of the building could be obtained through the motion sensors data, it was possible to determine whether the facility was wasting energy at any given time.

The combination of real-time and historical analysis components provided Powersmiths with the analytical insights that they had been looking for; they were able to extract additional value from the data they had already been gathering.

### C. Implementation

Each module from Figure 2 was implemented as follows:

- **Crowdsourcing Dashboard:** The *Crowdsourcing Dashboard* was implemented as a REST API using the Play Framework [21]. The *Sensor Database* was implemented using MySQL.
- **Sensor Interface:** The *Sensor Interface* was implemented as a Web service using Scala and the Play Framework. The service queried the sensors' REST API every 30 seconds and stored the results as unlabelled in the *Sensor Database*.
- **Gamification:** The game application was implemented on the Android platform. Actions were captured through the use of QR codes and a QR code scanner within the game. The *Game Database* was implemented using Parse [22].
- **Event Detection:** Both data pre-processing and event detection were implemented as database functions and triggers in the sensor database.
- **Event Labelling:** The event labelling component was implemented using database functions and R [23], a statistical analysis program.

- **Analytics Module:** The analytical components were implemented using the R platform, and the real-time and historical analytic applications were developed using R Shiny [24].

## V. EVALUATION

### A. Setup

Over approximately four months, the framework was deployed to monitor the lighting events on the second floor of the Powersmiths facility. The lights on the second floor were all monitored using a single device with multiple sensors, which provided the four readings used by the case study: the kW for each of the three current phases and the total kW consumption.

Throughout the case study, a total of 1,823,289 readings, were analyzed by the framework. To evaluate the framework, a manual gaming experiment was conducted with the help of two users playing the game using the mobile application. The experiment enabled the capture of 124 gaming action records. During the experimentation, the sensor interface was not serving data consistently or regularly, and the 30-second sampling interval was not maintained. In fact, of the expected 2880 daily readings from each sensor, only 1808 complete readings per sensor were recorded, equivalent to a sampling rate of 62.8%. As a result, of the 124 gaming actions, the framework captured only 90 corresponding readings, for a sampling percentage of 72.5%. The reason that some of the gaming actions could not be assigned was that no corresponding reading was available from the sensor API at the time the actions were completed. In addition, it was also discovered that 4 actions occurred within the same sampling window. Since this evaluation was only interested in single events, these 4 actions were discarded, resulting on an initial set of 86 gaming actions.

Given that the authors are not aware of any other framework that performs the same tasks, the evaluation was performed on a component basis. The various evaluations, as well as their objectives, methodology, and results, are discussed in the following subsections.

### B. Labelling Evaluation

This section evaluates the framework's ability to detect false events and wrong labels and its ability to translate gaming events into labels.

1) **False Gaming Event Detection:** The framework must be able to detect events, but as importantly, given the human component of the framework, it must prevent the introduction of false gaming events. False gaming events are defined as actions that are registered (scanned), but were not physically performed. If these were not eliminated, the created dataset would not be reliable and could not be effectively used for data analytics.

To evaluate false gaming event detection, an experiment was conducted during the case study: targets were scanned, but no corresponding physical actions were performed. The test's objective was to ensure that players that intentionally or accidentally scanned a target without performing an action would not introduce invalid data into the dataset. In a

monitored environment, nine targets were scanned without physically performing the associated action. The nine actions were recognized as gaming events, but because there was no matching sensor event, they were all discarded.

This test therefore shows that the framework was able to fully prevent false gaming events, which is critical to the success of the framework.

2) **Wrong Label Detection:** The framework links physical actions to sensor readings to create a labelled dataset for analysis. For the framework to fulfil its purpose, the labels apposed must be accurate to limit the amount of noise or wrong labels introduced into the dataset.

Therefore, the framework's ability to detect wrong labels was evaluated. Attempts were made to inject invalid samples into the dataset by having a user perform a labelled action while capturing the wrong target. In other words, a light switch was turned off, but another light target was scanned. The goal of this procedure was to evaluate whether the framework would accurately identify the results as invalid labels. The framework was successful in recognizing all four attempts to inject wrong labels as invalid actions. Table I shows these results. It can be observed that the clustering algorithm executed by the *Label Apposition* component placed the noisy readings into a different cluster than the cluster containing the properly labelled readings, leading to the rejection of the injected readings.

