Data Cleansing for Remote Battery System Monitoring

Gregory W. Ratcliff
Director, Life-Cycle Management
Emerson Network Power

Randall Wald
Senior Research Associate
Florida Atlantic University

Taghi M. Khoshgoftaar
Director, Data Mining and Machine Learning Laboratory
Florida Atlantic University

Abstract

Uninterruptable Power Supply (UPS) systems are essential in modern-day industry to ensure that equipment continues to function even in the event of power failure. These UPS systems must themselves be resistant to failure to guarantee that they will be working when necessary. Because monitoring and maintaining these UPS systems is beyond the scope of many industries, it is contracted out to firms that manage systems at hundreds of installations nationwide. These companies require automated monitoring tools to use sensor data (collected by instruments installed on the UPS systems) to determine when batteries are in need of replacement. In this paper, we discuss a tool for detecting unusual behavior in battery resistance readings (indicative of impending or recent battery replacement), and we present a case study demonstrating the effectiveness of this tool on real-world battery data.

I. Introduction

Many large-scale industrial systems rely on continuous power availability to function: at any given point in time, different components or materials may be in a fragile state which would be damaged by a sudden loss of power. These systems must have time to return to a safe state prior to shutoff, but unexpected loss of power does not grant them this time. To avoid such problems, those who operate these machines must rely on battery backup systems, which will provide enough power to sustain the machinery even if the primary power source is lost. These backup systems must work reliably as soon as they are needed, even though they may sit for months or years between uses.

Typically, a battery backup system will consist of one or more “strings,” each of which is composed of many “jars” (which are the actual voltaic components). With the data used in our case study, for example, each system consists of one or two strings, which are each composed of forty jars. The jars within a string are connected in series, which means the voltage drop across a string is the sum voltage drop across the string’s jars. Thus a failure of a single jar will degrade an entire string, and unless there are multiple redundant strings powering the same equipment, this would remove the backup for the protected system.

Because these battery backup systems are essential to maintain system reliability (there cannot be another backup for the backup), and have high voltages, they must be installed and monitored by professionals working for a firm that specifically handles deployment and maintenance of battery backup systems. These firms may be responsible for thousands of deployments throughout the country, which means it is not feasible for them to have personnel manually inspect all systems regularly to ensure that they are functioning properly. Instead, they employ remote sensors, which detect various properties of the battery systems (such as voltage, resistance, and temperature) and report these to a central office for monitoring. Because the data from these sensors will be used for determining the health of the battery systems, it is important to verify the quality of this data.

Many types of errors can be found in the sensor data from battery backup systems, not all of which indicates an actual problem with the system being monitored. For example, in many cases a sensor will only poll the physical hardware for a new value once every two weeks, or even once a month, but will retain the most-recently-read value on internal memory. If this sensor is queried for its status between readings, it will report that most recent value, even though the value may come from days or weeks in the past. Sensors can also be damaged or disconnected without actual harm to the underlying battery system, which may result in impossibly low or high reading values or values which are missing entirely. All of these types of errors will impact the quality of the data used to monitor the battery systems and will make it more difficult for humans or monitoring algorithms to process this data and detect real changes in the monitored systems.

In this paper, we present a case study of real-world battery monitoring data which exhibits the discussed errors, as well as the results of using a tool to remove these errors. Drawing on extensive data that has been collected over many years Ratcliff [1], we will show how the error types can make it difficult to interpret the raw data, and how once these have been automatically removed with the tool, the cleansed data will retain all of the important information about the monitored battery system, while being free of the noise and errors which could hinder interpretation. Finally, we will discuss how this tool may be integrated into a battery monitoring workflow to ensure that all data (even error data) is retained, so that errors resulting from direct damage to the sensors may be addressed appropriately.

The organization of this paper is as follows: Section II presents related work on the topic of data cleansing for remote monitoring systems. Section III discusses the error types in more detail, along with approaches used by our tool to deal with and cleanse these errors. Section IV contains our case study discussing both the raw data and how the tool was able to cleanse it. Finally, Section V holds our conclusions and discussion of how the demonstrated tool fits into the larger problem of building a remote battery monitoring system.

