J. Lopez, J. E. Rubio, and C. Alcaraz, "A Resilient Architecture for the Smart Grid", *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 3745-3753, 2018.
http://doi.org/10.1109/TII.2018.2826226
NICS Lab. Publications: https://www.nics.uma.es/publications

A Resilient Architecture for the Smart Grid

Juan E. Rubio, Javier Lopez, Cristina Alcaraz
Department of Computer Science, University of Malaga,
Campus de Teatinos s/n, 29071, Malaga, Spain
{rubio, alcaraz, jlm}@lcc.uma.es

Abstract

The Smart Grid offers many benefits due to the bidirectional communication between the users and the utility company, which makes it possible to perform a fine-grain consumption metering. This can be used for Demand Response purposes with the generation and delivery of electricity in real time. It is essential to rapidly anticipate high peaks of demand or potential attacks, so as to avoid power outages and denial of service, while effectively supplying consumption areas. In this paper, we propose a novel architecture where cloud computing resources are leveraged (and tested in practice) to enable, on the one hand, the consumption prediction through time series forecasting, as well as load balancing to uniformly distribute the demand over a set of available generators. On the other hand, it also allows the detection of connectivity losses and intrusions within the control network by using controllability concepts.
Keywords: Smart Grid, control systems, structural controllability, power dominance, prediction, load balancing, fault detection, resilience.

1 Introduction

The traditional architecture of the electricity grid has evolved in great measure since its original conception where the production and distribution of energy were supervised by a centralized system. With the introduction of Internet communication technologies in this scheme, there has been a shift towards a more interactive, interconnected and dynamic grid model of the 21st century, known as the Smart Grid. Its main benefit is the two-way flow of information, through which the user (i.e., by means of a smart meter installed in the household) and the utility company can communicate, making it possible to perform a fine-grain consumption metering, whose information is accessible to both of them [1]. This allows the user to participate in programs that aim to reduce electricity use when energy prices rise, and also allows him/her to sell the electricity generated at home (e.g., using solar panels). The utility company can also take advantage of this technology to improve Demand Response, by managing the generation and delivery of electricity in real time, so that grid operators can rapidly anticipate high peaks of demand and avoid power outages.

This metering model is put into practice through the Advanced Metering Infrastructure (AMI). This comprises all the elements that collect and transfer the consumption

data measured in the home domain through many aggregation points until it reaches the utility provider end, where the information is analyzed for billing and control purposes, by means of the so-called Meter Data Management Systems (MDMS).

This data acquisition process requires both industrial and information technology equipment. On the one hand, the industrial network is conformed by the SCADA (Supervisory Control And Data Acquisition) systems that are leveraged to remotely access the devices that sense the energy flow of many consumers in real time. These include, for example, the RTUs (Remote Terminal Units), and PLCs (Programmable Logic Controllers), that are present in the substations spread over the WAN (Wide Area Network or the Smart Grid). On the other hand, support for the MDMS procedures by interconnecting these industrial assets with external networks (e.g., Internet) and innovative technologies (e.g., cloud computing) to undergo further data analysis and support Demand Response.

This growing interconnection of SCADA systems (which traditionally work in isolation) has increased the number of cyber-security threats in this context [2], favoring the appearance of sophisticated attacks which aim to stealthily compromise nodes within the control network over a long period of time. The presence of these attacks can damage the infrastructure and jeopardize the availability of resources, which translates into the inability to hold the power supply and potential blackouts in the grid [3]. By the same token, safety measures must also be introduced to preserve the availability of the power supply against high peaks of demand (that may also be provoked on purpose), hence avoiding outages.

For the aforementioned reasons, we present the design and implementation of (1) a defense mechanism to detect topological changes in the industrial network arising as a consequence of these attacks, using collaborative decision algorithms and graph theory. In addition to ensuring security, here we also address the safety of the Smart Grid resources by implementing (2) a load balancing model that permits a successful energy supply for the entire grid taking into consideration the prediction of future consumption. Both safety and security measures are included in a novel architecture to be easily integrated in the current Smart Grid model.

The remainder of this paper is organized as follows: Section 2 presents the aforementioned architecture and all the networks and components needed for the safety security mechanisms. In Section 3 the load balancing and consumption prediction algorithm are defined. The respective fault detection and control protection mechanisms are explained in Section 4. The experimental tests with the security and safety techniques are discussed in Section 5. Finally, some conclusions are given in Section 6.

2 Architecture and initial assumptions

2.1 Five networks-based architecture, network of networks

The architecture of our approach is presented in this section and has two main purposes: (i) to predict high electricity peaks in comparison with the recent demand to uniformly distribute the energy supply to the consumption areas, and (ii) protect the control from external attacks. To achieve both goals, and therefore, the contributions of the work

presented here, an architecture of two main networks is modeled: an energy network (N^e) and a communication network (N^c). These two networks contain five independent but strongly interconnected subnetworks, which are shown in Fig. 1. Each subnetwork contains a set of Internet-enabled nodes (e.g., meter concentrators, gateways, RTUs, etc.) capable of interconnecting by itself with other subnetworks. As for the energy network, the following subnetworks have been defined:

$\mathbf{N}_1^e$ illustrates the customer's premises, subdivided into several power distribution areas or communities. In this case, each area characterizes a sub-part of a population, demanding energy according to its needs, requirements and life quality.

$\mathbf{N}_2^e$ represents the spinal column of the entire energy generation and distribution infrastructure, which remains in a fixed and static deployment and configuration state.

