*Methods of Data Quality Analysis and Detection for OPC Based Wind Farm SCADA Systems*

Hector Lopez  
*Florida Atlantic University*  
*NextEra Energy Resources*Jupiter, FL. United States

Dr.Yufei Tang  
*Florida Atlantic University*Boca Raton, FL. United States

*NextEra Energy Resources*Jupiter, FL. United States

*Abstract*—In the Internet of things (IoT), data gathered from a global-scale deployment of smart-things are the base for making intelligent decisions and providing services. This paper focuses on the IoT data from a Wind Site with many Wind Turbines and aims to implement a method for detecting and tracking data quality (DQ) of the IoT data. The paper aims to identify the definitions of the DQ dimensions specific to the domain for Wind Sites. After generating a method to detect data quality a dataset from a live wind site is used to determine the effectiveness of the DQ strategy. Data capture settings are adjusted to understand the tradeoffs between DQ dimensions. Data is also categorized and analyzed in batches to understand the DQ of the captured dataset. The results explore any improvements or actionable information that can be used to improve the DQ. In conclusion the possibility of future research and enhancements is presented.

Keywords—IoT, Data Quality, Smart Grid, Wind Turbines, Cloud, Statistical Detection, Machine Learning, Infrastructure

# Introduction

Wind power is the most growing renewable source (Appavou et al. (2016)), however the operation and maintenance of the wind turbines account for 25%–35% of the generation costs (Milborrow (2003)). In order to increase the economic competitiveness with respect to fossil fuels and accelerate the transition towards ecologically sustainable systems, there is a need for a more efficient management and this requires better monitoring of wind turbines. The trend of operational technology such as industrial control systems has moved towards more open communication between devices. The machine to machine communication across networks has increased data transfer and data consumption by users and automated operational systems. Modern wind turbines record more than 1000 variables at intervals of 50ms to 10 mins by means of their SCADA (Supervisory Control and Data Acquisition) system.

The SCADA system collects data from different parts of the turbine, which are grouped into systems (Vestas R+D (2004)) and providing information about: temperatures, electrical indicators, physical posi- tions, speeds, vibration, etc. These systems generate a huge amount of data (Voev and Siemens, 2014), which has to be preprocessed and modeled in a feasible time (Justin Heinermann (2015)). Due to this, cloud-based platform services seem to be a better option to process SCADA data taking into account its processing scalability (Rodero- Merino (2011)) and high availability to build prediction engines for early diagnosis (called prognosis), then improving the efficiency of the wind farms, which translates in the reduction of generation costs (Besnard and B, 2010).

These prognosis systems have to work in a wide set of wind turbines, with different models, manufacturers and SCADA- data configurations. Raw data obtained from SCADA contains several kinds of errors categorized as: missed data caused by communications failures, presence of extreme values due to sensors failures, data coming from poorly calibrated sensors or by replaced sensors which report outputs in a different range, errors in the SCADA system or even human errors (Gray (2011)).

Cloud-based platforms can be heavily leveraged to introduce greater statistical methods to improve the data quality of the raw data. The cloud based platforms also provide a new channel to consume modern sensors outside of traditional SCADA systems. The new devices that capture more raw data can help in providing deeper insights on operational technologies at wind farms. The growth of interconnected devices has skyrocketed in the recent years , lending to a surge in data capture and automated action without the need of human intervention. This phenomenon of interconnected devices communicating from machine to machine is called M2M or Internet of Things (IoT). s

# Windfarm Communication Architecture

## Overview of Industrial IoT

The Industrial Internet of Things (IIoT) goes beyond the normal consumer devices and internetworking of physical devices usually associated with the IoT. What makes it distinct is the intersection of information technology (IT) and operational technology (OT). OT refers to the networking of operational processes and [industrial control systems](https://www.trendmicro.com/vinfo/us/security/definition/industrial-control-system) (ICSs), including human machine interfaces (HMIs), supervisory control and data acquisition (SCADA) systems, distributed control systems (DCSs), and programmable logic controllers (PLCs).

These operational technologies are found in power generation systems such as wind and solar farms.

By integrating IIoT into these power generation systems the operational capabilities increase and drive improved processes that can generate cheaper renewable energy for consumers. Increases in data consumption from real-time sensors can lead to increases in visibility, insights and specific actions that enable the general office to bridge the gap with site operations. In the information age the data driven decisions from the IIoT allow these systems to become automated and more responsive to the needs of each interconnecting “thing”, allowing an ecosystem of devices to operate efficiently.

