Big Data Techniques For Supporting Prediction of Renewable Energy Production

Michelangelo Ceci
UNIBA
Via Orabona, 4, Bari Italy
michelangelo.ceci@uniba.it

Donato Malerba
UNIBA
Via Orabona, 4, Bari Italy
donato.malerba@uniba.it

Giuseppe Manco
ICAR-CNR
Via P. Bucci, Rende, Italy
manco@icar.cnr.it

Elio Masciari
ICAR-CNR
Via P. Bucci, Rende, Italy
masciari@icar.cnr.it

Aleksandra Rashkovska
UNIBA
Via Orabona, 4, Bari Italy
aleksandra.rashkovska@uniba.it

ABSTRACT

Predicting the output power of renewable energy production plants distributed on a wide territory is a valuable goal, both for marketing and for energy management purposes. *Vi-POC* (Virtual Power Operating Center) project aims at designing and implementing a prototype able to achieve this goal. Due to the heterogeneity and the high volume of data, it is necessary to exploit suitable Big Data techniques to perform quick and secure access to data, which cannot be obtained with traditional approaches for data management. In this paper, we describe Vi-POC – a distributed system for storing huge amounts of data, gathered from energy production plants and weather prediction services. We use HBase over Hadoop framework on a cluster of commodity servers in order to provide a system that can be used as a basis for running machine learning algorithms. In particular, we perform one-day ahead forecast of PV energy production based on Artificial Neural Networks in two learning settings, that is, structured and non-structured output prediction. Preliminary experimental results confirm the validity of the approach.

1. INTRODUCTION

Recently, renewable energy research is gathering a lot of attention due to the strategic and urgent need of reducing pollution emission and finding new revenue streams for utility companies. Indeed, wholesale vendors lead the utility sector because those companies provide energy to the most relevant share of private and industrial users. Due to their dominant position, they are able to gather huge amount of valuable information. In particular, they have access to both external and internal data, including sensor data from producing assets, real-time or end-of-day price data from a multitude of related markets, counter-party credit data, position management information, and many others. However, due to the availability of new (low cost) technologies, also small producers are able to collect data about their business. Indeed, data coming from small production plants are quite heterogeneous, they arrive continuously (fast) and their volume increases at an unprecedented growth rate. These features pose several challenges that can be solved using Big Data techniques [1, 2, 3]. Moreover, these challenges are crucial for achieving several business objectives, such as reducing enterprise risk and shortening decision response times, thus enabling traders and decision makers to quickly react to sudden changes of market quotations. Furthermore, Big Data techniques can help management staff to maximize company returns both in short and long time horizons.

In this perspective, Vi-POC project has been developed in order to support renewable energy providers with a framework for collecting, storing, analyzing, querying and retrieving data coming from heterogeneous renewable energy production plants (such as photovoltaic, wind, geothermal, Sterling engine, water running) distributed on a wide territory. Moreover, Vi-POC features an innovative system for real-time prediction of the energy production, integrating data coming from production plants and weather prediction services. Indeed, a key problem for low-business energy producers is the exact quantification of the amount of energy that can be pushed over the power supply network. This problem arises as energy cannot be stocked efficiently; thus, they are forced to give it for free if they produce more energy than the network can use or they will pay a huge penalty if they do not provide the expected amount.

In this paper, we describe our end-to-end framework that allows a better engineering of data structures in order to support data analysis and prediction of energy production, thus offering a good trade-off between effective storage and efficient analysis of data. As for the prediction of the energy production, we propose a method for long-term forecast (one-day ahead) of photovoltaic energy production based on Artificial Neural Networks (ANN) and investigate the performance in two settings - structured and non-structured output prediction. While in non-structured output prediction a prediction model generates the forecast for a specific hour in the future, in structured output prediction, a prediction model generates the forecast for 24 hours in the future [15][5]. In principle, the main advantage of structured output prediction consists in the implicit consideration of the dependence of the predictions at two consecutive hours.

In [10, 9], we described the high-level architecture of our system. In this paper, we will describe the actual implementation of the prototype along with the implementation issues that are crucial for building a system for Big Data management and analysis.

2. BACKGROUND

Nowadays, dealing with a big volume of data is very challenging because traditional technologies, like RDBMS or classical object oriented programming, are not well suited for the typical size and scalability requirements of Big Data. In order to meet these requirements, column oriented databases have been proposed. In our project, we exploited a *HBase* storage system.