In practice, electricity generators in N_2^e are interconnected in the power grid with the consumption areas in N_1^e through rigid transmission and distribution lines. Besides these energy subnetworks, we also deal with communication subnetworks that firstly transfer the energy usage data from the consumers to the provider and secondly transmit the control commands from the utility to adjust the generators according to the demand. In this sense, we define:

$\mathbf{N}_1^c$ represents the set of smart meters that collect the measured energy usage data in the home domain.

$\mathbf{N}_2^c$ corresponds to the cloud computing-based communication system to centralize all the computation and the forecasting process in nodes with high capacity to estimate new and nearby states, such as servers or proxies. In this way, it is possible to decouple the control processes and the demand management from additional computational processes that are required for the prediction.

$\mathbf{N}_3^c$ embodies all the control and automation processes, required to protect the most critical underlying systems, such energy distribution and transmission substations. In this context, different cyber-physical elements are characterized such as acquisition and supervision elements working as driver nodes (e.g. RTUs, PLCs or gateways), and observation and control elements serving as sensory and reactive devices (e.g. sensors and actuators).

In real world, cloud resources belonging to N_2^c aggregate the information received from the users (via their smart meters embedded in N_1^c) and compute an estimation of future consumption. According to this forecast, the generators of the production system are programmed by means of the actuators placed in the N_3^c subnetwork, which finally provide the electricity supply back to the consumption areas.

However, the conceptual construction of each of these subnetworks further entails working with aspects associated with graph theory and other concepts, related to structural controllability [4] and dominance [5]. For example, components of the subnetwork $\mathbf{N}_1^e$ and $\mathbf{N}_2^c$ are modeled on the basis of a random pattern, where the greater part

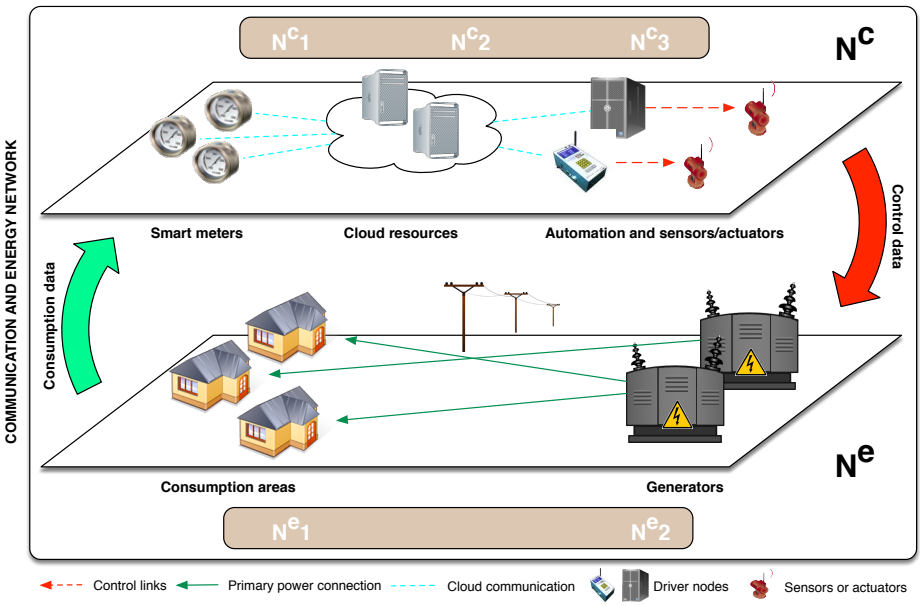


Figure 1. Five subnetworks-based architecture

of $\mathbf{N}_1^e$ is permanently linked to $\mathbf{N}_2^e$ elements (due to the fixed deployment of the energy distribution infrastructure) whereas a few nodes of $\mathbf{N}_2^c$ are permanently connected to elements of each area in $\mathbf{N}_1^c$ and driver nodes in $\mathbf{N}_3^c$. $\mathbf{N}_3^c$, to the contrary, follow specific network constructions centered on power-law distributions of type $y \propto x^{-\alpha}$. This constraint is due to the structural features of real control infrastructures, which are based on multiple interconnected substations with a few industrial nodes (e.g, RTUs, sensors, actuators). This conceptually follows a hierarchical network architecture based on nodes with high degree (i.e., the number of edges incident on the node) connected to nodes with lower degree; similar characteristics to the power-law distributions as stated in [6] and [7]. The authors in [7], additionally, justify why other models are not applicable for power grids, such as the small-world distributions. According to them, the conditions given by, for example, Watts and Strogatz [8] are not satisfied by Power Grid samples due to physical and economic issues.

$\mathbf{N}_2^e$, in turn, is based on specific grid distributions of type IEEE 118-bus or IEEE 300-bus as specified in [9], where we extract a subpart of these models to lead the practical case studies and the experimental results presented later.

2.2 Background and preliminary assumptions

To formalize the problem, we characterize two graphs, one related to N^e and another one to N^c. For N^e, let $\mathscr{G}^e(V^e, E^e)$ be a directed bipartite graph, such that V^e is the union of the nodes in N_1^e and N_2^e, and the set of n customer areas in $\mathbf{N}_1^e$ are connected to m grid generators of $\mathbf{N}_2^e$ through grid connections in E^e. For the resilience and load balancing,

we assume that each area is associated with δ generators, such that $\delta \geq 2$. Within N^c, we consider N_3^c to analyze the adversarial influence on the operational processes. Let $\mathscr{G}_3^c(V_3^c, E_3^c)$ be a directed graph, containing the minimum set of driver nodes (referred to here as D_N) capable of injecting control signals into the rest of the elements in V_3^c, also denoted here as the set of observed nodes (the set O), such that $\mathbf{D_N}$ and $\mathbf{O} \subseteq V_3^c$, all of them connected through communication links in E_3^c.