II. Related Work

Relatively little work has considered the importance of data cleansing for improving the quality of sensor data, especially in the context of monitoring battery systems. Jeffery et al. has developed a broad framework for addressing missing and unreliable sensor data from RFID tags, called the Extensible receptor Stream Processing (ESP) framework [2], [3]. This framework recognizes that in general, data points which are close in space and/or time will generally share similar values, and therefore readings which do not share values in this fashion are likely to be erroneous. A five-step preprocessing approach is employed to address these potentially erroneous data points by removing outliers, interpolating missing values both temporally and spatially (in separate steps), handling duplicated readings, and combining readings from different sensor types. By employing this framework, only the useful data will be retained, while erroneous and missing values will be fixed, thus producing a more useful dataset.

In addition to ensuring data quality by removing or replacing faulty data, it can be useful to retain information about the quality of each individual data point and ensure this information is propagated to downstream systems. Klein [4] developed just such a system, which uses a metamodel to assess data quality in terms of both accuracy and completeness, which then either processes the data to improve its quality or simply marks the data as potentially low-quality so that it may be dealt with accordingly in the future. Klein's system also takes into account the data storage and processing considerations which must be dealt with when handling large quantities of potentially-faulty data. Overall, this framework permits data to be interpreted and handled even in the event of data quality concerns so that all processing takes potential errors into account.

III. Battery Instruments and Error Types

Instruments installed on battery systems come in many types and can suffer from many faults. As discussed earlier, a battery backup unit is composed of strings which themselves contain jars. Sensing can take place either at the string or jar level: for example, typically voltage and resistance are measured per jar, while the temperature is measured across an entire string.

Temporal Resolution and Repeated Values

Different sensors have varied temporal resolution. Depending on the installation parameters, the voltage and resistance levels are either generated once every two weeks or once per month. These varying time scales pose a challenge for interpreting battery sensor data, a challenge made even harder by the problem of duplication. Whether or not the sensor is scheduled to perform a new measurement and run a new resistance test, the sensor will return a remote value daily, which is usually the last recorded reading. Since there is no overt signal whether the presented value is new or repeated, from a remote monitoring and data perspective it can be difficult to judge the importance of each value.

One naïve approach to alleviate this problem is to retain only the values provided once every two or four weeks and discard the rest. However, this misses an important subtlety of remote battery measurements: the testing interval is measured from when the jar was first installed or last replaced, and a jar can be replaced in the middle of a instrument stream of inbound data. In other words, the key dates which represent new sensor readings can change over the course of a jar's lifetime, meaning that using a fixed interval will give poor results.

To solve this problem, the tool implemented and used in this case study first operates by eliminating repeated values: that is, for each day, if the sensor value of that day is identical to the previous day's value, it will be removed. This ensures that genuinely new values are retained regardless of how long it was since a new value was previously recorded. Two problems remain with this technique, however. First of all, if a new value actually is identical to the old value, it will be eliminated as a repeated value even though it is new information. Secondly, if adjacent values are nearly but not completely identical (for example, in our case study we found tests where resistance values fluctuated between 4255 and 4256 mΩ), all of these values will be retained, even though they may represent repeated values. Although the present study does not address either of these potential problems, they will be considered in future work.

High and Low Values

In addition to producing duplicated values, certain types of sensor errors can result in abnormally high or low values. For example, if an engineer accidentally jostles a sensor while performing upgrades, its leads may be disconnected partially or short onto another part of the machinery. Generally speaking, experienced engineers examining the data will know which range of values are reasonable and which are indicative of these errors, but sometimes these "known good" ranges are unknown, especially for newer equipment which has yet to develop a broad baseline of normal behavior. Thus, two separate approaches are needed to address this problem of high and low sensor values.

The first of these approaches is conceptually simple: engineers who are familiar with the battery system define a low threshold and a high threshold, so that values which fall outside these are marked as low or high, respectively. When these thresholds are insufficient or unknown, however, a second approach may be employed. Here, statistical methods are used, considering all of the points collectively (disregarding their temporal sequence) and finding which constitute outliers in a mathematical sense. The threshold for outliers is found based on the data without any prior knowledge. Using this automatic method, outliers with high or values can be removed even if preexisting thresholds are unknown.