HBase is an open source, non-relational, distributed database modeled as Google BigTable and developed in Java. More in de-

tail, it is an Apache project and runs on top of HDFS, providing BigTable-like capabilities for Hadoop, i.e., it provides a fault-tolerant way of storing large quantities of sparse data. HBase main features are:

- good compression performances;
- in-memory operation execution;
- bloom filters on a per-column basis as in a BigTable specification.

Tables in HBase are used to perform Input and Output for MapReduce jobs running on Hadoop, and may be accessed through the Java API, but also through REST, Avro or Thrift gateway APIs. It is worth noting that HBase is not a column-oriented database in the typical RDBMS sense, but utilizes an on-disk column storage format. Rows are composed of columns, and those, in turn, are grouped into column families in order to build semantical or topical boundaries between the data, as shown in Figure 1. Furthermore, the latter data organization makes it possible to improve compression or specific in-memory operation.

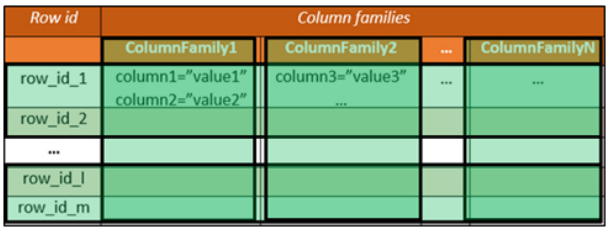


Figure 1: HBase storage organization

Columns are referenced as family having a qualifier represented as an array of bytes. Each column value (or cell) is either implicitly timestamped by the system or can be set explicitly by the user. Rows in the tables are sorted by a *row key* and this key provides access to information contained in the row. On the other side, columns are grouped into *column families* and can be updated at runtime (by specifying the column family through a prefix). Indeed, this model turns to be efficient and scalable, thus well suited for Big Data management. On the contrary, a row based approach is inefficient, and simple column based approaches are efficient but not scalable. Figure 2 summarizes the features and the difference among the approaches.

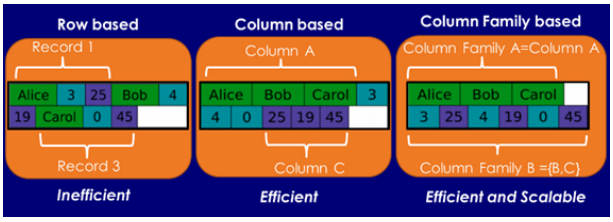


Figure 2: Features of several storage models

At the physical level, all columns in a column family are stored together in the same low level storage file, called an HFile. In addition to the notion of the column, table and row, HBase uses the so called "region". In fact, the HBase tables are automatically partitioned horizontally into regions that are distributed in a cluster. Each region consists of a subset of rows of a table and in this way a table that is too large to be contained in a server can be distributed on different servers in the cluster.

The HBase data model is "sparse, distributed, persistent, multi-dimensional sorted map". More in detail, data are *sparse* as they do not explicitly represent *null* values. HBase *distributed* and *persistent* features are guaranteed by automatically storing data in a redundant way through exploiting a specialized distributed file system as HDFS, that spreads data across different machines usually representing different nodes of a given cluster. Moreover, data are stored in a *multi-dimensional* map for fast indexing by row key, column and version. Finally, data are lexicographically *sorted* by row key. Row-key and column-qualifier can be of arbitrary type (i.e. raw bytes) while column family qualifier must be composed only of standard characters. Version identifier is a long integer, usually representing the time stamp of value insertion in the map.

However, this data model lacks some useful operations available for classical RDBMS solutions, like joins, foreign keys, referential integrity and transaction support. If the application being implemented requires these features, they need to be implemented ad-hoc. As for transaction support, although the CAP theorem holds, that is, it is not possible to guarantee both consistency and availability while partitioning data in a distributed system, HBase is partition-tolerant and consistent (*CP*).

3. RENEWABLE ENERGY CASE STUDY

The Vi-POC project aims at designing and implementing a prototype able to manage renewable energy production plants distributed over national territory. Vi-POC implements an innovative system for real-time prediction of the energy production. It exploits Big Data techniques in order to deal with the heterogeneity of data coming from different sources such as photovoltaic (PV), wind, geothermal, Sterling engine, and water running. Vi-POC is intended to predict real-time energy production with higher precision as it exploit historical information about production and weather conditions. The high accuracy and efficiency will allow energy market operators to implement a more effective purchasing strategy.