The paper seeks to provide a design architecture for capturing operational data from a wind site and apply methods to optimize data quality given real world constraints, such as network bandwidth, protocol limitations, connection losses, corrupted data, computational resources, noise, etc. The greatest challenge to IIoT is the optimization of data gathering. The constraints of data collection need to be mitigated. The more data collected the more potential for insights but can become a costly endeavor. By understanding the opposing constraints, a control “cost-to-go” function can be leveraged to optimize data collection properties when using ICS protocols.

## Real-World Data

The GE WindSCADA 10.0 SP2 did not use the IEC standard described for the GE 1.5 ESS wind turbines. The SIEMENS WPS 2.0 SCADA also did not use the IEC standard for its wind turbines. These systems need to have IEC standard mapped to the OEM naming convention in order to support a cohesive standard. SCADA data that follows the IEC 61400–25 format IEC (2006) uses a hierarchical structure of turbines (Logical devices) and physical systems (Logical nodes).

A sample set of 57 tags where used from a real wind farm controlled by GE’s WindSCADA 10.0. The site contains several turbines and many tags per turbine. By selecting these tags an analysis can be done to understand the data and convert the non-standard naming convention to a IEC 61400-25 standard. The benefit in standardizing the tags is a better description on what the behavior and classification of the data should be. By knowing the behavior of the data , better prognosis can be done on the data set.

#### The proposed wind farm information displays the type of SCADA, wind turbine technology and other specficiations about the wind farm that will be analyzed for data quality enhancments.

| Wind Farm Information | | | |
| --- | --- | --- | --- |
| ***# WTGs*** | ***Site MW*** | ***Base Rating*** | ***Rotor*** |
| 37 | 59.9 | 1.62 | 100 |
| ***Blade Types*** | ***Pitch Type*** | ***Tower Height*** | ***Controller*** |
| Glass | Salem | 80 | MarkVI |
| ***Converter Type*** | ***Software Version*** | ***SCADA Version*** | ***Converter Type*** |
| GE ESS | 44.76.00C | WindSCADA 10.0 SP2 | GE ESS |

## Data Acquisition (SCADA)

The required data to be transmitted from the wind farm to the SCADA system is divided in four categories:

* Substation data (statuses, alarms, meters, etc.)
* Weather towers data
* Turbines data
* Production data.

To achieve these requirements, a wind farm communication system has been proposed as described by Fig. 1. A data concentrator collects the data from multiple sources using TCP/IP or RS232/485 connections. The wind turbines and weather towers are connected to the substation using optic fiber connection. The wind turbines are either connected to a proprietary server which is connected to the data concentrator or directly connected to the substation network. In addition, the data concentrator collects the information of intelligent electrical devices such as meters, protection equipment and power quality modules. The collected data will be subsequently published to different data clients such as distribution grid SCADA, historian servers and local (Human Machine Interface) HMI.

Furthermore, the data described above can be delivered to end users in two different forms: real time data, which is basically a simple transmission of the acquired information from all the devices in the substation or computed from this data (for key indicator information as availability turbines counters, availability power at the wind plant, etc.) and statistical data which is calculated from real time data and transmitted periodically to SCADA systems (ref). This data is used by the utility to build up an historical database which is used for forecasting and impact analysis purposes (ref).

This paper will focus only on the data collected at the turbine level for simplicity.

1. Typical Windfarm communication archtictecture

#### IEC 61400-25: A Brief Overview: According to IEC61400-25 User Group [2], the IEC standard series 61400-25 provides a solution for access to wind power plant information with standardized data names and semantic. It gives possibilities to procure monitoring and control solutions as separate parts, and to use a single system to store, analyze and present wind power information. In addition, the standard opens up for control and monitoring of information from different wind turbine vendors in a homogeneous manner.

The approach of the standard is to decompose the application functions into the smallest entities, which are used to exchange information. These entities are called logical node (LN). A LN consists of a gathering of related data defined as Data Classes (DC). All the information in a logical node is contained in the respective DC. The structures of all LN is similar and has a standardized form where different types of LN can be constructed through the combination of different optional DC.[3][4].

