Final report

1.1 Project details

Project title	**Cyber-phySicAl security for Low-VoltAGE grids (SALVAGE)**
Project identification (program abbrev. and file)	**ForskEL 12254**
Name of the programme which has funded the project	**ERA-Net**
Project managing company/institution (name and address)	**Danmarks Tekniske Universitet (DTU)** **Anker Egelunds Vej 1, Bygning 101A** **2800 Kgs. Lyngby**
Project partners	**Kungliga Tekniska Högskolan (KTH), Stockholm/SE** **Politechnika Wrocławska (PWR), Wrocław/PL**
CVR (central business register)	**30 06 09 46**
Date for submission	**16/04/2018**

1.2 Short description of project objective and results

1.2.1 English version

The ICT infrastructure in power grids is increasingly considered to be part of the "system to be operated", rather than just "something that has to work". However, tool support for integrated operation is still underdeveloped.

The SALVAGE project has been working to address the problem of cybersecurity in smart distribution grids by developing and/or improving sets of tools for different types of analyses: DER intrusion detection, cybersecurity vulnerability assessment and power system impact. Furthermore, it investigated how to combine the assessment results from three different domains into an integrated assessment.

The project has developed and validated methods for model-based intrusion detection for power system components through real-time analysis of component behavior. Furthermore, it has extended the CySeMoL framework for cyber vulnerability assessments. Finally, it has developed a proof-of-concept of an integrated assessment method combining power system impact, intrusion detection and cyber vulnerability information.

1.2.2 Danish version

Elnettets IKT infrastruktur bliver i stigende grad set som en del af ”systemet i drift” fremfor at være ”noget som bare skal virke”. Men der mangler effektive værktøjer til integreret drift.

SALVAGE-projektet har arbejdet på problemet med cybersikkerhed i fremtidens intelligente eldistributionsnet ved at udvikle og/eller forbedre værktøjer til forskellige former af analyse: Intrusion Detection for decentrale enheder, sårbarhedsanalyse mht. cybersikkerhed og påvirkning af det fysiske elsystem. Derudover har projektet undersøgt hvordan analysedata fra tre forskellige domæner kan indgå i en samlet vurdering.

SALVAGE har udviklet og afprøvet metoder til modelbaseret Intrusion Detection for enheder tilkoblet elnettet gennem realtidsanalyse af enhedernes opførsel. Derudover har projektet udvidet sårbarhedsanalyseværktøjet CySeMoL og har udviklet en metode til integreret vurdering af cybersikkerhed ved at kombinere sårbarhedsanalyse, Intrusion Detection og systempåvirkning.

1.3 Executive summary

The operation and management of electric power systems depends on the integrity of computerized industrial control systems. In the future smart grid, the number of assets with embedded intelligence and communication links is expected to grow strongly together with the degree of their interconnectedness. Inevitably, this development will create many additional opportunities and new vectors for cyberattacks. The biggest changes are expected to happen in the distribution grid due to an increasing level of distribution grid automation as well as the proliferation of controllable, customer owned energy resources such as electrical vehicles or distributed generation.

There is a lack of tool support for managing cyber security in a coherent fashion in the context of utility specific requirements. An often used approach to security is simply to apply standalone tools such as vulnerability scanners and to perform penetration tests. The problem with this approach is also that it is heavily dependent on the competence of the auditor and it is difficult to make complete audits in process control systems.

The purpose of the SALVAGE project was to develop better support for managing and designing a secure future smart grid. This approach includes cyber security technologies dedicated to power grid operation as well as support for the migration to the future smart grid solutions, including the legacy of ICT that necessarily will be part of it, with a special focus on smart grid with many small distributed energy resources, in particular MV and LV substation automation systems and LV distribution systems.

SALVAGE has achieved results in three main areas:

- Introduction of the novel idea of an intrusion detection system based on power system component models and comparing observed and expected behaviour of those components. Several detection algorithms were developed and tested in a laboratory environment.
- Refininement and extension of the Cyber Security Modeling Language (CySeMoL), a framework for making predictions of how vulnerable a certain ICT architecture is towards cyber attacks by using attack graphs.
- Development of a proof-of-concept of a novel type of framework that performs an integrated assessment of the state of the power grid, using aspects from multiple domains (power system impact, intrusion detection, cyber vulnerability) and multiple sources of input (power system measurements, distributed energy resources (DERs), IT and OT systems) in orderto provide a joint prioritization of possible threat/impact-scenarios, taking into account uncertainty of input information.

1.4 Project objectives

1.4.1 Background

Already today, the operation and management of electric power systems depends on the integrity of computerized industrial control systems. In the future smart grid, the number of assets with embedded intelligence and communication links is expected to grow strongly together with the degree of their interconnectedness. Inevitably, this development will create many additional opportunities and new vectors for cyberattacks.

Keeping these systems secure and resilient to external attacks is vital for the reliable delivery of power which society depends upon.

In the future smart grid the traditional centralized production will be replaced with many distributed DERs placed near the location of power consumption, in the distribution grid. Substations and assets supplied or delivering power to it, are considered vulnerable in the cyber-physical security context. Control concepts which require tighter interaction between utilities and third parties, such as Demand Response, Virtual Power Plants and remote control of Distributed Energy Resources (DERs) open new possibilities for cyberattacks. Fully automated substations can be attacked directly or remotely. Slow data poll cycles, or the unavailabiltiy of real-time data available from substations or DERs complicates the detection of malicious actions. This creates a need for distribution grid operators to address and counteract these threats.

1.4.2 Problem statement

Unfortunately there is a lack of good support for managing cybersecurity in the context of utility specific requirements. Perhaps the most clear support are a number of guidelines and standards. These are good material outlining best practices for increasing cyber security, but they do not help utilities to prioritize different cyber security controls and mechanisms. A more practice-oriented approach to security is simply to apply hands on tools such as vulnerability scanners and to perform penetration tests. The problem with this approach is also that it is heavily dependent on the competence of the auditor and it is difficult to make complete audits in process control systems.

Moreover, there is not only a lack of planning and decision making support for utilities and smart grids, also secure technology tailored for the domain is missing. Standard ICT security mechanisms are certainly a cornerstone for the future smart grid. However, it is not enough also domain specific solutions will be needed. New solutions combining cyber security measures and distribution power grid stability assessment need to be researched and developed to evaluate the cyber-physical security of the low voltage power grid infrastructure.

1.4.3 Project objective

The objective of the project was the development of instruments for designing and operating a secure future smart grid. The project had a particular focus on distribution grids with a high number of small distributed energy resources.