We exploited a HBase storage system designed for storing weather information and plant sensor data. The data is exploited by clients running data mining algorithms to predict output power of plants. Every plant sends periodically all the data collected by installed sensors. The time granularity is set based on the type and the dimension of the plant. Data coming from plants usually consists of different measures, gathered from several sensors at a given timestamp. Indeed, the number and the type of sensors may differ among plants. Forecast data instead, consists of various predicted weather parameters forecasted for a given time and location.

Our architecture stores the data on a HBase system consisting of three tables: one for storing plants information, one for storing measurements from plants and one for storing weather forecast information. To store data regarding a location, we use Geohash[1]. It is a standard way to represent latitude/longitude information as a string of characters having very useful properties. As an example, sites close to each another share the same prefix in the string.

HBase performances heavily decrease when more than three column families are used. This is exhibited because flushing and compaction are performed on a per-region basis, thus, if a column family is carrying the bulk of the data being flushed, the adjacent families will also be flushed even though the amount of data they carry is small. As a consequence, when many column families are exploited, the flushing and compaction interaction can heavily de-

[1]www.geohash.org

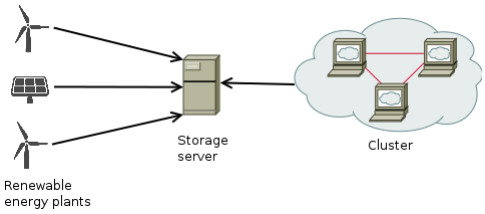


Figure 3: System Architecture

crease system performances. In this respect, we designed column family schemes having at most two column families.

3.1 System Architecture

As depicted in Figure 3, we can see the interactions between the different subsystems in our architecture. As stated before, there are many renewable energy plants that send data periodically to the system. Separation between the plants and the computation cluster is a key concept. The plants, in fact, do not send their measurements directly to the computation cluster, but to a separated storage level made of several file servers. Different fault tolerance strategies are applied among these levels in order to avoid blockage of the entire system due to the failure of one of the components. Data is then taken by computation cluster's *Extract, Transform and Load (ETL)* tool and stored in a non-relational distributed database.

Figure 4 shows the software architecture implemented on a cluster of computing nodes. It is composed of several levels. As stated before, we use HDFS as distributed file system. Data is stored on different commodity machines in the computation cluster. We plan to simplify the software setup of the computation cluster through the use of software containers (in particular Dockers [2]), thus providing platform as a service (PaaS) style deployment.

On top of HDFS we run HBase, which provides BigTable-like capabilities for Hadoop. The large quantities of data, coming from renewable energy plants, are stored in a fault-tolerant way across the nodes of the cluster. The tables in HBase serve as the input and output for MapReduce jobs. We use *Apache ZooKeeper* that provides services like distributed configuration, synchronization and naming registry. Cloudera Distribution including Apache Hadoop [3](*CDH*) offers a quick way to deploy all of the above components.

We wrote a custom ETL tool which manages the interaction between the storage level and the computation cluster. The tool periodically downloads the new data from the storage servers. This data is in csv format and needs to be transformed in order to be stored, according to the schema discussed in Section 3.2. The ETL tool provides this transformation and the subsequent upload to HBase tables. The definition of queries on data and the visualization of results are made by another custom tool that stands on top of our architecture.

3.2 Table schemas

Based on the above considerations, we designed tables described below (we do not report the actual name of each attribute as they are coded by the plant owner and they are not easily understandable).

Table *Plants*:

- *RowKey*: concatenation of the type of the plant (solar, wind, hydroelectric) and a plant identifier;

[2]https://www.docker.com/

[3]http://www.cloudera.com/content/cloudera/en/home.html

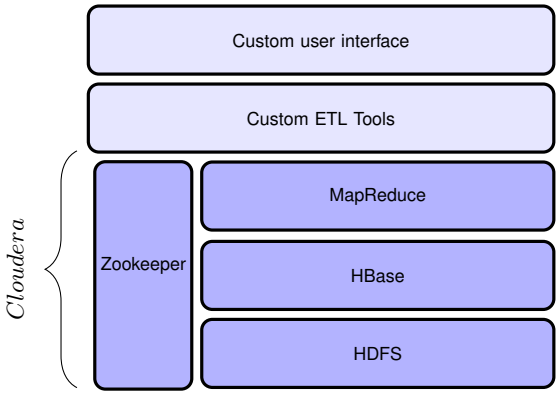


Figure 4: Architecture layers

- *Column family 1*: contains as many attributes as the cardinality of data. Every attribute represents raw information about the configuration parameters or the coordinates of the plant;
- *Column family 2*: stores log information about maintenance operations for the specific plant.