Under these conditions, several threat assumptions should be considered during the modeling and simulation of study cases. Firstly, attacks to be analyzed in this paper are concentrated in $\mathscr{G}^e(V^e, E^e)$ and $\mathscr{G}_3^c(V_3^c, E_3^c)$, where the adversarial model follows a weak approach, in which it is also assumed the attacker has high mobility in both subnetworks (to perform attacks against the power supply and the control network, respectively). The threats can be multiple and varied, where the adversary may target β nodes or edges, and depending on the network, the interests may be very different. An attack in $\mathbf{N}_1^e$ may, for example, focus on producing concurrently anomalous deviations in the real demand and potentially overloading the power grid, misusing the energy during peak times. Contrarily, an attack in $\mathscr{G}_3^c(V_3^c, E_3^c)$ may mean the constant removing of a few random communication links in specific nodes, simulating a denial of service. In this case, the attackers' goal would be to alter the structural controllability to strategically unprotect the control itself and the functionality of $\mathbf{N}_2^e$.

Given this and the interconnected nature of the electrical systems and the technologies for the control and automation in real time, two independent, but narrowly related, approaches are presented in this paper. These are intended to protect the following: (1) the processes of production and distribution of energy and the (2) the control processes in response to unexpected changes which may also have a (mild, severe or irreparable) rebound effect on the dependent subnetworks (e.g., outages in $\mathbf{N}_1^e$, overloading in $\mathbf{N}_2^e$) [10].

3 Consumption prediction and load balancing

Taking into account the aforementioned architecture, the first task required for the cloud infrastructure in $\mathbf{N}_2^c$ is the ability to provide load balancing support to the generators according to the demand, for an effective electricity supply. Specifically, the main concern is the anticipation of upcoming peaks of demand, which could also be caused on purpose to cause blackouts in certain areas of the grid. By possessing this knowledge in advance it is possible to rapidly distribute the existing demand, at a given moment, among all the generators available in the grid (located in $\mathbf{N}_2^e$), so that the affected consumption areas can keep receiving the requested energy and the continuity of the service is ensured.

In order to test the proposed load balancing algorithm in practical terms, it is desirable to firstly devise a way to simulate the generation of consumption data in real time. We intend to imitate the demand response under normal conditions and in the presence of anomalies (by introducing eventual outages in the data), with the aim of performing predictions that serve as input for the load distribution. This way, we can check the effectiveness of the algorithm against peaks and adverse conditions in a timely manner. For the sake of veracity when designing the generation of the bulk data that is used to

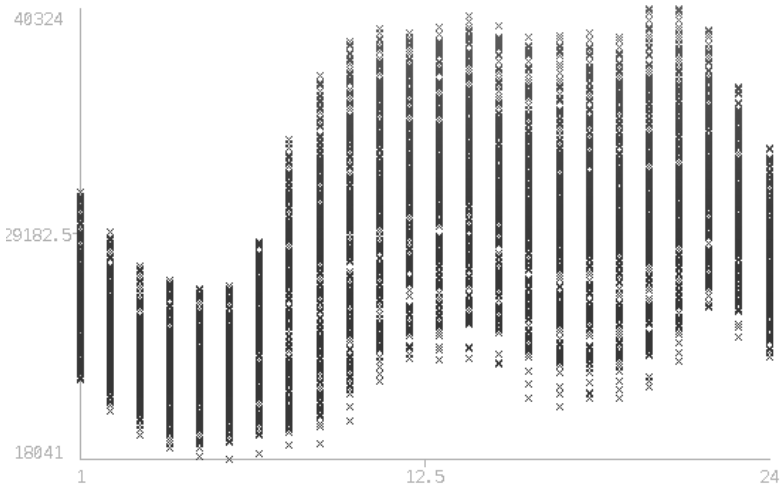


Figure 2. Hourly load values of Spain in 2015 [11]

check the accuracy of predictions, we have based our work on the datasets provided by the European Network of Transmission System Operators for Electricity (ENTSO-E) [11]. This organization represents 43 electricity transmission system operators (TSOs) from 36 countries across Europe, and provides hourly load values of all those countries at monthly intervals. Specifically, we have designed a custom dataset comprising all the hourly consumption values (in MW) of Spain from 2015, the last year for which data is available. If we show all these samples in a window of 24 hours, we obtain the graph in Fig. 2.

As we can see in the figure, all the daily consumption values over the 365 days are plotted, resulting in a curve where most of the electricity demand is concentrated in the evening and decreases during the night. Based on this information, we use the actual data to define a mathematical function (henceforth the F function) that automatically generates consumption values indefinitely. For that, we perform a non-linear regression using the Gauss-Newton algorithm that finds a function of the type $y = Asin(Bx + C) + D$ that conforms to a set of data points (x_i, y_i). For the sake of clarity and the purpose of showing the efficiency of the prediction and load balancing method, we assume that the resulting consumption value only depends on the day of the week and the time of the day at which we want to predict the usage value (as independent variables of the function). Additionally, the month could be considered to analyze the influence of seasons. However, to provide a degree of randomness in the data and avoid returning the same value for a given set of input arguments (i.e., day of the week and hour), we consider adding certain deviations, whose value is arbitrarily chosen from a uniform distribution $U(-\lambda, \lambda)$, where λ represents the maximum divergence value. In addition, we have included the possibility of experiencing a peak of consumption (i.e., a considerable increase in certain values) under a probability γ. We must also

mention that the original electricity values from the datasets have been divided by 100 to represent the conceptual consumption of a single area or province in Spain. By doing this, the demand of multiple consumption areas over the grid is simulated, which is accomplished through the execution of the aforementioned F function, in parallel, for several instances. Apart from this function to simulate the consumption, we must find a way to predict future values based on previous behavior. Altogether, this information will serve as input for the load balancing algorithm executed in the $\mathbf{N}_2^c$ systems, that finally is responsible for the prevision of the energy supply for all areas within the grid.