The objective was to be achieved through advances in three different domains: Model-based, cyber-physical intrusion detection, cyber-vulnerability assessment and power system impact analysis. Furthermore, it aimed at investigating the challenge of combining the heterogenous output from these detection domains into an integrated analysis of the current system state.

1.4.4 Project structure

The project was organized in five main work packages (WP1-5), in addition to a project management work package (WP0) and a dissemination task (WP6).

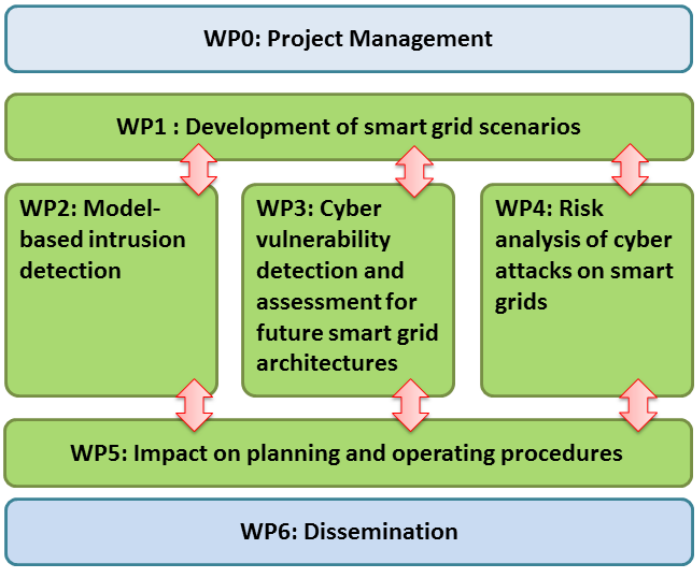

Figure 1: Overall project structure

The project was designed to start with a joint activity in WP1, the development of case scenarios for use across all other work packages. The scenarios were supposed to reflect a variety of aspects of smart grid operation as well as specific areas of interest of the three project partners. Once an initial set of scenarios would be agreed upon, the project would split into three parallel activities, each led by one of the partners, and each developing a set of tools for a different aspect of cybersecurity analysis: WP2 (lead by DTU) would investigate methods for the novel idea of cyber-physical intrusion detection, WP3 (lead by KTH) would further develop existing cyber vulnerability assessment capabilities, and WP4 (lead by PWR) would develop a tool for the impact analysis of specific cyber attacks on the power grid.
In the final phase of the project, WP5 would seek to develop a framework able to combine the very different types of information generated by these tool sets into an overall cybersecurity assessment.

1.4.5 Project execution

The project generally followed the planned trajectory, however one factor had a significant influence on the outcome of the project. Work in WP4 never progressed beyond the initial deliverable D4.1, due to non-performance of the responsible partner (PWR). It appears that this was at least in part owed to the narrow focus of the Polish funding agency on the number of publications, rather than overall project success, as a performance indicator. While PWR has published a lot within the project, these publications were mostly not related to the work plan of WP4, and were not contributing to project progress. Due to the particular construction of ERA-NET projects where every partner is responsible to their national funding body, both the present and previous project manager do not feel they have had the necessary leverage to motivate non-participating partners.

As a result, WP5 which was meant to integrate the output of three sets of tools (intrusion detection, cyber vulnerability analysis and grid impact analysis) did not receive any input on the grid impact analysis aspect. WP5 has finalized and published a concept for a (semi-) automatic hypothesis testing framework, but the original ambition of WP5, the system integration of three tool sets, had to be abandoned due to the lack of WP4 contributions.

The missing grid impact analysis also had an impact on WP6, where the lack of insight on the power system impact of the chosen scenarios made the original dissemination strategy of the project obsolete. Instead of the stakeholder workshop, a more academic dissemination strategy was chosen. The project co-organized two workshops on smart grid cybersecurity together with the EU FP7-funded partner projects SPARKS and SEGRID.

1.5 Project results and dissemination of results

1.5.1 Results overview

The project was started to develop better support for managing and designing a secure future smart grid. Its particular focus has been on smart low-voltage grids with many small distributed energy resources. There are three main project results:

1. DER intrusion detection. The composition of the infrastructure in future smart grids will be significantly more dynamic than the systems in use today. This makes recognizing cyber attacks on the components of the power system more complicated and poses a challenge to network intrusion detection systems (IDS) which are one of the most important defenses against cyberattacks on ICT infrastructures. One of the weaknesses of existing IDS is that the detection of anomalies is limited to network traffic and does not take the physical behaviour of the connected devices into account. Among the attacks most difficult to detect in this way would be those which do not significantly change the communication to and from a controlled component, but only change the way a component's control system is behaving. SALVAGE has worked on developing methods for model-based intrusion detection for power system components through real-time analysis of component behavior.

2. Vulnerability quantification. Another aspect of the assessment is to estimate how difficult it is to compromise smart grid ICT architectures as a whole. With smart grids as systems-of-systems, vulnerabilities in any part of the system could potentially lead an attacker to end up reaching all other parts of the system. Cyber security thus needs to be estimated, managed and designed on this system-of-systems level to ensure cyber security of smart grids. SALVAGE has addressed this aspect by refining the Cyber Security Modeling Language (CySeMoL), a framework for making predictions of how vulnerable a certain ICT architecture is towards cyber attacks by using attack graphs.

3. Multi-domain analysis. Traditionally, and from the point of view of a control room operator in an electrical power system, the ICT aspect of the power system (which includes the SCADA and DMS systems) has not been part of the system to be operated; it was an infrastructure which was assumed to work. Today, the ICT domain (both OT and IT) is explicitly taken into account during operations. However, classical ICT risk assessment still tends to separate security relevant events from ICT and physical domains. Cross-dependencies between domains make it hard to isolate the impact analysis of the physical and cyber domains. Consequences of IT-domain breaches which manifest themselves in the physical domain, are not quantifiable using the same metrics as a pure analysis of the IT domain. Furthermore, the model types and propagation mechanisms are different in each of these domains. The SALVAGE project has worked towards a proof-of-concept of a framework that performs an integrated assessment of the state of the power grid, using aspects from multiple domains (power system impact, intrusion detection, cyber vulnerability) and multiple sources of input (power system measurements, distributed energy resources (DERs), IT and OT systems) -- a framework that aims to provide a joint prioritization of possible threat/impact-scenarios, taking into account uncertainty of input information.

The following subsections will describe these results in more detail.