Table *Measure*:

- *RowKey*: concatenation of the identifier of the plant, the reverse time-stamp and the measurement type;
- *Column family*: stores all collected measurements. The number of attributes is equal to the cardinality of counters being collected.

Table *Predicted Measure*:

- *RowKey*: the same structure as the table *Measure*;
- *Column family*: stores the values predicted by mining algorithms. The number of attributes is equal to the cardinality of predicted data.

Table *Weather Data*:

- *RowKey*: concatenation of Geohash, reverse time-stamp, measurement type and server identifier (used to trace which server sent the prediction);
- *Column Family 1*: stores collected weather data.
- *Column Family 2*: stores predicted weather data (weather forecasts).

Figure 5 shows the implemented Hbase schema definition for representing the information described above.

3.3 Long-term forecast of PV energy production

During the last years, the forecast of PV energy production has received significant attention since photovoltaics are becoming a major source of renewable energy for the world [12]. Forecasting methods depend on the tools and information available, the forecast horizon, the number of plants considered and the size of the geographic area they cover [18]. Diverse resources are used to generate solar and PV forecasts, ranging from measured weather and

Table: Weather Data			
row key:	geohash + Reversed Timestamp + Measurament Type +Server Id		
Family:	c	column	collected data
	p	column	predicted data

Table: Plants Info				
Row key:	Type+PlantID			
Family:	c	columns:	<attribute>	<value>
	i	columns:	<timestamp>	maintenance_description

Table: Predicted Measures				
row key:	plantID+reverse_timestamp+MeasuramentType			
family:	c	columns:	<uid>	<value>

Table: Measures				
row key:	plantID+reverse_timestamp+MeasuramentType			
family:	c	columns:	<uid>	<value>

Figure 5: HBase table schemas

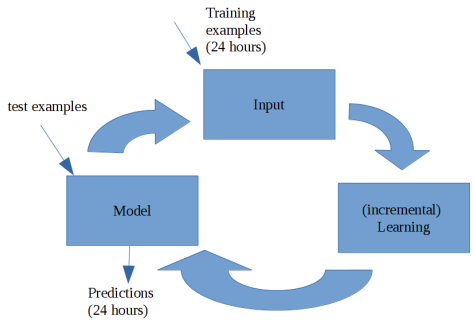


Figure 6: Our learning scheme

PV system data, satellite and sky imagery cloud observations, to Numerical Weather Prediction (NWP) models [14]. The short-term forecasts typically use measured weather and PV system data, and satellite and sky imagery observations of clouds, while the long-term forecasts use numerical weather prediction (NWP) models. The best approaches make use of both measured data and NWP models.

In the literature, several data mining approaches have been proposed for renewable energy power forecasting. We typically distinguish between physical and statistical approaches. Physical approaches deal with refining NWP forecast with physical considerations, while statistical approaches deal with building models that establish a relationship between historical values and forecasted variables. Methodologically, there are approaches based on time-series [11] and approaches that learn adaptive models [4][20].

It has been noted that physical (e.g. wind speed and solar irradiation) property behavior exhibits a trail called concept drift, i.e., they change characteristics over time [7]. In this respect, adaptive models are generally considered to produce more reliable predictions regarding concept drift, but require a continuous training phase. For example, in [16], the estimation of the model parameters is based on an exponential weighted adaptive recursive least squares controlled by a forgetting factor. A different solution is proposed in [19], where a recursive method for the estimation of the local model coefficients of a linear regression function is proposed. In this case, the time dependence of the cost function is ensured by exponential forgetting of past observations.

In [13], the author uses a stochastic gradient for online training of neural networks in wind power forecasting. Another work which uses neural networks is [6], where the authors train local recurrent neural networks of online learning algorithms based on the recursive prediction error. Bacher et al. [4] propose to forecast the average output power of rooftop PV systems by considering past measurements of the average power and NWP forecasts as inputs to an autoregressive model with exogenous input (ARX).