In this paper, the prediction of the energy usage between neighborhoods is based on time series forecasting. Contrary to traditional machine learning methods, which also work with multiple datasets but treat all the observations equally, time series adds an explicit order dependence to all of them: the *time dimension*. This gives higher importance to the last observations rather than all data available, which is valuable for prediction. In addition, the analysis of time series can also determine seasonal patterns, trends or the relationship with external factors. In our case, the aim is to forecast future values of a time series, that is, the one described by the consumption curve. Specifically, we use the statistical model ARIMA, which stands for *AutoRegressive Integrated Moving Average* and counts on three different components, expressed as *ARIMA(p,d,q)*:

- **Autoregression (AR):** use of a dependent relationship between an observation and a number of lagged observations, represented by the p parameter.
- **Integrated (I):** in order to make the time series stationary, it differentiates between raw observations (e.g. subtracting an observation from an observation at the previous time step). The number of times that the observations are differentiated is represented by d.
- **Moving Average (MA):** use of a dependency between an observation and a residual error from a moving average model. The size of the moving average window is represented by q.

These parameters (p,d,q) are characterized according to the general ARIMA model:

$$Y_t = -(\Delta^d Y_t - Y_t) + \phi_0 + \Sigma_{i=1}^{p} \phi_i \Delta^d Y_{t-i} - \Sigma_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (1)$$

where $\phi_1, ..., \phi_p$ are the parameters of the autoregressive part of the model and $\theta_1, ..., \theta_q$ belong to the MA, and the rest of parameters are part of the integration filter. Lastly, ε adds an error margin. The parametrization and accuracy of the ARIMA model for our purposes are discussed later, specifically in Section 5. The result of applying this model provides a set of future energy readings, taking into account the last consumption reports. As explained, once we have this information, the last step for load balancing consists in uniformly distributing the available electricity supplied by the generation resources in the grid among all the consumption areas at a given moment (which is represented with the graph $\mathscr{G}^e(V^e, E^e)$), taking into account the forecasted value of the amount of requested energy by each of these areas.

In more detail, for the design of the load balancing algorithm, we have the following constraints: let us assume a set of generators G of $\mathbf{N}_2^e$ that supply electricity for a set of

areas A. Each generator i has a maximum load denoted by g_i, and each area j demands a_j units of energy, having $1 \leq i \leq |\mathrm{G}|$ and $1 \leq j \leq |\mathrm{A}|$. As initial conditions, we accept that:

- **C1:** there does not exist any area j whose a_j is higher than any g_i, for all $i \in \mathrm{G}$. This ensures that every area can be supplied by at least one generator.
- **C2:** the sum of electricity requested by all the areas does not exceed the sum of electricity supplied by the generators; formally, $\sum_{j=1}^{|\mathrm{A}|} a_j \leq \sum_{i=1}^{|\mathrm{G}|} g_i$. This ensures that all areas can be provided with the requested energy.

Therefore, what we want to find is a relationship $R \subseteq$ A x G between areas and generators, such that each area is assigned with a generator and the sum of electricity requested by the areas associated with a generator, does not exceed its capacity. This can be modeled as a search algorithm, since we explore a set of candidate solutions in the form of a tree, beginning with the initial one (an area is assigned to a generator) and gradually adding associations in the search for a valid solution, which is when all areas are assigned with a generator and **C1** and **C2** are consistently satisfied.

More specifically, we have designed a novel algorithm that makes use of backtracking, which is widely used for constraint satisfaction problems [12]. It incrementally builds candidates in the solution, and discards each partial candidate as soon as it determines that it does not comply with the proposed conditions, which makes it impossible for the candidate to be completed as a valid solution. The resultant technique is explained in Algorithm 1.

Algorithm 1 Load Balancing (A, G)

```
output (R = {(a_j, g_i)} where 1 ≤ i ≤ |G| and 1 ≤ j ≤ |A|)
R ← {}
R ← SolveLoadBalancing(A,G,R)

function SolveLoadBalancing(A,G,R)
    if |R| = |A| then
        Found ← True return R[a]
    else
        Found ← False
        j ← 1
        while not Found and area a_j not assigned and j ≤ |A| do
            i ← 1
            while not Found and i ≤ |G| do
                if energy assigned to generator i + a_j ≤ g_i then
                    R' ← R ⋃ (a_j, g_i)
                    SolveLoadBalancing(A,G,R')
                end if
                i ← i + 1
            end while
            j ← j + 1
        end while
    end if
end function
```

[a] A solution where an assignation has been found

In this case, a partial candidate represents a relationship R where not all areas are assigned to a generator. As described, the algorithm begins by assigning one random area to one random generator, and keeps iterating in the search for a valid solution,

assigning new areas to generators if their capacity still allows it and recursively calling the function (which is modeled by the inner loop of the algorithm). Otherwise, the partial candidate is discarded and another area is assigned in the first loop of the algorithm. Thus, the *Found* variable finally indicates whether or not there is a feasible relationship between areas and generators that successfully distributes the energy, complying with **C1** and **C2** conditions.