1. IEC 61400-25 Logical Nodes for Wind Turbine Components

#### Converting to IEC-61400-25 Standard: The GE SCADA system provides data in a naming convention that does not follow the IEC standard. By not having a hierarchal naming convention like the one laid out in the IEC-61400-25 it is very difficult to begin contextual analysis.

## Data Gathering (OPC)

The Supervisory Control and Data Acquisition (SCADA) server is a computer system that hosts a software from the wind turbine OEM that polls the data from the turbines and then sends information to the entire wind farm in order to control the overall operation of the site. The SCADA server provides a way for users to extract real-time data from the system in order to analyze and record the data. The protocol used to expose this data can be any of the standard Industrial Control System(ICS) protocols, such as DNP3, OPC, Modbus, etc. The protocol that shall be discussed in this paper is OPC. Data is gathered via an OPC(OLE for Process Control) (OPC Foundation (2016)) with update periods of 5–10min, producing several types of indicators. Only failure events and statistical indicators are kept.

An OPC Connection requires an OPC client such as a free utility called “Matrikon OPC”. This application allows a user to connect to a local or remote OPC server and browse the tags by creating an OPC group with some connection properties. The group allows settings that can affect the collection of the data. Update Rate, Percent Deadband , Async/Sync , Time Bias . By focusing on these group setting changes to the capture of the data can be affected. Intrinsic data quality measurements can suffer if these settings are not tuned properly. The goal is to optimize these settings in order to maximize the data quality and capture of critical tags.

## Data Ingestion (Cloud)

A software application that can collect OPC data in a persistent way is required at the site level in order to push that data to cloud services that can distribute the data into data storage and consumption services. A proposed software model is presented. Highlighting the components needed to configure, subscribe, collect, cache and push to a cloud-based service over a IoT standard protocol called MQTT (message queue telemetry transport).

1. Proposed Software Model of Data Logging Application

# IoT at Wind Farms

[TODO]

## Metrics

* + *Time series, Time-stamped observations*
  + *Context*

## Static Attributes

* + *Units*
    - *Bytes*
    - *Requests*
  + *Data Type*
    - *Counter*
    - *Gauge*
  + *Granularity*
  + *Raw or Cooked*

## Dynamic Attributes

* + *Slowly Changing Dimensions (SCD’s)*
    - *Not Constant, but not always changing*
    - *Hosts in cluster*
    - *Number of replicated microservices, etc.*

## Collections

* + *Entities/Elements*
    - *Virtual or Physical*
    - *Dynamic attributes as time series represented as states as it moves through time*
    - *Example, a javamethodcall over time*
  + *Relationships*
    - *Containment, ie. Load Balanced Cluster*
    - *Sequencing, “Workflow”*
  + *Interactions*
    - *Business level work flow*

# Data Quality Dimensions

## Intrinsic Data Quality

Intrinsic DQ dimensions can be directly measured from data collection system.[TODO]

## Contextual Data Quality

Contextual DQ Dimensions require insights, benchmarks, analysis and historical data to measure. [TODO]

## Definitions of Data Quality

There are many forms of Data Quality dimensions and it varies due to the domain that the data quality is being measured and consumed. The focus of this paper is to define the major dimensions and the subsequent categories that a SCADA system and IIoT enabled wind farm would be able to track effectively. The idea of consumption is also important since data quality can be highly subjective to the end user. The major dimensions that are pertinent to provide metrics for maintaining and provisioning a IIoT pipeline are proposed in Table I.

| Data Quality Dimension Definitions | | | |
| --- | --- | --- | --- |
| Data Quality Dimension | Class | Properties | Description |
| Connectivity | Intrinsic | Infrastructure, Application, Site | Measure of source availability while collecting data |
| Completeness | Intrinsic, Contextual | Critical Data, Dark Data, Data Leakage | Measure of data collected from a specified data set |
| Timeliness | Intrinsic | Processing Latency, Synchronization | Data synchronized across all sources in spite of time stamp capture or system error |
| Accuracy | Intrinsic, Contextual | Sampling Resolution, Precision, Filtering | Measure of how accurate the data captured represents the generated data (in time and value) |
| Consistency | Contextual | Batching, Smooth Variance, Presentation | Measure of data collected from a specified data set over time |

# Wind Farm Domain of Data

# Data Collection Constraints

GE OEM performance limits creates a limitation that can effect the operation of the SCADA system. The data collection must keep performance constraints in its design when increasing the number of tags to consume.