Sharma et al. [20] consider the impact of the weather conditions explicitly and used an SVM classifier in conjunction with a RBF kernel to predict solar irradiation. Bofinger et al. [8] propose an algorithm where the forecasts of an European weather prediction center were refined by local statistical models to obtain a fine tuned forecast. Other works on temporal modeling with applications to sustainability focus on motif mining. For example, Patnaik et al. [17] proposed a novel approach to convert multivariate time-series data into a stream of symbols and mine frequent episodes in the stream to characterize sustainable regions of operation in a data center. Finally, Chakraborty et al. [11] propose a Bayesian ensemble which involves three diverse predictors, that is, naïve Bayes, K-NN and sequence prediction.

In this case study, we propose an adaptive method for long-term forecast (one-day ahead) of PV energy production based on ANNs. The proposed approach exploits NWP to benefit from uncontrollable factors (such as weather conditions). We investigate the predictive performance of structured (all hours of the forecasted day are outputs from a single model) and non-structured output prediction models (each hour of the forecasted day is output from one model).

3.3.1 Method

The machine learning task is to predict the PV power generation using the following input attributes:

- the geographic coordinates of the plant: latitude and longitude,
- the sun positions at the location of the plant: altitude and azimuth, queried by SunPosition [4],
- the properties of the plant: site ID, brand ID, model ID, age in months,
- weather data: ambient temperature, irradiance, pressure, wind speed, wind bearing, humidity, dew point, cloud cover, and descriptive weather summary.

Additionally, in the case of structured output prediction, also the day is passed as feature, while in the case of non-structured output prediction, besides the day, also the hour. In the training phase, we use historical weather information collected by sensors, while for prediction purposes, we use weather forecast data provided by NWP systems. The output is the prediction of the power production (KWh) for the next day at one hour intervals. The prediction models are updated on a daily basis as depicted in Figure 6.

3.3.2 Data preprocessing

Since the aim is to predict the energy production at a hourly granularity, the data was aggregated so that each row represents an hour. Additional, we addressed also the issue of missing data, irregularities and outliers, and performed normalization of the data before the learning process.

[4]http://www.susdesign.com/sunposition/index.php

In order to fix completely missing hourly data points, we adopt the following approach. Missing production values in kWh are replaced by the average value observed by sensors in the same month at the same hour. Missing temperature values are replaced by historical data. Moreover, we also observed that sometimes the irradiance assumes a zero value while the plant is in a productive state. To correct irregularities of that type, we consider the average irradiance value in the same month of the same year at the same hour to replace the zero value. In any other case in which the irradiance is zero, we check if the average irradiance value in the same month of the same year at the same hour is zero too: if not, the resulting average value replaces the missing value.

After replacing missing values, we check the presence of outliers in the irradiance and temperature data. For example, if the irradiance (irr) observed by sensors is out of the range defined by [avg(irr) - 4* stddev(irr), avg(irr) + 4* stddev(irr)], this value is replaced by the average of the irradiance observed in the same month of the same year at the same hour. The same approach applies also fro handling outliers in the temperature data. Furthermore, we observed that irradiance measured locally by sensors has often lower values compared to irradiance extracted by NWP models, possibly because sensors located on plants can be covered by obstacles or dirt. Training a model by means of sensors data and using it to extract predictions with NWP data can lead to inaccurate predictions. To overcome this issue, we calculate the percentage of change between monthly NWP irradiance (extracted by PVGIS) and irradiance detected by sensors on historical data (same month at the same hour), and we normalize the latter.

In order to train the neural network, data was normalized in the range between 0 and 1. Hence, we applied a min-max normalization for each feature, considering the min and max of observed values. Actually, we considered the max increased by 30 percent, to handle future situations in which observed values of each feature might exceed the current maximum.

3.3.3 Experiments

In our empirical evaluation, we consider a real dataset collected at regular intervals of 15 minutes (measurements start at 2:00 AM and stop at 8:00 PM every day) by sensors located on 18 plants in Italy. The time period spans from January 1st, 2012 to May 4th, 2014. The weather data is queried from Forecast.io [5], while the irradiance is queried from PVGIS [6]. As anticipated before, in order to resolve measurement errors, the raw data are preprocessed and normalized before using them for learning.