4 Fault detection and control protection

Together with the prediction and load balancing algorithm that ensures the safety of the power supply infrastructure, the other task required for the resilient architecture consists in the security of the control elements belonging to $\mathbf{N}_3^c$, represented with the graph $\mathscr{G}_3^c(V_3^c, E_3^c)$. We aim to secure the structural controllability domain by developing a distributed decision algorithm that enables us to detect subtle changes in the underlying network, that may be the result of an stealth attack. If we assume a set of finite agents uniformly distributed over the industrial network (named driver nodes in Section 2), it is possible to execute cooperative algorithms that allow them to accurately identify what parts of the topology have suffered changes, which is determined by exchanging information about their surroundings with each other. This information can be used to deploy effective recovery techniques to guarantee the continuity of the service.

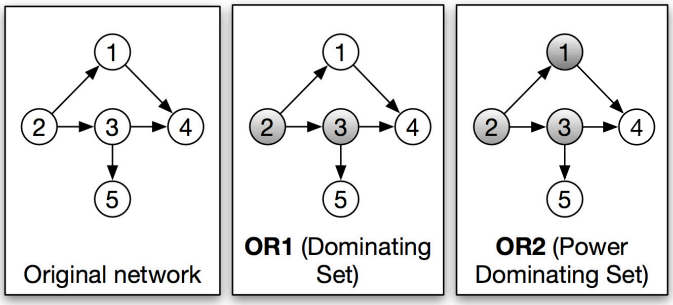


Figure 3. Observation rules for the election of the driver nodes

The aforementioned driver nodes ($\mathbf{D_N} \in V_3^c$) are selected according to the following two rules [13], which are represented in Figure 3. Note that although these two rules are treated as observation rules, we apply them here to the dual problem related to controllability:

OR1 *A driver node, n_d in $\mathbf{D_N}$, observes itself and all its neighbors*, which conforms the DOMINATING SET of nodes. This implies that every node not in $\mathbf{D_N}$ is adjacent to at least one member of $\mathbf{D_N}$.

OR2 *If a driver node, n_d in $\mathbf{D_N}$, of degree $d \geq 2$, is adjacent to $d-1$ driver nodes, then the remaining node v in $\mathbf{O}$ becomes driver node as well, such that* $\mathbf{D_N} \longleftarrow$

$\mathbf{D_N} \cup \{v\}$ *and* $\mathbf{O} \longleftarrow \mathbf{O} \setminus \{v\}$. Thus, the set of driver nodes provided by **OR2** includes the fulfilment of **OR1**, which conforms the POWER DOMINATING SET. It means that every edge in E_3^c is adjacent to at least one member of $\mathbf{D_N}$.

Specifically, we have considered that the set of driver nodes fulfills **OR1** and **OR2** conditions. The detection algorithm between these agents is a light modification of the opinion dynamics algorithm described in [14], processed by each driver node n_d in $\mathscr{G}_3^c(V_3^c, E_3^c)$ in discrete time. This approach creates a fragmentation of the affected zones within the network once the agents share information about their surrounding topology.

For the computation of opinion dynamics it is necessary to define a matrix W of size n x n (with $n = |V_3^c|$) holding the weights that represent the confidence between the agents' opinions. Each agent assigns a weight to the rest of agents in its surroundings (particularly those nodes sharing a communication link) based on the closeness between their opinions. Altogether, the vector $x_{nd}(t)$ holds the opinion of each driver node such that it is updated by $x_{nd}(t+1) = W(t, x_{nd}(t))x_{nd}(t)$, where t refers to another iteration of the algorithm. The logic of this equation is equal to $x_{nd}(t+1) = w_{i1}x_{nd_1}(t) + w_{i2}x_{nd_2}(t) + \cdots + w_{in}x_{nd_n}(t)$ such that $wij = 1$. Each opinion is originally calculated according to the new state of the network with respect to the original topology, which is computed with the difference in the node betweeness centrality:

$$BC_(v) = \sum_{s,t \in V} \frac{\delta(s,t \mid v)}{\delta(s,t)} \tag{2}$$

where $\delta(s,t)$ represents the shortest (s,t)-paths (with $s \neq v \neq t$) and $\delta(s,t \mid e)$ the paths passing through the node v. This way of representing structural behaviors includes the way to characterize the principal control loads in $\mathbf{N}_3^c$ [15], assuming that the main network control dynamics flow through the shortest paths. Therefore, any topological variation impacts on *BC* and subsequently on the new upgrading of $x_{nd}(t+1)$ in time $t+1$. Once we execute the opinion dynamics with t tending to infinity (i.e., a high number of steps), it is possible to visualize the consensus between clusters of agents about topological changes on different parts of the network. Opinions $\simeq 1$ mark topological changes within $\mathbf{N}_3^c$ that are generally located in the surroundings of those local driver nodes that detect the deviation. This also means that a persistent, yet subtle change, over time, with values close to or exceeding 0.5 can mean the approximation of a structural change.

In order to prove the effectiveness when detecting topological changes, we must simulate the action of a stealth attack, taking its nature into consideration. Specifically, these mutations appear as a consequence of the lateral movements taken to find new victim nodes and hence gain influence within the network. These attacks have to be planned strategically instead of leading arbitrary attacks, where the target must be focused on the control and its dynamics as stated in [16]. Based on the general attack behavior described in that paper, we have defined three different attack models:

STG1 : the attacker focuses on an arbitrarily chosen node within the network and performs a change on any of its adjacent edges, to subsequently move to a neighbor

node in a random way.