## Data Types

When determining accuracy of a value the data type is important to understand the convention used to represent a value. The OPC protocol uses a Microsoft VB data type definition called Variant. The Variants have varying sizes and are used differently when configured with SCADA systems.

The VARIANT types VT\_I2, I4, R4, R8, CY, DATE, BSTR, BOOL, UI1 as well as single arrays of these types (VT\_ARRAY) are expected to be most commonly used (in part because these are the legal types in Visual Basic).

It is recommended that whenever possible, clients request data in one of these formats and that whenever possible, servers be prepared to return data in one of these formats. It has been found in practice that some servers (for example those connecting to remote locations) are unable to determine the Native Datatype at the time an item is added or validated. It has become common practice for such servers to return VT\_EMPTY as the native data type. Such servers will retain the requested type (which may also be VT\_EMPTY) and will return the data in the requested type (which may be 'Native') when the data becomes available and they are able to determine its actual type

| OPC Data Types | | |
| --- | --- | --- |
| ***Type*** | ***Range*** | ***Size*** |
| Integer | -2147483648..2147483647 | signed 32-bit |
| Real | 1.5 x 10^-45 .. 3.4 x 10^38 | signed 32-bit |
| Double | 5.0 x 10^-324 .. 1.7 x 10^308 | signed 64-bit |
| DWORD | 0..4294967295 | unsigned 32-bit |
| WORD | 0… 65535 | unsigned 16 bit |
| BYTE | 0…255 | unsigned 8 bit |
| BSTR | Pointer to TCHAR array LPCTSTR | 64-bit |
| Date | Pointer to TCHAR array LPCTSTR | 64-bit |
| I2 | -2147483648..2147483647 | signed 32-bit |
| I4 | 1.5 x 10^-45 .. 3.4 x 10^38 | signed 32-bit |
| UI4 | 0..4294967295 | unsigned 32-bit |
| R8 | 5.0 x 10^-324 .. 1.7 x 10^308 | signed 64-bit |

## Data Update Rate

When a client connects to the OPC DA server using an OPC DA 2.0 connection, the variable values in a group are updated once when the group goes active, and again when a variable changes. The update on change only contains the variables that changed since the last update. In addition, OPC DA 2.0 allows for a group deadband. When any variable changes by more than that deadband, the variable is updated to the client. OPC DA 3.0 also allows a client to establish a deadband per variable. The client requests an update rate when adding a group. The OPC DA server replies with the selected rate, which is the closest multiple of the Maximum Client Rate configured in the WorkstationST OPC DA server tab, or in the Runtime Monitor Config Options menu. For example, if the Maximum Client Rate is set to 100 ms and the client requests a rate of 80 ms, the client is given a rate of 100. If the client requests 160 ms, the client is given a rate of 200 ms. The actual update rate of the variables in a group depends on the rate that the variable is being updated to the OPC DA server. EGD variables are updated at the EGD exchange rate. For SDI variables, the SDI live list is requested to the controller at the group rate. In server performance testing.

* 5000 Boolean variables changed at 640 ms, and updated on one EGD exchange at 1000 ms
* 10000 floating point variables changed at 32 ms, and updated on 40 EGD exchanges at 1000 ms
* 100 floating point variables changed at 32 ms, and updated on one EGD exchange at 100 ms

The server maximum client connection rate was set to 10 ms and one client with one group was connected with a rate of 100 ms. With the client connected, the OPC DA server used between 20 and 30 percent of a Pentium® 4 2.6 GHz CPU. Without the client connected, the CPU utilization was around 10 percent.

## Dead Bands

The percent Deadband impacts the Data Quality Accuracy Dimensions. The range of the Deadband is from 0.0 to 100.0 Percent. Deadband will only apply to analog items. The EU Low and EU High values for the item can be used to calculate the range for the item. This range will be multiplied with the Deadband to generate an exception limit. An exception is determined as follows:

*Exception : if (absolute value of (last cached value - current value) > (pPercentDeadband/100.0) \* (EU High- EULow)*

The Percent Deadband can be set when AddGroup is called, allowing the same Percent Deadband to be used for all items within that particular group. However, with OPC DA 3.0, it is allowable to set the Percent Deadband on a per item basis. This means that each item can potentially override the Percent Deadband set for the group it resides within If the exception limit is exceeded, then the last cached value is updated with the new value and a notification will be sent to the client’s call back (if any). The Percent Deadband is used to keep noisy signals from updating the client unnecessarily.