In this paper, we use the *encog* implementation of the Resilient Propagation (RPROP+) algorithm for training neural networks. RPROP+ is one of the best general-purpose neural network training methods implementing the back-propagation technique. We use RPROP+ since it has been proven effective for renewable energy prediction [7]. For the evaluation, the dataset is randomly split into training days (85%) and testing days (15%). Experiments are run three times and average results are collected. For each run, the network is trained incrementally on the training dataset until a testing day is found. Then, it is repeatedly first tested on the testing day and after that it is re-trained with the tested sample added to the training set, together with all the training days before the next testing day. At the end, the average performance over all the test samples is reported as a result.

We distinguish between hourly (non-structured) and daily (structured output) settings. In the hourly setting, we investigate non-structured models with single output - the production of the plant at a specified day and specified hour. In the daily setting, we investigate structured models with 19 outputs - the productions of the plant for the hours from 2:00 AM to 8:00 PM on a specified day. Furthermore, we consider scenarios with and without the latitude and the longitude of the plant taken as descriptive variables. The later will investigate whether the geographic coordinates play an important role for the prediction performance.

[5]http://forecast.io/

[6]http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php

Table 1: Performance results for one-day ahead PV power forecast for hourly (non-structured) and daily (structured output) settings. No spatial (Lat Lon) indicate results without (with) geographic coordinates of the plant.

	RMSE	MAE	Impr. [%]
No Spatial Hourly	0,120	0,079	17,410
No Spatial Daily	0,109	0,068	24,810
Lat Lon Hourly	0,120	0,078	17,443
Lat Lon Daily	0,111	0,069	23,915
Persistence model	0,146	0,085	

3.3.4 Results and discussion

The results for the investigated hourly and daily scenarios are reported in Table 1. We consider three indicators of the predictive performance, namely the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE), and the improvement over the persistence model (i.e., the model that forecasts the same production observed 24 hours before).

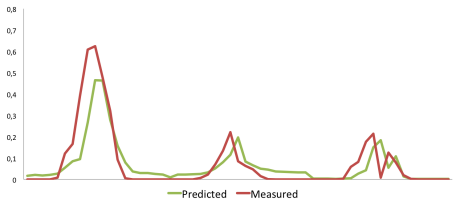
The results clearly show improvement of the predictive performance over the persistence model, with the structured-output prediction model clearly outperforming the non-structured one. From Table 1, we can also notice that geographic coordinates improve the prediction effectiveness, suggesting that data are subject to the spatial autocorrelation phenomena. [21].

The predictive performance of the model can also be graphically inspected from Figure 7, where the predicted vs. measured power production are presented for three consecutive typical cold (in January) and warm (in May) days. In both cases, we report predictions for partially cloudy days. The predictions are obtained using the best performing model, i.e. structured output considering the latitude and longitude as input attributes.

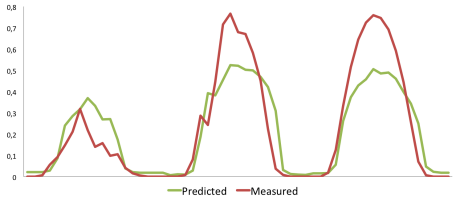
4. CONCLUSIONS AND FUTURE WORK

Big Data analysis is a challenging task as we need to take into account the velocity, variety and volume of information to be analyzed. In this respect, we have proposed a design option to implement a prototype for accurate prediction of renewable energy production plant output. In this paper, we have presented the project Vi-POC – a distributed system for storing, querying and analyzing data collected from renewable energy production plants. In particular, we have described its data model and its forecasting capabilities. As for this last aspects, we have empirically shown its predictive capabilities and compared cases with structured output prediction and non-structured output prediction. Results confirm that predictive capabilities are better in case of structured output prediction, probably because of the implicit consideration of the dependence of the predictions at consecutive hours.

As future work, we plan to further explore prediction techniques based on clustering, along with the integration of additional data sources in our system in order to achieve more accurate results. More in detail, we plan to test our system in different regions having different weather condition w.r.t. south of Italy in order to generalize our technique for a widespread commercial use.



(a) January 4^{th}, 5^{th}, 6^{th}.



(b) May 4^{th}, 5^{th}, 6^{th}.