Missing Values

Finally, some sensor faults can result in values missing altogether: a sensor can be entirely disconnected, can break, or can lose its signal to the remote data center which collects the data from different installations. All of these will result in a time instance which is missing an appropriate value entirely. The options for dealing with these values are to interpolate them based on known values or remove them altogether; this latter approach (removal) is used in the present work.

IV. Case Study

In this case study, we developed a tool to apply the data cleansing techniques discussed in Section III, and apply them to a case study of real-world data. The dataset itself is presented in Section IV-A, while the results of the tool are shown in Section IV-C.

Dataset

The data used in this case study is collected from a dataset which has been acquired during many years of large-scale deployment of battery systems. It consists of values reported from each jar over the course of approximately four years (specific durations vary by jar), with each instance containing the string tag and string number (unique IDs used to represent which installation and which string at a particular installation is being monitored), jar number (specific jar within a given string), resistance value, and reading date. Resistance value was used as the sole sensor due to the scope of this case study, and the reading date is used as the x-value to represent how the instances are separated in time (all other features are merely used to identify individual points). In total, there are 16 strings (unique combinations of string tag and string number) and 40 jars in each string.

Tool Development

The first processing step used by the tool is to collect this data and reorganize it, such that each string and jar can be manipulated separately and examined in sequence. Although this step is simple conceptually, the tool is able to accomplish this step efficiently even when operating on large (> 1 GB) data files that are generated due to the life of a typical battery jar in this type of service. As discussed in above, there are three primary preprocessing filters applied with the tool: removing duplicated values, removing values which surpass user-defined thresholds, and removing values which are mathematical outliers. Different datasets may not need all of these filters applied: for example, in a given circumstance, user-defined thresholds may not be known, or duplicates may be desired. To accommodate these different use cases, all preprocessing steps can be enabled or disabled separately, based on what forms of cleansing the user wishes to apply. In the experiments discussed in the following section, all three cleansing steps were applied.

Results

When looking at different string and jar combinations, we find several some exhibit some errors while others exhibit others. Some combinations exhibit a combination of errors. In Figures 1 through 6, we see a number of different variations, with each figure containing the data both before and after cleansing. The primary goal is to separate instrument data that is anomalous from real battery data which represents a physical or chemical change in a jar over its entire lifetime.

Figure 1 presents a fairly typical case, where the resistance data breaks into three clear time sequences: one where it starts a bit over 5000 mΩ and slowly rises, one where it starts at nearly 5000 mΩ exactly and stays relatively stationary, and finally one where it starts well below 5000 mΩ and also does not show significant movement. Post-cleansing, we see that the duplicated values found so often in the second and third regime are mostly removed, retaining only the slight variances which were found in the first regime as well. This demonstrates that the cleansing can help remove the differences between datasets which share many similar underlying properties.