Therefore, this test showed that the framework is not susceptible to noise introduction because it can counteract accidental or purposeful introduction of noisy data.

3) **Assigned Labels:** The main objective of the framework is to use gamification to create a labelled dataset, in order to enable analytics. Therefore, it is important to evaluate the framework's ability to translate gaming events into labelled sensor readings. This evaluation was performed by comparing the size of the final labelled dataset to that of an *ideal* dataset, where all relevant gaming events were turned into labels.

However, before the data labelling conversion rate could be evaluated, the dataset had to undergo a filtration step to remove the invalid data that had been purposefully introduced by the false event and wrong label detection tests. During the experiment, 86 gaming events were recorded, but after removing the purposefully injected actions used to evaluate the framework a set of 73 valid unique detected actions remained. This means that if every valid action was correctly labelled, the resulting dataset should contain 73 labelled readings. However, this was not the case. Indeed, the framework has a built-in mechanism to remove potentially invalid or noisy data, which is known as the *clustering phase* within the *Event Labelling* module. This step was therefore carried out, clustering was performed for each of the target sensors, and a number of potentially noisy data readings were removed.

From the 73 valid detected actions, 53 were left. Therefore the framework translated 72.6% of the gaming actions into labels. However, this percentage could well have been higher if the experiment were performed under ideal conditions with a consistent sensor reading interval.

TABLE I: Wrong Label Detection.

<i>ActionTime</i>	<i>ReadingTime</i>	<i>TargetNumber</i>	<i>AssignedCluster</i>	<i>TrueCluster</i>
2014-12-16 12:55:29.692000	2014-12-16 12:55:32	12	3	1
2014-12-16 13:02:18.226000	2014-12-16 13:02:32	10	2	3
2014-12-16 13:05:26.451000	2014-12-16 13:05:32	10	1	3
2014-12-16 13:33:55.048000	2014-12-16 13:34:02	4	1	2

TABLE II: Framework Comparison with Unlabelled Techniques.

<i>No</i>	<i>Case</i>	<i>Algorithm</i>	<i>Accuracy</i>
1	Unprocessed Event Readings (No Noise Removal)	<i>K</i> -Means	31.08%
2	Pre-Processed Event Readings (No Noise Removal)	<i>K</i> -Means	64.86%
3	Unprocessed Event Readings With Noise Removal	<i>K</i> -Means	35.84%
4	Pre-Processed Event Readings With Noise Removal	<i>K</i> -Means	71.69%
5	Pre-Processed Event Readings and Labeled Centres	KNN	88.67%

### C. Evaluation of the Overall Framework

The value of the framework depends on its ability to perform better than other known solutions, more specifically in terms of accuracy. Because the authors are not aware of frameworks with the same purpose as the novel solution proposed here, the evaluation was conducted by performing various analytical comparisons against well-established algorithms and the various components of the framework.

The accuracy metric used as proposed by Metz [25] is defined by Equation (2):

$$Accuracy = (tp + tn)/total \quad (2)$$

where  $tp$  represents the number of true positives (the number of sensor events correctly labelled),  $tn$  represents the number of true negatives (the number of unlabelled or incorrectly labelled events) and  $total$  represents the total number of samples in the dataset.

The five experiments listed in Table II were carried out to evaluate different components of the proposed framework:

1) **Unprocessed Event Readings:** This experiment served as a baseline for comparing the contributions of the framework components. It evaluated the accuracy of unsupervised technique (clustering) using raw data as input. Here, the raw data set consisted of sensor readings corresponding to the events identified by the framework. This dataset was classified using the  $k$ -means algorithm implemented in the Weka [26] knowledge analysis tool. In an unsupervised unlabelled context, the only information that would be accessible to the user would be the number of potential clusters or targets. Therefore, this information was provided to the algorithm, meaning that the algorithm was aware of the number of possible targets available.

2) **Pre-Processed Event Readings:** The objective of this experiment was to assess the benefits of pre-processing. It was the same as experiment #1, except that the data were pre-processed according to Equation (1) instead of being raw data.