1.5.2 Scenarios

In total three scenarios were formulated for the project. The first scenario, called PowerCap, relates to power peak shaving in low voltage grids, and has been chosen for its simplicity. The scenario is described in section 1.5.2.1. The second scenario is related to protection and control and described in section 1.5.2.2. The third scenario is related to smart metering and observability on the distribution grid level and described in section 1.5.2.3.

All three scenarios build on a single comprehensive IT architecture, in order to remain as realistic as possible in terms of what IT is involved - not necessarily just in terms of the systems and functions most immediately relevant to the scenario itself, but also the other systems actually present in the networks and thus relevant for the possibilities of cyber adversaries, which can use any present systems irrespectively of their logical relation to a given scenario.

1.5.2.1 The PowerCap scenario

The scenario (described in more detail in [1] and [2]) is set in a low-voltage (LV) distribution grid, on a grid feeder in a predominantly residential area. Connected to the feeder are mostly building loads and a small number of distributed generation units (photovoltaics in this case). Some of the buildings and PV units can be externally controlled in order to offer flexibility services to the grid.

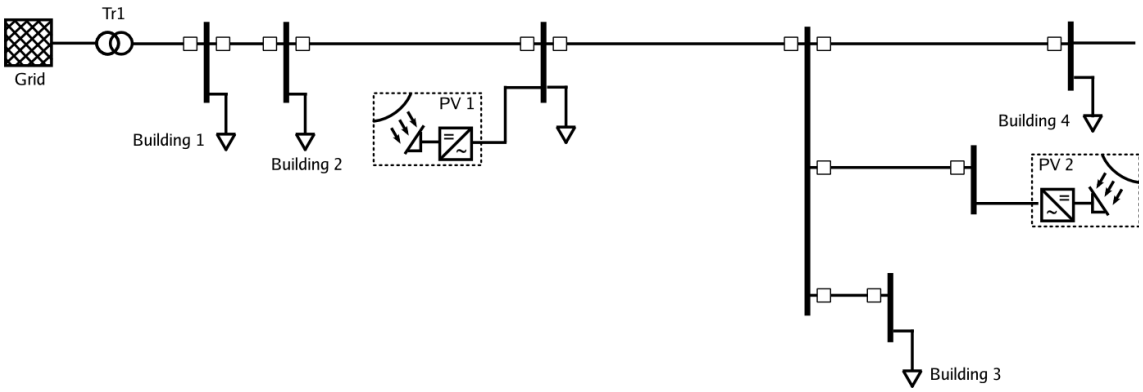

Figure 2: Power scheme of the PowerCap scenario

The flexibility service chosen for this scenario is PowerCap. The service is used by a distribution grid operator (DSO) to ensure that certain assets in the grid, such as transformers or cables, do not get overloaded in extreme power flow situations. In order to achieve this, the DSO periodically measures the power flow through the asset in question and, if the asset is loaded higher than a preset limit, asks the flexible assets to reduce their consumption or to increase their production by the difference amount between load and load limit.

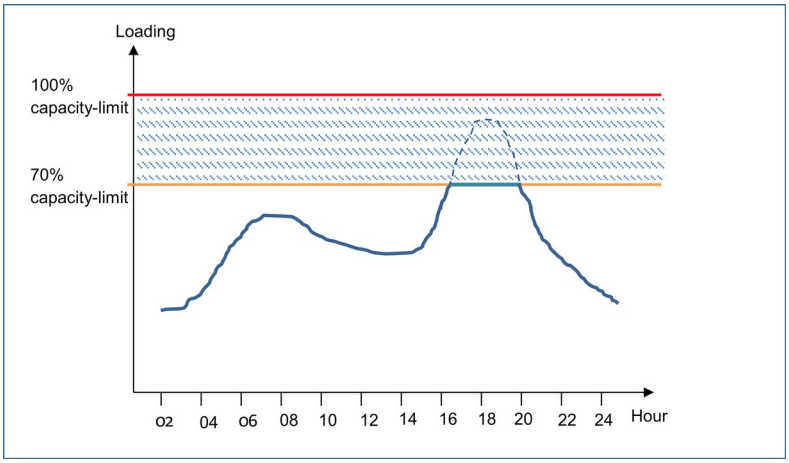

Figure 3: Illustration of the flexibility service function

Because of the stochastic nature of the individual units, they are bundled or aggregated in order to be able to offer a consistent service. The full PowerCap scenario covers three phases or stages relevant from a communication/cybersecurity point of view:

- Scheduling, which covers the interactions before the operating hour, including the commitment/bidding processes for the service,

- Operation, which covers the real-time, closed-loop control process during the operating hour, and

- Settlement, which involves validation and billing of the delivered service.

The project has focused on the operation stage. Possible extensions of the scenario are listed further below.

Actors

The basic scenario has three actors:

- A DER owner. The DER owner operates a production, consumption or storage unit which is connected to the distribution grid. The unit usually fulfills a primary purpose for the owner; as a secondary purpose, flexibility in consumption of production patterns (such as e.g. time-shifting of consumption) can be provided to the grid as long as the impact on the fitness for the primary purpose is not affected.

- A distribution system operator (DSO). The DSO owns and operates the infrastructure required for power delivery. This includes measurement and control infrastructure (SCADA and substation automation).

- An aggregator. The aggregator has contracts with a portfolio of DER units in which these DER units offer flexibility against payment. The aggregator operates the technical infrastructure to communicate with the DER units and is financially responsible in case of missing service delivery.

To cover the scheduling and settlement stages, additional actors might be added at a later point in time.

Data flow

For the operation case, the flow of data consists of the following steps:

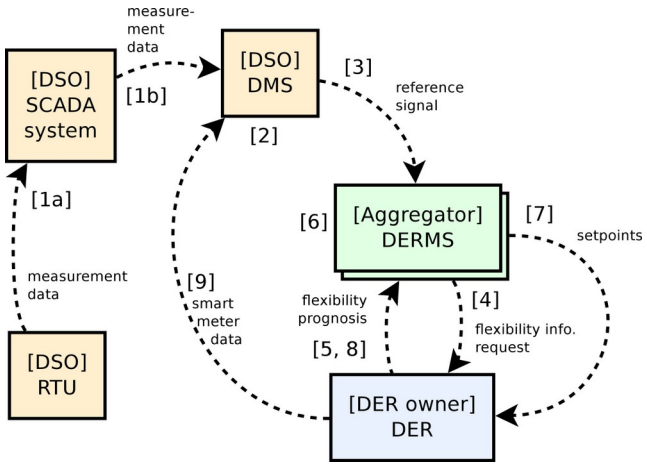

Figure 4: Data flow in operation

1. The SCADA system reads measurement data from Remote Terminal Units (RTU) in the field (e.g. in substations) and delivers the data to the distribution management system (DMS).