STG2 : the threat is concentrated on those driver hubs with the highest degree d^+ and d^-, where the attack aims to randomly remove a few edges.

STG3 : the adversary is able to attack the node with the highest influence over the control by simply observing the traffic and its bandwidth. Through graph theory, this representation is possible through the highest edge betweeness centrality of the neighborhood as specified in [16].

Taking into account these three threats, Algorithm 2 outlines the life cycle described by a stealth intrusion like this. It takes the original network described with $\mathscr{G}_3^c(V_3^c, E_3^c)$ and performs a succession of individual attacks against the edges E_3^c (i.e., either the addition or removal of incoming or outcoming edges), resulting in the modified network represented with $\mathscr{G}_3'(V_3^c, E_3')$. After each edge modification, the attacker propagates to an adjacent node in accordance with one of the strategies presented before. At this point, the decision dynamics algorithm can be executed to detect the portions of the network that are affected by the attack.

Algorithm 2 Stealth attacks life cycle

output: $\mathscr{G}_3'$ *representing the resulting matrix M*
local: $\mathscr{G}_3^c(V_3^c, E_3^c), numOfAttacks, STG_x$
$attackedNode \leftarrow random\ v_i \in V_3^c; \mathscr{G}_3' \leftarrow \mathscr{G}_3^c$

for i:=1 **to** *numOfAttacks* **step** 1 **do**
  $attack \leftarrow randomAttack\ over\ attackedNode$ (*edge addition or removal*)
  update $\mathscr{G}_3'$ *based on attack*
  if $STG_x = 1$ **then**
    $attackedNode \leftarrow random\ v_i \in V_3^c$
  else if $STG_x = 2$ **then**
    $attackedNode \leftarrow$ NEIGHBOURWITHHIGHESTDEGREE$(M, attackedNode)$
  else if $STG_x = 3$ **then**
    $attackedNode \leftarrow$ NEIGHBOURWITHHIGHESTBETWEENESS$(M, attackedNode)$
  end if
end for

5 Experimental results and discussions

After successfully designing mechanisms to firstly ensure the safety of the grid and also the security of the control elements involved, our aim is to test these services in practice.

To start with, we have to implement the F function in charge of generating the consumption plot that in conjunction with the information provided by the prediction process, serves as input to the load balancing algorithm. As described in Section 3, we have leveraged the annual consumption dataset in Spain as of 2015 to adjust a nonlinear correlation of the data to create the F function. This function simulates the consumption for a specified *hour* and a $dayOfTheWeek$, over which we have also added some extent of randomness λ (here we assume $\lambda = 15$) and a potential *peak* (a value of 50 has been considered) in the energy usage under a given probability γ. The result is the following expression:

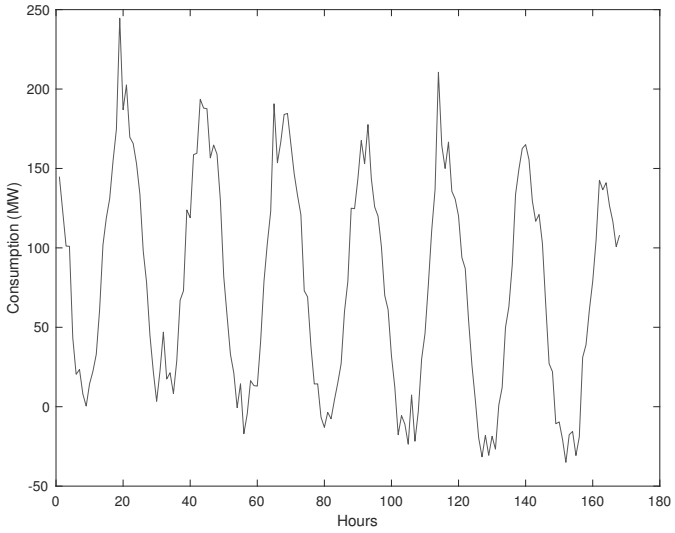


Figure 4. Weekly consumption generated by F function

$$F = 100 * cos(hour/3.82 + \pi/3) + 100 \\ -8 * dayOfTheWeek + \lambda + peak \tag{3}$$

Where the output value is expressed in MW, and holds the value of consumption for certain regions within the grid. For instance, Fig. 4 shows the result of executing the F function for an entire week (i.e., showing the evolution over its 168 hours), with a peak probability of 5%. Taking a close look, we can rapidly see the two peaks produced on Monday and Friday at night. It is also clear that the overall progression evolves towards a lower consumption as the weekend approaches.