3) **Unprocessed Event Readings w/ Noise Removal:** The goal of this experiment was to explore the effect of the noise removal process described in Section III-E. This experiment

attempted to demonstrate the effect of the noise removal process. Like experiment #1, it used the raw data set and  $k$ -means algorithm implemented in Weka. However, before clustering, noise was removed using the noise removal process from the proposed framework.

4) **Pre-processed Event Readings w/ Noise Removal:** Similar to experiment #3, this experiment also showed the effect of noise removal. Whereas experiment #3 used raw data, this experiment used data pre-processed according to Equation (1).

5) **Supervised Classification:** The goal here was to demonstrate the need for labelling. It is possible to stipulate that unsupervised classification may be sufficient for the scope of this work and that labelling may not provide significant advantages. Therefore, this experiment demonstrated the importance of having access to labelled data by comparing the results obtained by unsupervised algorithms ( $k$ -means) to those obtained by the  $k$ -nearest neighbour (KNN) algorithm trained with the labelled dataset created by the framework. The KNN algorithm was implemented in R. The training set provided to the KNN algorithm was chosen to be equal to the centroid of each of the cluster labels (each cluster contains all the event data labelled for a specific target), and the number of neighbours was set to one.

From those five experiments and their accuracy as shown in Table II, the following conclusions can be drawn:

- Pre-processing significantly improves the classification accuracy. When pre-preprocessing was added to pure clustering in experiments #1 and #2, the accuracy increased from 31.08% to 64.86%. Moreover, when pre-preprocessing was added to the scenarios with noise removal, experiments #3 and #4, accuracy increased from 35.84% to 71.69%
- Noise removal process improves overall classification accuracy. Experiment #3 added noise removal to experiment #1, resulting in an accuracy increase from 31.08% to 35.84%. Similarly, experiment #4 added noise removal to experiment #2, resulting in accuracy change from 64.86%

to 71.69%.

- Labels significantly increase the classification accuracy. The supervised method using the created labels showed an accuracy of 88.67%, whereas the best unsupervised approach, experiment #4, exhibited an accuracy of 71.69%.

#### D. Analytical Evaluation

The analytics module was not formally evaluated, but informal interviews were conducted to review the observed accuracy of the real-time and historical analysis components. The interviewee commented on how the reports enabled him to change some wasteful consumption habits effectively. For example, one area was identified as wasteful and the daily reports truly helped Powersmiths rectify the situation. These reports enabled the administrator to track the effectiveness of gamification in producing behaviour changes. It was remarked that the game had a positive effect and that the waste observed within a specific area was almost completely eliminated since the framework was implemented. A positive correlation between the findings in the reports and the observed waste was successfully established, thereby validating the insights provided by the framework.

#### VI. CONCLUSION

The IoT aims to create a responsive environment where devices, humans, and software can seamlessly interact. However, knowledge and information must first be extracted from sensor data using analytics techniques such as machine learning. Machine learning relies on data labels: supervised learning needs them for training and unsupervised learning for validation. Although some data, such as images, can be relatively easily labelled by humans, users cannot simply look at sensor data and identify what activity the sensors are measuring.

This paper proposes a gamification framework for sensor data analytics. By translating sensor data into human readable gamification tasks, the framework enables data labels to be applied automatically, thereby improving sensor data analytics in the context of the IoT. In addition, use of gamification for crowdsourcing enables humans to become immersed within the entire process and encourages behaviour changes by engaging users to accomplish specific tasks. A case study demonstrated the use of the gamification framework for labelling electrical sensor data. The results showed that the framework can assign labels accurately to sensor data. Moreover, game users positively changed their behaviour, and these changes were reflected in the framework's historical reports.

Future work will investigate adapting the gamification framework for Big Data to accommodate fully the needs of an IoT environment with a large number of sensors. The efficiency and applicability of the framework in more complex scenarios will also be evaluated.

#### ACKNOWLEDGMENTS

This research was supported in part by an NSERC CRD at Western University (CRDPJ 453294-13). In addition, the authors would like to acknowledge the support provided by Powersmiths.