2. A state estimator in the DMS calculates power flow estimates for all grid assets. If any of the assets are loaded above the limit, the DMS calculates the inverted difference as a reference signal.

3. The DMS sends a reference signal to one or several aggregators. In the case where several Aggregators are jointly providing the PowerCap service, the signal will be split and be sent to all contracted aggregators corresponding to each aggregator's proportional share in the installed capacity or service commitment.

4. The aggregator requests flexibility information from all DER units in its portfolio.

5. The DER units respond with a flexibility prognosis.

6. The aggregator performs an internal optimization of its portfolio, in order to be able to deliver the service in the cheapest and most optimal way.

7. The aggregator sends set-points to all connected units and requests flexibility updates.

8. The DER units respond with an updated flexibility prognosis.

9. Smart meters at the DER owner provide measurements to the DSO.

Items 1-3 as well as 4-8 are executed continuously and independently. Item 9 provides power measurement feedback directly from the DER owner. The data flow described above is depicted in figure 4.

IT architecture

This section provides a brief picture of the IT architecture of the PowerCap scenario, as well as the other scenarios described further below. The architecture consists of several subparts (SCADA, substation automation, AMI etc.) which can be varied in more detail, for example, how many substations the example includes. Moreover, the architecture presented here does not describe technical details and assumptions regarding the different systems, for example, what protocols are used, what types of hardware or software, etc. This is elaborated in more detail in section 1.5.4.

Figure 5 outlines the high level IT architecture model of the proposed scenario. Submodels of the overall architecture (highlighted in yellow) are covered in figure 6 (SCADA reference model), figure 7 (substation IT architecture), figure 8 (Aggregator IT architecture and DER control), and figure 12 (AMI reference model). All acronyms in the figures are referenced in the glossary section.

The models presented in this section are based on [3], [4], [5], among others.