# Data Quality Analysis

An overall intrinsic Data Quality metric can be given to the data collection phase because known variables can be adjusted to minimize the quality that is needed for a set of tags. This analysis can be used to tweak the factors necessary and maximize the intrinsic data quality as it is provided to the data storage and prognosis phase.

## Data Connectivity

The connectivity of the data can be understood in the OPC protocol as a valid connection to collect data. With this quality dimension the focus is on keeping track of how the data is not connected at any given point in time. By leveraging OPC quality codes one can begin to track the possible failure modes of each of the opc items.

The OPC quality code is made up of 16 bits. The high 8 bits are avalible for vendor specific use and should be all 0's when not used.The low 8 bits are broken into three sections. The first two bits can pass the meaning Good, Bad or Uncertain. If a server does not support quality codes, a good value will always be passed. The next four bits can be used to provide additional specific information about the general quality. The last two bits are used if limit information is supported by the server.

In VB these are returned as Hex values that can them be converted to meaningful information. The two most common OPC quality codes are:

• 192 or hex C0 is good quality.

• 0 (decimal or hex) is bad quality

| OPC *Error Code* | |
| --- | --- |
| ***OPC Error Code*** | ***Description*** |
| OPC\_E\_BADTYPE | The passed data type can not be accepted for this item from server |
| OPC\_E\_BADRIGHTS | The Item is not having either Readable or writable access rights |
| OPC\_E\_RANGE | The value was out of Range |
| OPC\_E\_INVALIDHANDLE | Clients Item handle is invalid when requested to server |
| E\_NOINTERFACE | The possible version conflict between the OPC DA server version and OPC Client version while communicating |
| OPC\_E\_UNKNOWNITEMID | The Item ID is not part of OPC Namespace in the OPCDA server |
| OPC\_E\_INVALIDITEMID | The client requested item Name has invalid convention (for ex some invalid characters) |
| OPC\_E\_DUPLICATENAME | Trying to add a group which is already present in server |
| OPC\_E\_NOTSUPPORTED | If a Client attempts to write any value, quality,timestamp combination and the server does not support the requested combination(which could be a single quantity such as just timestamp), then the server will not perform any write and will return this error code |
| E\_OUTOFMEMORY | Not Enough memory to complete the requested operation. This can happen any time the server needs to allocate memory to complete the requested operation |
| E\_FAIL |  |
| OPC\_S\_CLAMP | The Value was accepted but was clamped |
| E\_INVALIDARG | An invalid argument was passed(like when client requests data to server the argument of dwcount should be >0 but if dwcount=0 then this error code will be returned |
| CONNECTION\_E\_CONNECTIONT | The client has not registered it communication channel with server for the data updation |
| OPC\_E\_DEADBANDNOTSUPPORTED | The dead band is not supported by the server |
| OPC\_S\_UNSUPPORTABLERATE | Server does not support requested rate,server returns the rate that it can support in the revised sampling rate |
| OPC\_E\_NOBUFFERING | The server does not support buffering of data items that are collected at a faster rate than a group update rate |
| OPC\_E\_UNKNOWNPATH | The Item’s access path is not known to the server |
| OPC\_S\_INUSE | The operation cannot be performed because the object is being referenced |

## Data Completeness

Define what a critical set of tags out of a full set of tags would be defined as completeness. The more tags the more data is ingested. This is a dimension that would be impacted if the data ingestion needs to be decreased. Completeness can be heavily impacted by connectivity. The difference between connectivity and completeness is that completeness is heavily dependent on the needs of the user. Different customers may ask for differing requests. Completeness is a metric that would capture that business case and track the complete subset of data for each customer.