#### REFERENCES

- [1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [2] M. Swan, "Sensor mania! the Internet of Things, wearable computing, objective metrics, and the quantified self 2.0," *Journal of Sensor and Actuator Networks*, vol. 1, no. 3, pp. 217–253, 2012.
- [3] A. Tarasov, S. J. Delany, and C. Cullen, "Using crowdsourcing for labelling emotional speech assets," in *Proceedings of W3C workshop on Emotion ML*. Dublin Institute of Technology, 2010.
- [4] T. Kulesza, S. Amershi, R. Caruana, D. Fisher, and D. Charles, "Structured labeling for facilitating concept evolution in machine learning," in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '14, 2014, pp. 3075–3084.
- [5] D. Ennis, "Using pre-and post-process labeling techniques for cluster analysis," Ph.D. dissertation, University of Nevada, Reno, 2014.
- [6] A. Kittur, "Crowdsourcing, collaboration and creativity," *XRDS*, vol. 17, no. 2, pp. 22–26, 2010.
- [7] Amazon. (2017) Mechanical turk. [Online]. Available: <https://www.mturk.com/mturk/welcome>
- [8] "What is gamification?" Gamification Summit, Tech. Rep., 2013. [Online]. Available: <http://sf13.gsummit.com/sf13/what-is-gamification/>
- [9] M. Stikic, D. Larlus, S. Ebert, and B. Schiele, "Weakly supervised recognition of daily life activities with wearable sensors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 12, pp. 2521–2537, 2011.
- [10] I. P. Machado, A. L. Gomes, H. Gamboa, V. Paixao, and R. M. Costa, "Human activity data discovery from triaxial accelerometer sensor: Non-supervised learning sensitivity to feature extraction parametrization," *Information Processing & Management*, vol. 51, no. 2, pp. 204–214, 2015.
- [11] D. Roggen, K. Förster, A. Calatroni, A. Bulling, and G. Tröster, "On the issue of variability in labels and sensor configurations in activity recognition systems," in *Workshop at the 8th International Conference on Pervasive Computing (Pervasive 2010)*, 2010.
- [12] I. Cleland, M. Han, C. Nugent, H. Lee, S. McClean, S. Zhang, and S. Lee, "Evaluation of prompted annotation of activity data recorded from a smart phone," *Sensors*, vol. 14, no. 9, p. 15861, 2014.
- [13] S. Nicholson, "A user-centered theoretical framework for meaningful gamification," *Games+ Learning+ Society*, vol. 8, no. 1, 2012.
- [14] B. Burke, *Gamify: How gamification motivates people to do extraordinary things*. Routledge, 2016.
- [15] C. E. Bess, "Gamification: Driving behavior change in the connected world," *Cutter IT Journal*, vol. 26, no. 2, pp. 31–37, 2013.
- [16] J. Cechanowicz, C. Gutwin, B. Brownell, and L. Goodfellow, "Effects of gamification on participation and data quality in a real-world market research domain," in *Proceedings of the First International Conference on Gameful Design, Research, and Applications*, ser. Gamification '13, 2013, pp. 58–65.
- [17] K. Robson, K. Plangger, J. H. Kietzmann, I. McCarthy, and L. Pitt, "Game on: Engaging customers and employees through gamification," *Business Horizons*, vol. 59, no. 1, pp. 29 – 36, 2016.
- [18] G. Kazai, Lumi, F. Hopfgartner, U. Kruschwitz, and M. Meder, "Ecir 2015 workshop on gamification for information retrieval (gamifir'15)," *SIGIR Forum*, vol. 49, no. 1, pp. 41–49, 2015.
- [19] M. Greenacre and R. Primicerio, *Multivariate Analysis of Ecological Data*, ser. Manuales Fundación BBVA. Fundación BBVA, 2014.
- [20] Powersmiths. (2017) Powersmiths: Power for the future. [Online]. Available: <http://ww2.powersmiths.com>
- [21] Play. (2017) Play framework. [Online]. Available: <https://www.playframework.com>
- [22] Parse. (2017) Parse. [Online]. Available: <https://www.parse.com>
- [23] R Core Team, *R: A Language and Environment for Statistical Computing*, Vienna, Austria, 2013. [Online]. Available: <http://www.R-project.org/>
- [24] RStudio, Inc, *shiny: Easy web applications in R*, 2014. [Online]. Available: <http://shiny.rstudio.com>
- [25] C. E. Metz, "Basic principles of roc analysis," vol. 8, no. 4, pp. 283–298, 1978.
- [26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.