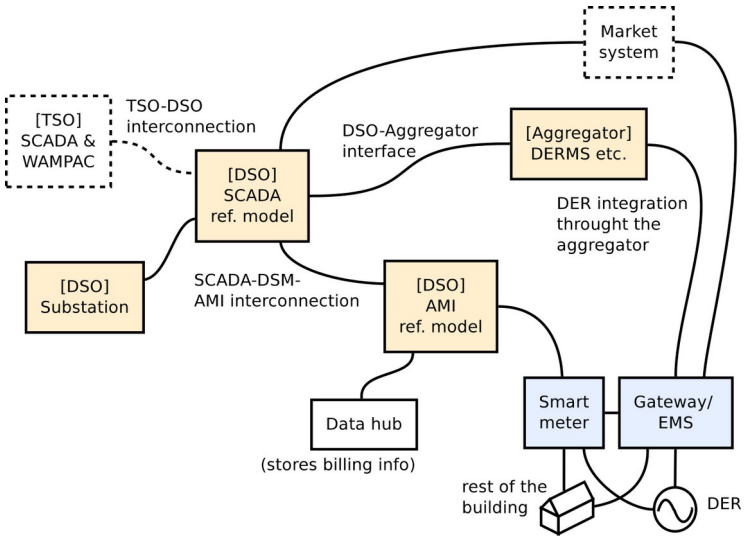

Figure 5: Overall IT architecture model

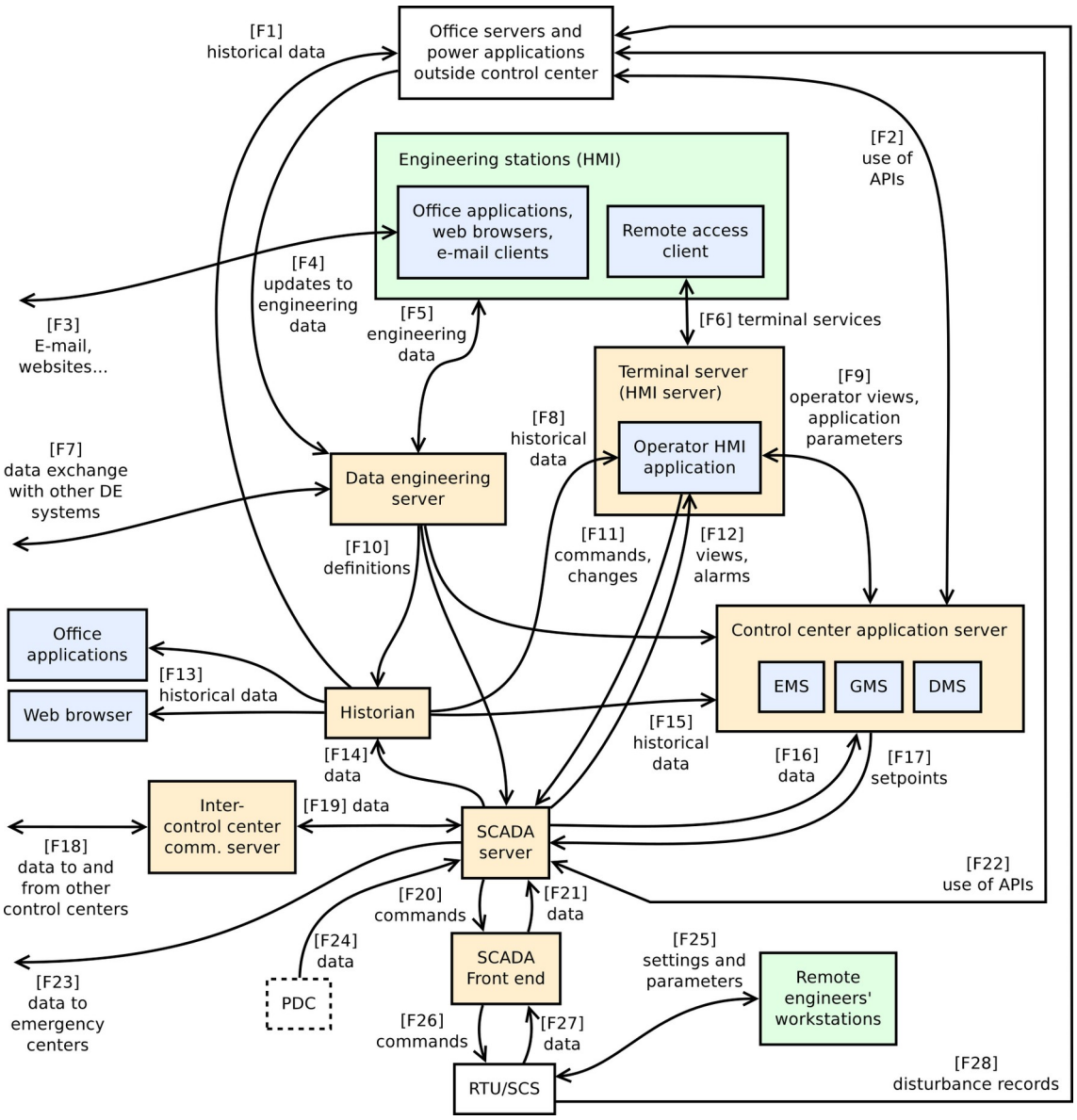

Figure 6: SCADA reference model

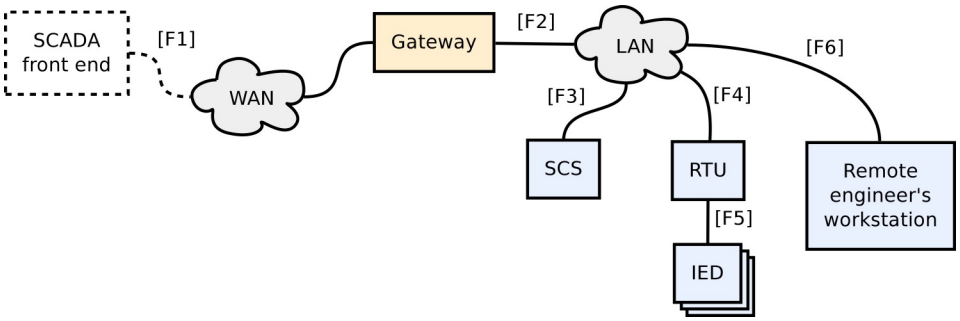

Figure 7: Substation IT architecture model

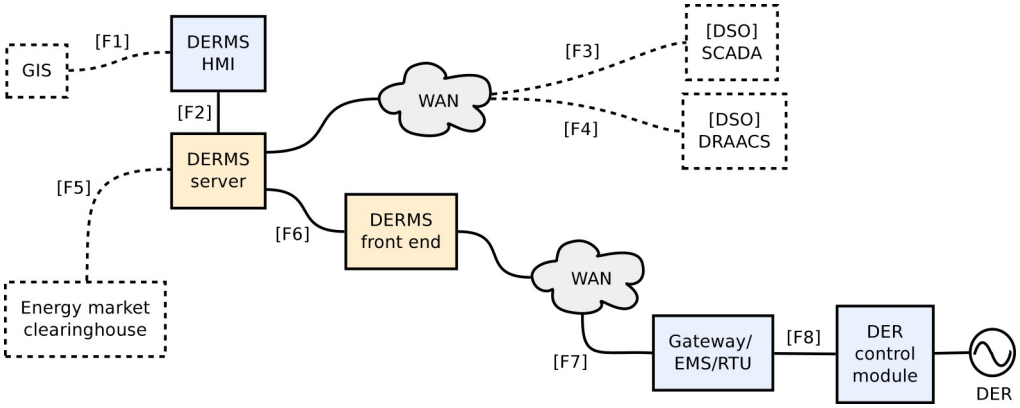

Figure 8: Model of the process control related IT architecture of an aggregator

Variations, extensions and further remarks

Within the above described PowerCap scenario, the following variations can be considered:

- In case of multiple aggregators, reference signal in step 4 is divided between several aggregators.

- Scheduling and settlement stages.

According to all of the three communication/cybersecurity stages of the PowerCap scenario (see above), a separate reference model for DRM (demand response management) might be relevant to consider.
In certain cases, DRM also uses AMI as its communication infrastructure between the DSO and its customers (e.g., to issue load shedding notifications to smart meters). Normally, DRM communication directed to and from DERs flows through the Aggregator.

Additional systems of supportive nature that might be considered upon need are the DSO's GIS (geographical information system), asset and facility management system, the DSO's and the aggregator's CIS (customer information system), and the DSO's and the Aggregator's office IT systems, which are, although less directly, related and connected to the process.
In likeness with the SCADA and AMI reference models depicted above, even office computers with applications might be beneficial to consider in other parts of the architecture (e.g., Aggregator).

1.5.2.2 Protection and control scenario

This scenario focuses on studying cyber-physical security and intrusion detection at the level of a DSO's power grid. It also studies possibilities for adjusting the selectivity of power system protection functions. Most attention in this scenario is directed at the power grid topology, the grid's physical components, and the process observability and control equipment (e.g., sensors, actuators, control systems), regarding both its physical environment and its cyber environment to the extent necessary.
This scenario presupposes a fully automated grid.
The power grid topology considered in this scenario can be seen in figure 9.

Actors

This scenario has a single main actor: a DSO. The DSO owns and operates the power system infrastructure required for power delivery. This includes the measurement and control infrastructure in a control center (i.e., SCADA) and substations (i.e., substation automation).