Once we have modeled the F function and we are able to successively generate consumption values over a time period, we move on to parametrize the ARIMA statistical model so as to treat the consumption output as a time series and perform the forecasting. In order to find the optimal value for the p, d, and q parameters, it is necessary to follow a formal methodology that estimates each one by examining the AR or MA behavior of the series and testing with initial values to subsequently analyze how the model fits the original data [17]. For this purpose, the Simple and Partial Autocorrelation functions (AFC and PACF, respectively) are used. Once the appropriateness of the model has been compared, its residual errors are checked with the Akaike Information Critera (AIC). For our particular case of forecasting the consumption time series, we have automated this process through the R *forecast* package [18], which enables the estimation of its coefficients and also gives a ratio of likelihood. For example, if we gather the consumption values of ten days, it determines that the *ARIMA(3,0,1)* model

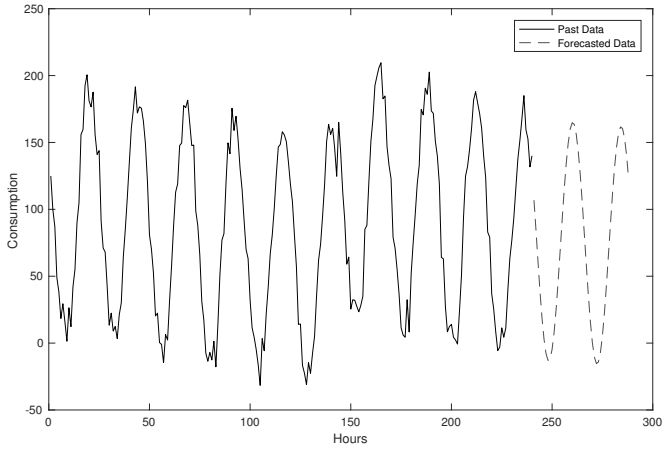
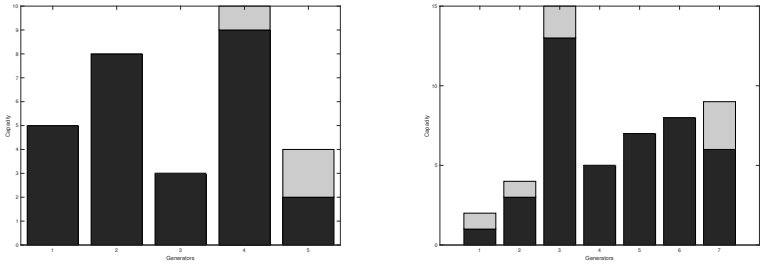


Figure 5. Forecast after 10 days using ARIMA



(a) IEEE-14: $\delta = 5, \theta = 11$ (b) IEEE-57: $\delta = 7, \theta = 42$

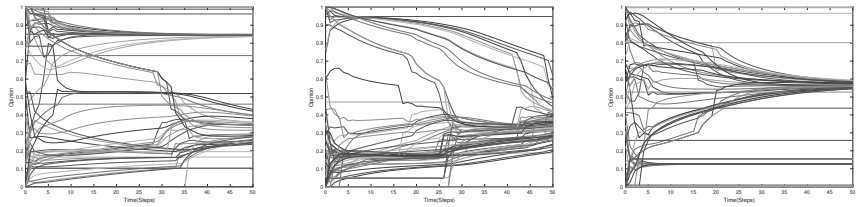
Figure 6. Load balancing for the two proposed systems

is suitable to fit the information, the value of which is computed as follows (taking into account Eq.1):

$$Y_t = a_1 Y_{t-1} + a_2 Y_{t-2} + a_3 Y_{t-3} + b_1 \varepsilon_{t-1} + \varepsilon_t \tag{4}$$

After defining the model, it is possible to perform the prediction of upcoming days. Specifically, Fig. 5 represents the forecast of two more days after a given period of time, which shows the accuracy of the ARIMA when predicting the consumption curve, that follows the expected progression.

Once we have the information about the future status of the grid at our disposal, we are in a position to execute the load balancing algorithm that uniformly distributes the electricity demand among all the generators available. Specifically, the energy usage prediction for all the individual areas spread over the Smart Grid provide sufficient input to the utility to carry out Demand Response. Recalling the concepts of the proposed



(a) STG1 (b) STG2 (c) STG3

Figure 7. Opinion dynamics after 50 attacks

architecture previously introduced in Section 2, $\mathbf{N}_2^e$ represents the energy distribution infrastructure, composed by the generators and substations that supply the electricity to the consumption points (e.g., neighborhoods and electric vehicle charge points). These assets are interconnected following the network described by the graph $\mathscr{G}^e(V^e, E^e)$, where we assume there are θ areas demanding energy to δ generators.

In the interest of veracity and taking into account that the aforementioned network remains rigid in its topology and configuration state, we have considered the IEEE-14 and IEEE-57 bus systems to carry out simulations based on a real-grid test case [19]. Both of them consist of a simple approximation of the American Electric Power system as of the early 1960s. The first system has 11 loads (assumed to be the areas of consumption for our purposes) connected to 5 generators, whereas the second model has 7 generators and 42 loads. A test case has been defined for each one, with as many areas (θ) and generators (δ) as each system respectively defines. We have supposed that every generator i in the G set has a maximum load g_i that is randomly selected in a defined interval, and each area j demands a_j units of energy whose value is, at most, the maximum value of capacity for a single generator. Taking these parameters into consideration, Fig. 6 shows the simulation of the load balancing algorithm for the IEEE-14 and IEEE-57 systems, where we can see how the consumption areas are accommodated to the available generators. To simplify, we have considered a maximum of capacity per generator of 10MW and 15MW, respectively.

So far, we have put into practice the mechanism that preserves the availability of the AMI infrastructure in its safety dimension. As for security, we now show the effectiveness of the intrusion detection technique based on opinion dynamics. For this, we have randomly created a network of a power-law distribution composed by 100 nodes, where we have conducted a set of 50 topological attacks as described in Algorithm 2. If we run the opinion dynamics algorithm over the set of 70 agents (which are driver nodes of $\mathscr{G}_3^c$), we can check how the opinions evolve to reach a consensus and create different clusters within the network. More specifically, Fig. 7 shows how the total number of agents of the network are divided into substantial sets depending on the degree of change, for the three attack strategies that we define in Section 4.