## Data Accuracy

Accuracy in the data involves two major components, the Resolution of the data that deals with the time dimension and the Precision of the data that deals with the space dimension. The resolution can be measured by performing an analysis over many snapshots at an arbitrary window size. Choosing a window size of 1 minute helps to standardize the unit of measure when discussing resolution. As discussed in other sections of the paper the limitation of data points is usually the update rate that limits changes of a value typically to 1 second. This change rate would make it reasonable to measure resolution in minutes. By analyzing multiple windows over another arbitrary period of time and averaging over the samples allows one to view what the resolution is on average for a given tag. A window of 1 minute was chosen with 49 windows taken over the course of 48 hours. The windows where taken periodically with an hour apart. The resolution metrics of each window was averaged for each tag. The standard deviation was used to create a Margin of error with a confidence metric of 80%.

1. Average data points per minute captured 49 times over the course of 48 hours at the top of each hour, showing resolution of critical SCADA tags for an arbitrary wind turbine at a running wind farm

## Data Timeliness

Define Timelines as the latency across an ingestion , storage and consumption pipeline that occurs due to the mechanisims of the pipeline. The timeliness quality is not just latency but in a SCADA environment it also tracks the time stamp of when the data was captured, received recorded and consumed. These varying timestamps must be synchronized over a common time server. The challenges of having segregated networks is that there is no common time sync . Many time sthe timestamps will become out of sync causing impact to the validation of the origin time.

# Proposed Data Pipeline

The data ingestion begins at the SCADA system but must then be encapsulated and transferred into storage and eventually consumed by users. The proposed data pipeline tries to touch on an example of how the data can be ingested at the SCADA system then controlled to avoid common situations when dealing with industrial IoT systems. The next steps of the pipeline involve the packaging and storing of the data in an encoded form to reduce on network bandwidth, CPU, memory and costs in transferring data into a cloud platform. A cloud platform is chosen in the proposed model due to the rapid adoption of elastic computing and the capabilities to perform advanced statistical analysis in the cloud environment that would lend to deeper data quality insights from the collected data. Finally, the pipeline takes into account the consumption of the users by measuring the latency of the intrinsic data and the rendering time due to searching the data from a data storage solution. Cloud

## Data Pipeline Optimization

A point anomaly is an observation that is unusual when compared with all the rest of available observations.

## Optimization of Data Pipeline

A point anomaly is an observation that is unusual when compared with all the rest of available observations.

*a**b* 

# Data Anomoly Detection

The data quality of a set of data can be categorized for anomaly detecton. There are different types of anomaly detection that can be applied to the different levels of data collection.

## Point

A point anomaly is an observation that is unusual when compared with all the rest of available observations.

## Contextual

A contextual anomaly is an observation that is unusual in a certain context but not in other contexts.

## Collective

A collective anomaly occurs when a collection of related data instances is anomalous with respect to the entire data set. Correlation of multivariate variables

| Data Anomaly Detection Types | | | |
| --- | --- | --- | --- |
| ***Anomaly type*** | ***Detection Data Requirement*** | ***Anomaly type*** | ***Detection Data Requirement*** |
| Point | Uni-variate | Point | Uni-variate |
| Contextual | Uni-variate + contextual attributes | Contextual | Uni-variate + contextual attributes |
| Collective | Multi-variate+ contextual attributes | Collective | Multi-variate+ contextual attributes |

## Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus ( / ), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

*a**b* 

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

##### Acknowledgment

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. Sunaina Sulthana Sk, Geethika Thatiparthi, Raghavendra S Gunturi Cloud and Intelligent Based SCADA Technology,International Journal of Computer Science and Electronics Engineering, Volume 2, Issue 3 , March 2013.
2. Shyam R, Bharathi Ganesh HB, Sachin Kumar S,Prabaharan Poornachandran, Soman K P, Apache Spark a Big Data Analytics Platform for Smart Grid , SMART GRID Technologies, August 6-8, 2015, 2212-0173 © 2015 Published by Elsevier Ltd.
3. Elisa Bertino and Hyo-Sang Lim , “Assuring Data Trustworthiness - Concepts and Research Challenges⋆” Department of Computer Science, Purdue University, USA, {bertino,hslim}@cs.purdue.edu, W. Jonker and M. Petkovi ́c (Eds.): SDM 2010, LNCS 6358, pp. 1–12, 2010. Springer-Verlag Berlin Heidelberg 2010.
4. David Yates, Jim Kurose,Prashant Shenoy, Data Quality and Query Cost in Wireless Sensor Networks, Proceedings of the Fifth Annual lEEE International Conference on Pervasive Computing and CommunicationsWorkshops (PerComW'07) 2007 IEEE
5. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.