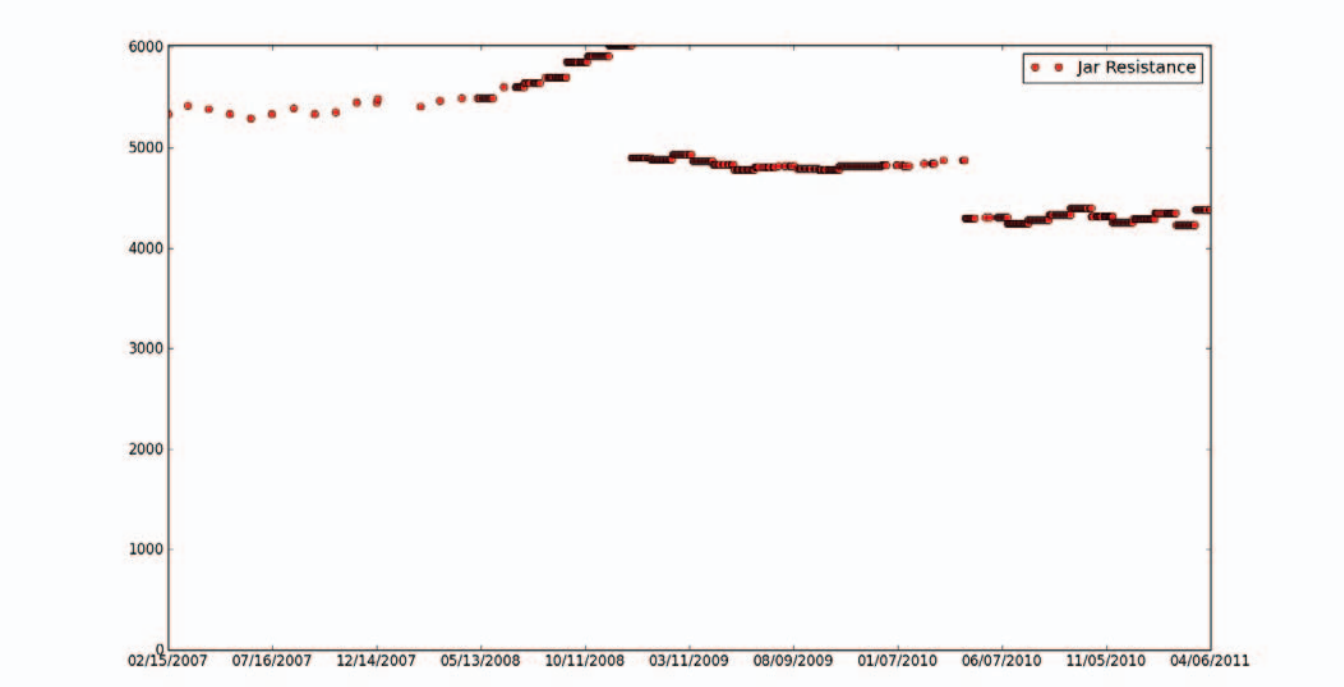

Figure 1 - Results for String Tag 1136286, String Number 1, Jar 6

The combination presented in Figure 2 shows much the same, although with only one distinct regime of data that rises and falls.