Figure 7: Predictions (green) and measurements (red) of the productions for three consecutive days of a single plant. The three consecutive days are taken from January and May. Results are obtained with the daily (structured) setting. We recall that the time intervals considered are 2:00 AM - 8:00 PM. Results are obtained including also geographic coordinates as attributes.

5. REFERENCES

[1] Big data. *Nature*, September 2008.

[2] Drowning in numbers - digital data will flood the planet - and help us understand it better. *The Economist*, Nov 2011.

[3] D. Agrawal et al. Challenges and opportunities with big data. A community white paper developed by leading researchers across the United States. Mar 2012.

[4] Peder Bacher, Henrik Madsen, and Henrik Aalborg Nielsen. Online short-term solar power forecasting. *Solar Energy*, 83(10):1772 – 1783, 2009.

[5] Gökhan H. Bakır, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar, and S. V. N. Vishwanathan, editors. *Predicting structured data*. The MIT Press, 2007.

[6] T. G. Barbounis and J. B. Theocharis. Locally recurrent neural networks for wind speed prediction using spatial correlation. *Inf. Sci.*, 177(24):5775–5797, December 2007.

[7] R.J. Bessa, V. Miranda, and J. Gama. Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting. *Power Systems, IEEE Transactions on*, 24(4):1657–1666, 2009.

[8] S. Bofinger and G. Heilscher. Solar electricity forecast - approaches and first results. In *20th Europ. PV conf.*, 2006.

[9] M. Ceci, N. Cassavia, R. Corizzo, P. Dicosta, D. Malerba, G. Maria, E. Masciari, and C. Pastura. Innovative power operating center management exploiting big data techniques. In *18th International Database Engineering & Applications Symposium, IDEAS 2014, Porto, Portugal, July 7-9, 2014*, pages 326–329, 2014.

[10] Michelangelo Ceci, Nunziato Cassavia, Roberto Corizzo, Pietro Dicosta, Donato Malerba, Gaspare Maria, Elio Masciari, and Camillo Pastura. Big data techniques for renewable energy market. In Sergio Greco and Antonio Picariello, editors, *22nd Italian Symposium on Advanced Database Systems, SEBD 2014, Sorrento Coast, Italy, June 16-18, 2014.*, pages 369–377, 2014.

[11] Prithwish Chakraborty, Manish Marwah, Martin F. Arlitt, and Naren Ramakrishnan. Fine-grained photovoltaic output prediction using a bayesian ensemble. In *AAAI*, 2012.

[12] EPIA European Photovoltaic Industry Association. Global Market Outlook for Photovoltaics 2014-2018. http://www.epia.org/news/publications/global-market-outlook-for-photovoltaics-2014-2018, June 2014.

[13] George Kariniotakis. *Contribution to the development of an advanced control system for the optimal management of wind-diesel power systems*. PhD thesis, 1996.

[14] Jan Kleissl. *Solar Resource Assessment and Forecasting*. Elsevier, 2013.

[15] Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Džeroski. Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3):817–833, 2013.

[16] H. A. Nielsen, P. Pinson, L. E. Christiansen, T. S. Nielsen, H. Madsen, J. Badger, G. Giebel, and H. F. Ravn. Improvement and automation of tools for short term wind power forecasting. In *EWEC*, 2007.

[17] Debprakash Patnaik, Manish Marwah, Ratnesh K. Sharma, and Naren Ramakrishnan. Temporal data mining approaches for sustainable chiller management in data centers. *ACM Trans. Intell. Syst. Technol.*, 2(4):34:1–34:29, July 2011.

[18] Sophie Pelland, Jan Remund, Jan Kleissl, Takashi Oozeki, and Karel De Brabandere. Photovoltaic and solar forecasting. Technical report, IEA PVPS, 2013.

[19] Pierre Pinson, Henrik Aa. Nielsen, Henrik Madsen, and Torben S. Nielsen. Local linear regression with adaptive orthogonal fitting for the wind power application. *Statistics and Computing*, 18(1):59–71, March 2008.

[20] Navin Sharma, Pranshu Sharma, David E. Irwin, and Prashant J. Shenoy. Predicting solar generation from weather forecasts using machine learning. In *SmartGridComm*, pages 528–533. IEEE, 2011.

[21] Daniela Stojanova, Michelangelo Ceci, Annalisa Appice, Donato Malerba, and Saso Dzeroski. Dealing with spatial autocorrelation when learning predictive clustering trees. *Ecological Informatics*, 13:22–39, 2013.