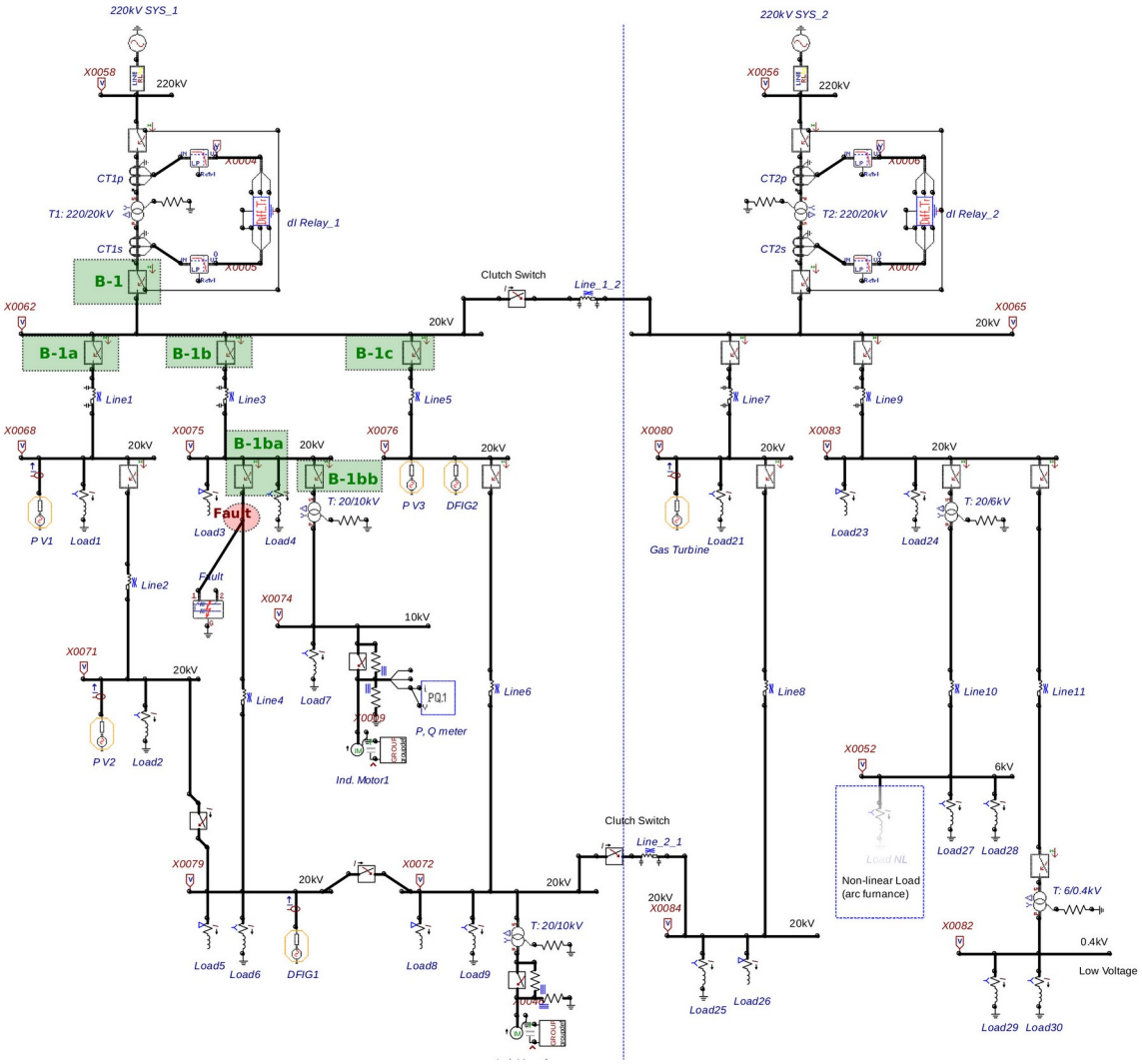

Figure 9: Power grid topology for the protection and control scenario

As highlighted in figure 9, there is a fault between the breaker marked B-1ba connected to busbar X0075, and Line 4. The nearby breakers of most interest have been marked according in the figure (see the green rectangles containing the breaker and a symbol).

IT infrastructure

The IT architecture of the scenario reuses the IT architecture described in section 1.5.2.1 (the PowerCap scenario), especially the industrial control components in substations (e.g., RTUs, IEDs) and in the SCADA system - the IT infrastructure directly supporting the protection and control functions.

1.5.2.3 Smart metering and DSO observability scenario

This scenario focuses on the cyber-security, cyber-physical security and intrusion detection on the level beyond the visibility provided by the equipment in the DSO's power grid. In other words, it relates to the Advanced Metering Infrastructure (AMI) of the DSO, and its functions providing observability and control complementary to that of the power grid equipment (e.g., SCADA, IEDs, etc.). This is expected to be more common and prevalent compared to the traditional scenario of relying only on consumer-level power metering eventually complemented by power grid measurements from the TSO level substations.

Actors

This scenario has two main actors:

- A DSO. The DSO owns and operates the power system infrastructure required for power delivery. This includes the measurement and control infrastructure in a control center (i.e., SCADA) and substations (i.e., substation automation).
- A customer (or several customers). The customer has a household (a private person) or a company/industrial premise such as a factory or office space (an organization).

IT infrastructure

The IT architecture of the scenario reuses that described in section 3, however, with the dominant focus on the AMI infrastructure and its immediate physical and cyber environment. A brief overview of the core connections of an AMI infrastructure is in figure 10, and an overview of its network layout in figure 11. An overview of the AMI reference model can be found in figure 12.