In these plots, each line represents the change in the opinion of the corresponding agent when the algorithm is executed over 50 steps ($t = 50$ in the opinion dynamics algorithm). Altogether, the presence of big clusters of opinions mean a confident consensus of agents about a change experienced in a particular area, whose level of criticality

is higher as it approaches 1. This is particularly evident in the **STG3** test case, where most of the agents agree on a topological attack in a specific part of the network, with approximately 60% of change. This behavior occurs due to the attack model chosen: the attack in **STG3** always propagates to nodes with higher influence on the control (i.e., a higher betweeness centrality, this is, the driver nodes), which makes it easier for the agents to locate the subtle changes (and is also the most realistic pattern, since the attacker commonly aims to gain the control of the network). This result is somewhat similar to **STG2**, because it is expected that those nodes with a higher degree are precisely the ones that have greater hierarchy over the network. However, since **STG1** focuses on propagating the attack in a random way, it is harder for the agents to reach a consensus on the portions of the network that are affected, resulting in a fragmentation of multiple opinions. On the whole, this constitutes a valuable insight into deploying accurate response techniques to overcome the effects of one of these threats.

6 Conclusions

The Smart Grid is a innovative technology that brings many benefits to both operators and users, although it does have some drawbacks when it comes to security and safety, which can slow down its adoption in practice. Like other critical infrastructures, the evolution of its industrial equipment towards a highly connected and distributed model imposes several issues related to guaranteeing the availability of resources.

In this paper, we have proposed a novel architecture divided into five subnetworks that firstly permits the integration of a cloud infrastructure in charge of performing predictive analytics to comply with Demand Response, implementing a load balancing algorithm. In addition, we have leveraged a cooperative algorithm to allow operators to detect the presence of subtle attacks on the industrial network, which means a first line of defense. Experimental tests have been carried out to demonstrate the feasibility of our approach. Future work will involve the creation of advanced response techniques to allow the continuity of the network in the presence of attacks in the different, sensitive sections of the Smart Grid. In addition to this, we aim to compare our opinion dynamics-based approach with other distributed models (e.g., distributed consensus) in order to find more precise solutions in the detection processes.

7 Acknowledgements

This work was supported by the research project SADCIP (RTC-2016-4847-8), financed by the Spanish Ministry of Economy and Competitiveness. Likewise, the work of the second author has been partially financed by the Spanish Ministry of Education under the FPU program (FPU15/03213).

References

[1] R. R. Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on advanced metering infrastructure," *International Journal of Electrical Power & En-*

ergy Systems, vol. 63, pp. 473–484, 2014.

[2] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[3] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344–1371, 2013.

[4] C.-T. Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 201–208, 1974.

[5] T. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning, "Domination in graphs applied to electric power networks," *SIAM Journal on Discrete Mathematics*, vol. 15, no. 4, pp. 519–529, 2002.

[6] R. Albert, I. Albert, and G. L. Nakarado, "Structural vulnerability of the north american power grid," *Physical review E*, vol. 69, no. 2, p. 025103, 2004.

[7] G. A. Pagani and M. Aiello, "The power grid as a complex network: a survey," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 11, pp. 2688–2700, 2013.

[8] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, p. 440, 1998.

[9] P. Ranganathan and E. N. Kendall, *A Distributed Linear Programming Model in a Smart Grid*, ser. Power Electronics and Power Systems. Springer, 2017.

[10] L. Martins, R. Girao-Silva, L. Jorge, A. Gomes, F. Musumeci, and J. Rak, "Interdependence between power grids and communication networks: A resilience perspective," in *DRCN 2017-Design of Reliable Communication Networks; 13th International Conference; Proceedings of*. VDE, 2017, pp. 1–9.

[11] "Consumption datasets," https://www.entsoe.eu/data/data-portal/consumption/Pages/default.aspx, last retrieved in August 2017.

[12] V. Kumar, "Algorithms for constraint-satisfaction problems: A survey," *AI Mag.*, vol. 13, no. 1, pp. 32–44, Apr. 1992.

[13] J. Kneis, D. Mölle, S. R., and P. Rossmanith, "Parameterized power domination complexity," *Information Processing Letters*, vol. 98, no. 4, pp. 145–149, 2006.

[14] R. Hegselmann and U. Krause, "Opinion dynamics and bounded confidence, models, analysis and simulation," *Journal of Artificial Societies and Social Simulation*, vol. 5, no. 3, p. 2, 2002.

[15] C. Alcaraz and J. Lopez, "Safeguarding structural controllability in cyber-physical control systems," in *The 21st European Symposium on Research in Computer Security (ESORICS 2016)*, vol. 9879, Springer. Crete, Greece: Springer, 2016, pp. 471–489.

[16] J. E. Rubio, C. Alcaraz, and J. Lopez, "Preventing advanced persistent threats in complex control networks," in *European Symposium on Research in Computer Security*, vol. 10493, 22nd European Symposium on Research in Computer Security (ESORICS 2017). 22nd European Symposium on Research in Computer Security (ESORICS 2017), 09/2017 2017, pp. 402–418.

[17] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[18] R. Hyndman, "Forecast: Forecasting functions for time series and linear models in R," https://cran.r-project.org/web/packages/forecast/index.html, last retrieved in October 2017.

[19] W. Stevenson, *Elements of power system analysis*, ser. McGraw-Hill series in electrical engineering: Power and energy. McGraw-Hill, 1982.