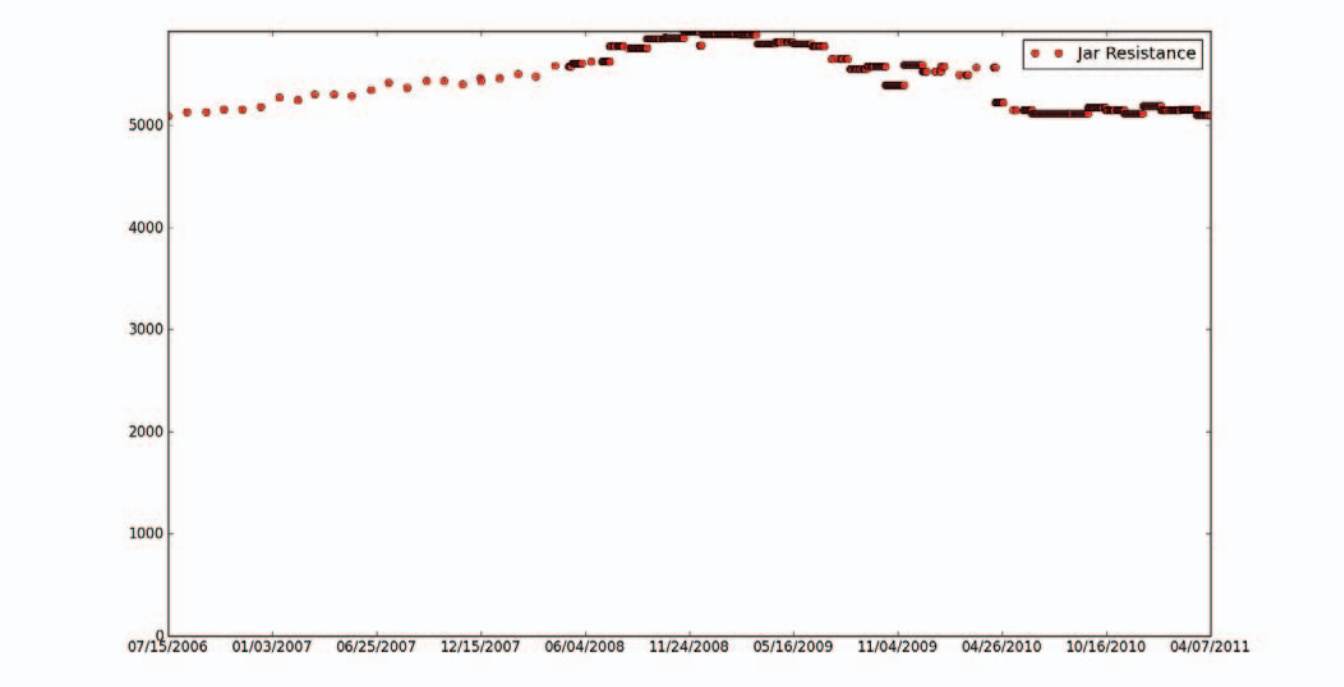

Figure 2 - Results for String Tag 1154622, String Number 1, Jar 39

Figure 3 shows a relatively constant data source with similar patterns.

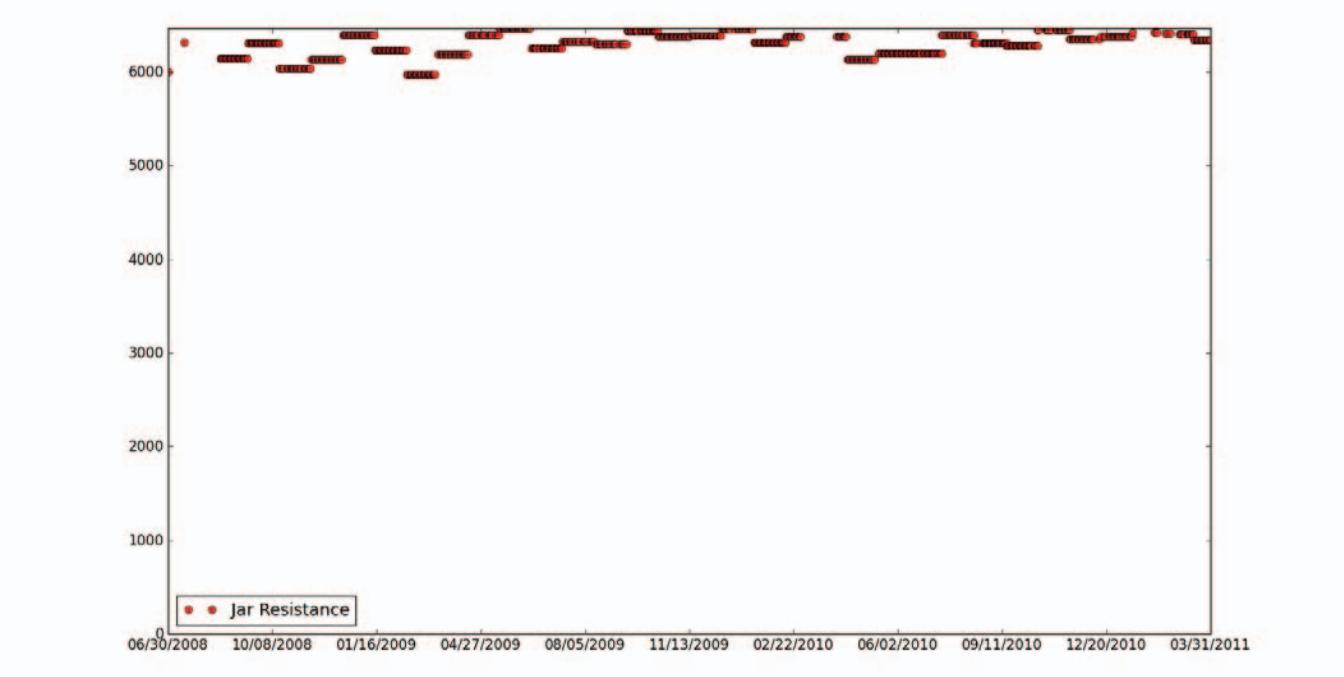

Figure 3 - Results for String Tag 1290668, String Number 1, Jar 6

Figures 4 through 6 demonstrate another problem: outliers. Figure 4 shows an example with extreme high outliers (and one low outlier at the end), while Figure 5 shows one with extreme low outliers (both at the beginning and in the middle of its time sequence) and Figure 6 has notable problems in both directions. Note that while repeated values are primarily a problem for automated analysis of sensor data (because the frequent repeated values could erroneously lead an algorithm to believe that a signal is more stable than it really is), outliers (and in particular high values) can obscure human interpretation of sensor data, because graphs such as these which scale to the largest observed value are often useless when the highest observed value dwarfs the important values. Outliers are very important data elements to be analyzed as they may represent impending catastrophic failure of a jar or string.