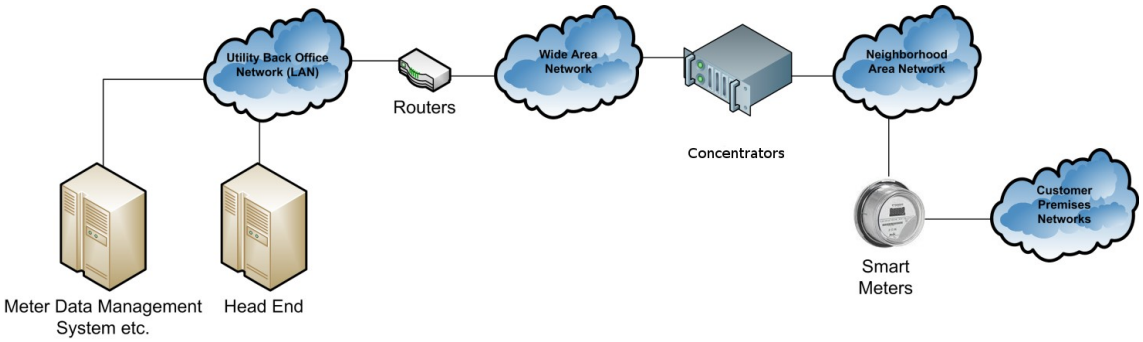

Figure 10: Brief overview of the core AMI system connections

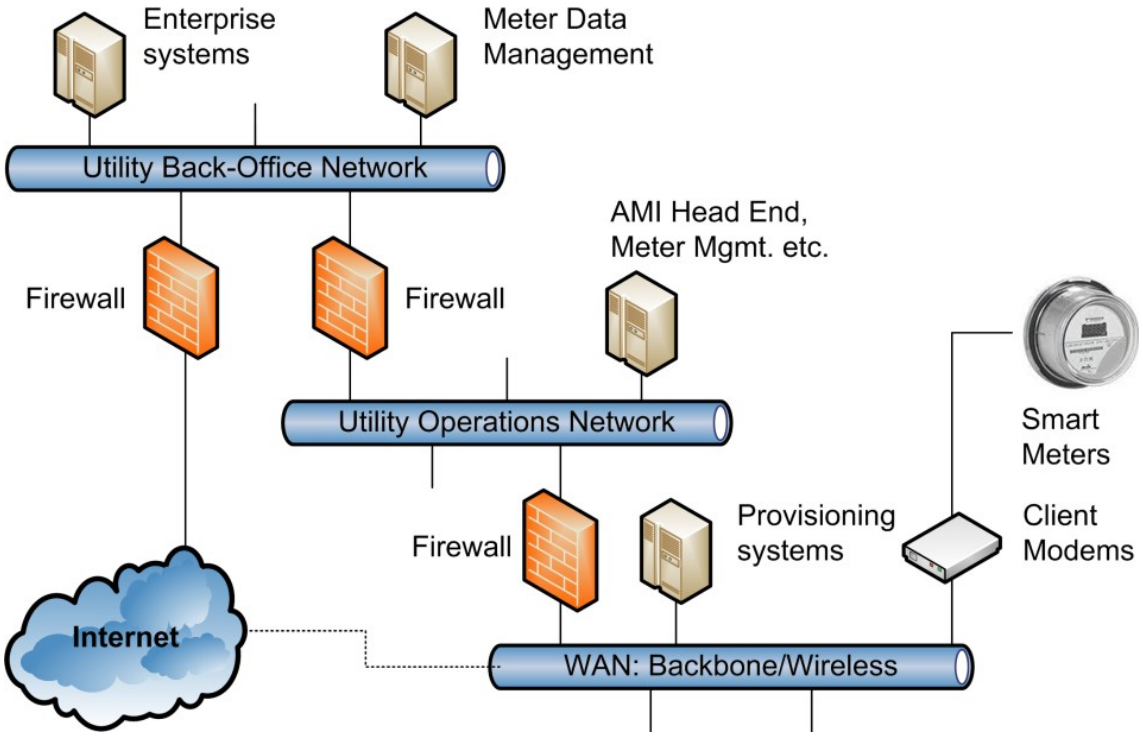

Figure 11: Brief overview of an AMI infrastructure network layout

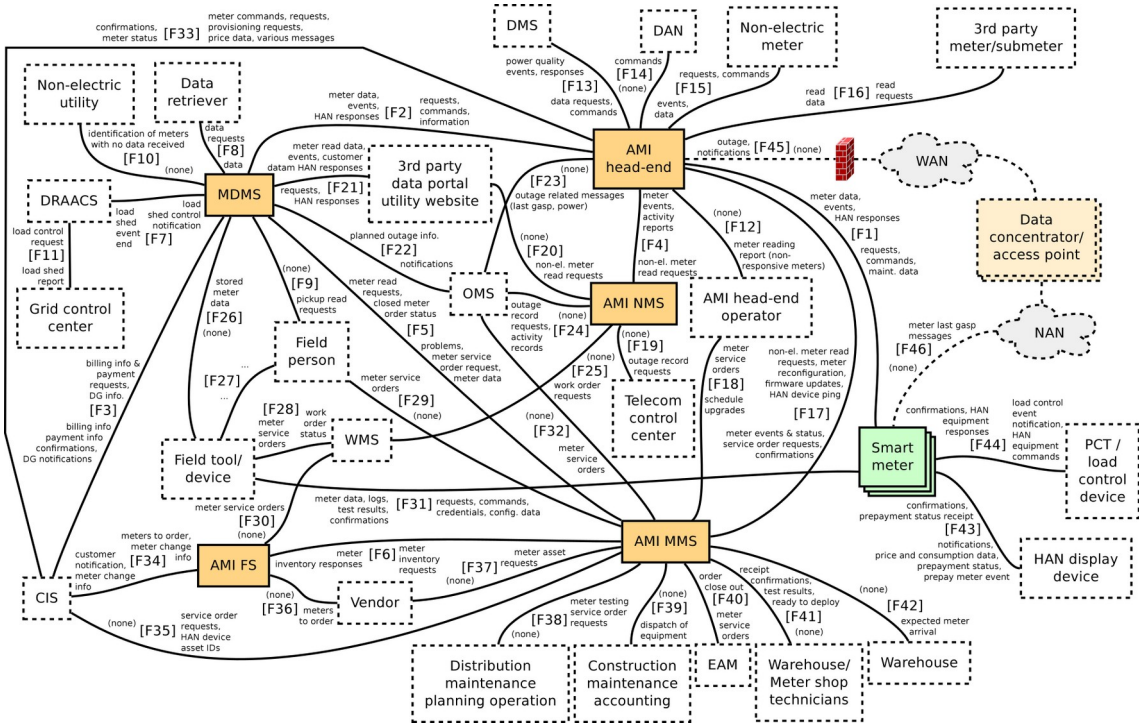

Figure 12: Overview of the AMI reference model

1.5.3 Threat analysis

1.5.3.1 PowerCap scenario

Adverse events and attack goals

The adverse events in the scenario include the following:

- Loss of supply (blackout). E.g., a protection failure at a lower level might fail to isolate a fault resulting in permanent infrastructural damage causing blackout, or a higher-level protection mechanism cutting supply at more customers than neccessary. Alternatively, a false triggering of a protection mechanism might trip breakers and cut off power supply to an unspecified number of customers. Another example includes an error in the process of adjusting set-points for distributed power production, which might result in overloading the input from the main power grid, and causing a blackout.

- Damage to infrastructure or equipment. E.g., equipment overload of transients might result in the power infrastructure and its equipment being damaged.

- Damage to customer equipment. E.g., outages and degraded power quality due to poorly functioning protection and control functions at the infrastructure level, might result in customer equipment being damaged.

- Reduction of power quality. E.g., faults or imperfections in, or attacks on the control of the power grid including the distributed power generation might result in the reduction of power quality at the customers.

- Privacy violations. E.g., an outside adversary (e.g., an Internet hacker with a specific harmful intent) compromises the confidentiality of customer power consumption records and other customer data (e.g., payments etc.).

- Market manipulation (destabilization, fraudulent/harmful bidding).

- Damage to goodwill (reputation, trustworthiness). E.g., failure to deliver up to agreed levels of power uptime or quality, might result in the DSO's goodwill being damaged. Similarly, incorrect billing of customers (e.g., due to compromised metering or billing records), can result in unwanted publicity and damaged goodwill (of the DSO).

- Loss of revenue. E.g., the need to replace pieces of infrastructural equipment (e.g., due to damage from overload or adversarial incidents including cyber attacks) might inflict heavy operational spending and affect the revenue (of DSOs and DER owners). Similarly, poor power delivery or degraded operations might result in loss of revenue for the DSO. Yet another example includes the failure of the Aggregator to optimize the operation of the DER portfolio, which might result in a loss of revenue (of the Aggregator and DER owners).