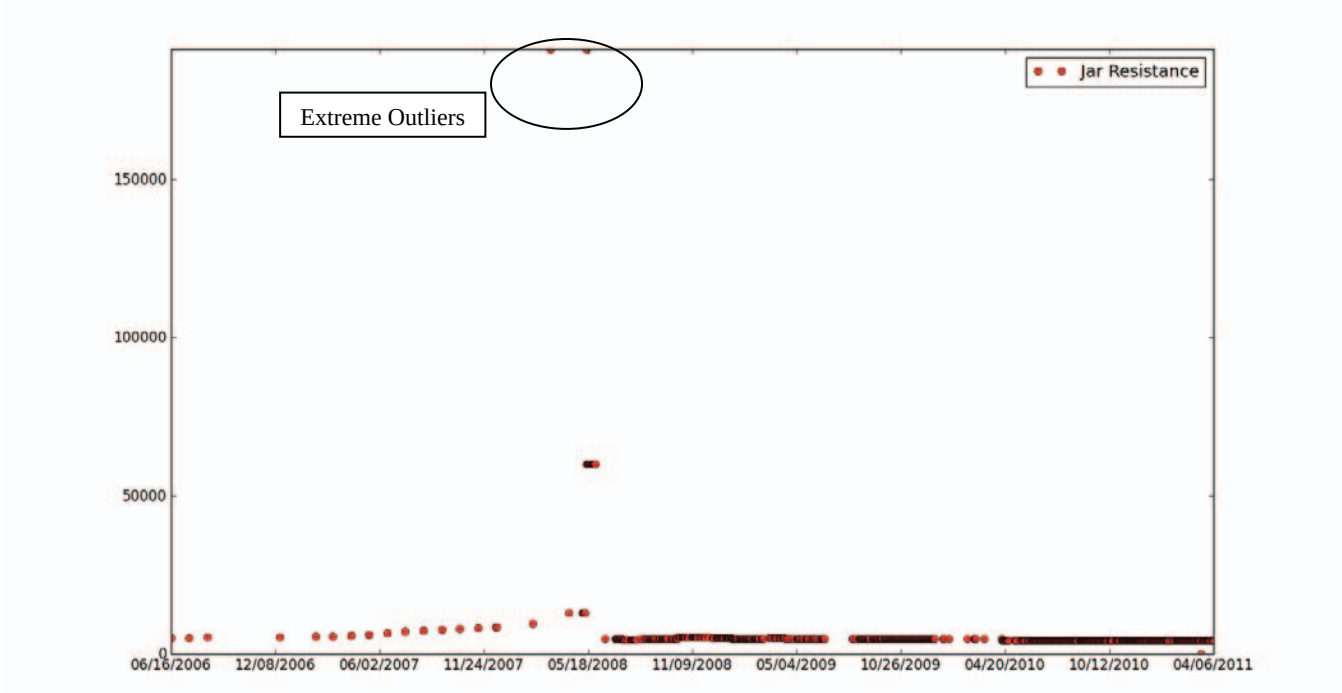

Figure 4 - Results for String Tag 1293310, String Number 4, Jar 2

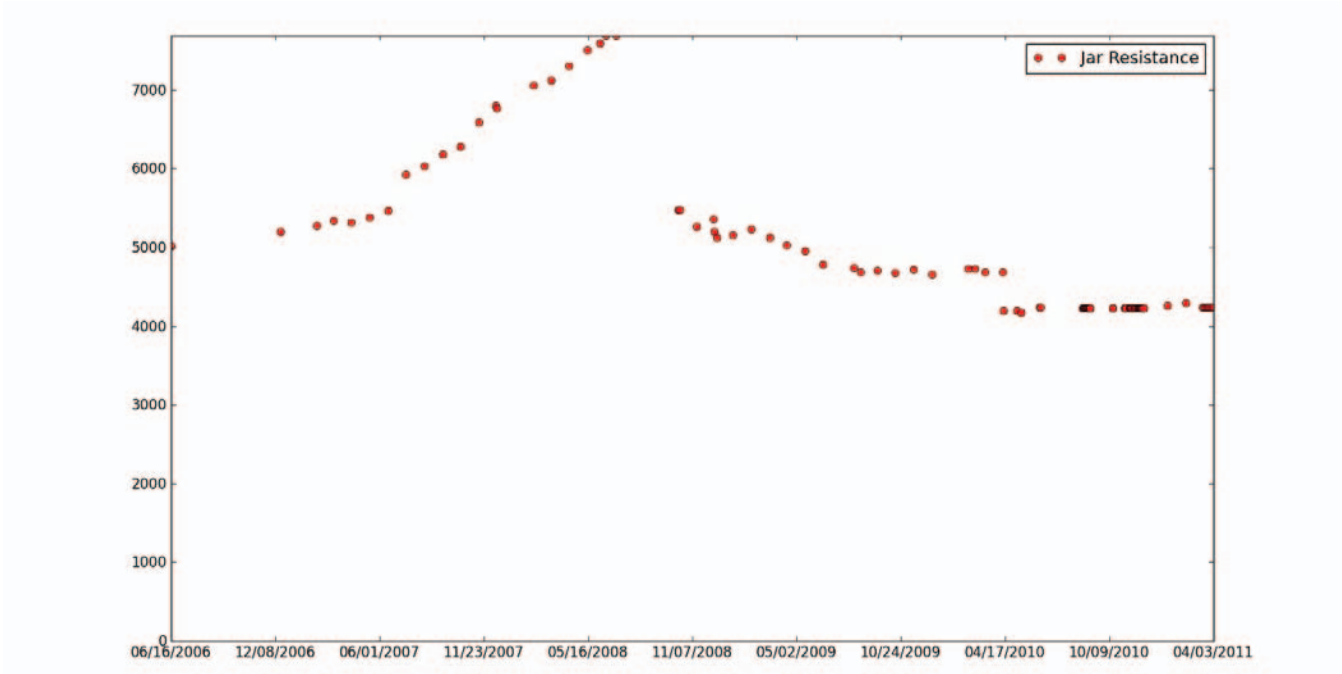

Figure 5 - Results for String Tag 1293307, String Number 1, Jar 6

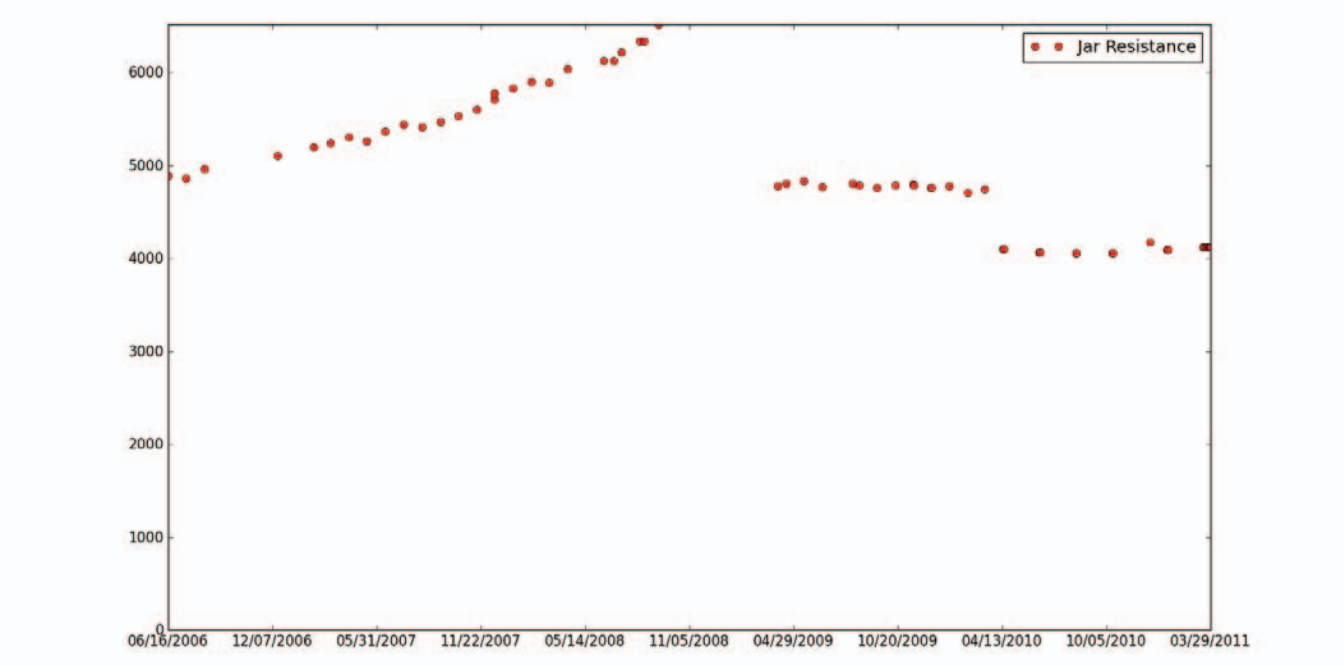

Figure 6 - Results for String Tag 1293309, String Number 6, Jar 9

Note that in some cases, such as Figure 4, the removal of outliers can also eliminate some useful values (in particular, many of the values towards the end of the first time regime, which appeared to be steadily progressing upwards without creating a true outlier). This is an inherent risk of removing instances, however, and in this case practitioners could add back the missing data as appropriate, or use this to help choose better threshold for future analysis.

Overall, we see that post-cleansing, the important traits of the data are preserved (the general pattern of how values change over time, both smoothly and in discontinuous jumps), but erroneous values which could have led to misleading results (such as duplicated values and extremely high or low values) have been removed. This will make the processed data easier to handle both by humans and with automated systems.

V. Conclusion and Discussion

In this paper, we have discussed different error types which can affect instrument data collected from remote monitoring and analysis systems, shown options for addressing these errors, and presented a case study which uses a newly-implemented tool to cleanse data collected from a real-world system. This will allow practitioners to have more trust in their data and believe that the sensor values they are looking at represent real traits of the battery system being monitored and not false values caused by malfunctioning instruments, wires or connectors. In addition, this more meaningful data can be integrated into a larger system for battery system performance monitoring.

In the future, we will examine additional types of battery instrument error and consider alternate ways to detect and resolve these errors. We will also use this data to create a model that can predict when a battery or a string is about to fail. The provided data requires significant preprocessing to prepare it for the modeling process. There are many unnecessary portions that are cleansed in the manner that was discussed in this paper. From this cleansed data, we intend to automate a pattern recognition process that generates a model that shall accurately label the unlabeled present data streaming in real time. From the labeled information generated from the models, semantic information shall be readily available, which can immediately indicate interesting events that may warrant attention.

References

[1] Ratcliff, G., “A Comprehensive Management Approach to Maximizing UPS Availability”. Battcon Conference Proceedings 2011. AlberCorp.

[2] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, “Declarative support for sensor data cleaning,” in *Pervasive Computing*, ser. Lecture Notes in Computer Science, K. Fishkin, B. Schiele, P. Nixon, and A. Quigley, Eds. Springer Berlin Heidelberg, 2006, vol. 3968, pp. 83–100. [Online]. Available: http://dx.doi.org/10.1007/11748625 6

[3] S. Jeffery, G. Alonso, M. Franklin, W. Hong, and J. Widom, “A pipelined framework for online cleaning of sensor data streams,” in *Proceedings of the 22nd International Conference on Data Engineering, 2006.*, April 2006, p. 140.

[4] A. Klein, “Incorporating quality aspects in sensor data streams,” in *Proceedings of the ACM First Ph.D. Workshop in CIKM*, ser. PIKM ’07. New York, NY, USA: ACM, 2007, pp. 77–84. [Online]. Available: http://doi.acm.org/10.1145/1316874.1316888.