- Reduction of operational decision capability due to bad information (e.g., planning, forecasting).

- Degraded condition monitoring / degraded asset management. E.g., compromised information lines from field devices to other field devices or control system components, missing or incorrect measurements, might degrade condition monitoring or asset management at the DSO. Similar situation can occur at the Aggregator and DER owner.

- Suboptimal operation of the grid (specifically increased losses). E.g., compromised power grid monitoring (e.g., missing or incorrect measurements, or equipment malfunctions) might affect the efficiency of the power grid operation, including the distributed power generation.

- Manipulation of bilateral contracts. E.g., an adversary might launch an IT attack and inadvertently modify contractual information at some of the actors, particularly the Aggregator and the DSO.

- Legal damage. E.g., a failure to deliver services up to the agreed levels of quality and other attributes might result in legal liability for all actors involved in the scenario.

- Espionage. E.g., an adversary might use an IT attack to extract sensitive information regarding customers and distributed energy producers (e.g., high-resolution power consumption and production data, billing information, contact information); power grid topology and technology used in the power infrastructure.

Several of the above mentioned adverse events / attack goals are detailed below.

A) Damage to infrastructure or equipment

Damage mechanisms:

1. Sustained thermal overloading of transformers or lines/cables; and

2. Insulation degradation by voltage overloading of e.g. cables and insulators.

Physical triggers:

1. Failure to adjust power production or consumption in a critical grid situation (e.g., high load); and

2. Failure to adjust active or reactive power injection in order to stay within voltage limits.

Trigger functions:

1. A false setpoint being applied to a DER unit; and

2. A false measurement feedback returned from a DER unit.

B) Damage to customer equipment

Damage mechanisms:

1. Overvoltage;

2. Cycling of demand response units, e.g., compressor-based equipment; and

3. Disruption of critical processes by supply interruption (e.g., fridge).

Physical triggers:

1. Failure to adjust active or reactive power injection in order to stay within voltage limits; and

2. Direct adjustment of setpoints.

Trigger functions:

1. A false setpoint being applied to a DER unit; and

2. A false measurement feedback returned from a DER unit.

C) Reduction of power quality

Damage mechanisms:

1. Over- and undervoltage; and

2. Voltage oscillation and flicker.

Physical triggers:

1. Failure to adjust active or reactive power injection; and

2. Induction of rapid production or consumption changes by influencing DER units.

Trigger functions:

1. False setpoint applied to a DER unit (in particular droop settings are interesting for flicker induction); and

2. False measurement feedback returned from a DER unit.

D) Market manipulation (market destabilization, fraudulent/harmful bidding)

Damage mechanism:

Cobweb effect.

Physical triggers:

Synchronization of kickback effects in DER control (mostly thermostatically controlled load).

Trigger functions:

1. False setpoint applied to a DER unit; and

2. False measurement feedback from a DER unit (e.g., in order to disturb an aggregator's anti-kickback).

E) Reduction of operational decision capability due to bad information (e.g., planning, forecasting)

Damage mechanisms:

1. Failure to track system state at DSO;

2. Failure to track system state at Aggregator; and

3. Failure to perform correct forecasts (loads, flexibility).

Physical triggers:

1. Compromising of historian;

2. Compromising of SCADA database;

3. Compromising of substation RTUs and other data ackquisition equipment such as IEDs;

4. Compromising of of HMI; and

5. Compromising of smart metering system.

Trigger functions:

1. Direct manipulation of databases and/or applications;

2. Manipulation of communication (RTU central entity);

3. Manipulation of RTUs;

4. Creation of false external data;

5. Injection/interception of [false] external data;

6. Manipulation of smart meters; and

7. Injection/interception of [false] smart meter data.

1.5.3.2 Protection and control scenario

Adverse events and attack goals

The adverse events in the scenario include the following:

- Loss of supply (blackout). E.g., false triggering of a protection mechanism might trip breakers and cut off power supply to an unspecified number of customers.

- Damage to infrastructure or equipment. E.g., equipment overload of transients might result in the power infrastructure and its equipment being damaged.

- Damage to customer equipment. E.g., outages and degraded power quality due to poorly functioning protection and control functions at the infrastructure level, might result in customer equipment being damaged.

- Damage to goodwill (reputation, trustworthiness). E.g., failure to deliver up to agreed levels of power uptime or quality, might result in the DSO's goodwill being damaged.

- Loss of revenue. E.g., the need to replace pieces of infrastructural equipment might inflict heavy operational spending and affect the revenue of the DSO.

- Degraded condition monitoring / degraded asset management. E.g., compromised information lines from field devices to other field devices or control system components, missing or incorrect measurements, might degrade condition monitoring or asset management at the DSO.

- Suboptimal operation of the grid (specifically increased losses). E.g., compromised power grid monitoring (e.g., missing or incorrect measurements, or equipment malfunctions) might affect the efficiency of the power grid operation, including the distributed power generation.

- Espionage. Although espionage is hardly a direct and immediate threat to power delivery; an adversary might launch a cyber attack to extract sensitive information about the power grid topology, its operating limits, and technology used to monitor, protect and control the power infrastructure. Doing so could enable the adversary to subsequently launch an attack to effectively sabotage the power infrastructure or its parts, causing a blackout and/or causing permanent damage to equipment.

Two of the above mentioned adverse events / attack goals are detailed below, in a combined manner.

Loss of supply (blackout) and damage to infrastructure or equipment

The considered attack goals are (1) [partial] blackout and (2) damage to equipment, which can either be (2a) generators or (2b) grid components.

Damage mechanisms:

1. Disconnection, loss of supply;

2. Islanding, lack of appropriate supply; and

3. Overloading of equipment (both mechanically and electrically).

Trigger functions:

1. Blocked or delayed tripping; and

2. Unwanted or premature tripping.

Means:

1. Changing relay settings (e.g., via engineering / configuration files) - both remotely (e.g.,
via remote shell access or a ftp upload), and using local physical access;

2. Online interventions:

a) Line protection (e.g., communication between distance protection relays, power-line communication or dedicated communication lines);

b) Bus protection (e.g., GOOSE over IP or Ethernet) - denial of service, or injection of
false data/commands.

There are two subscenarios in this scenario, detailed below.

Subscenario 1: Reconfiguration after a fault (circuit breaker can be configured)

Attacks:

1. Configuration has been modified;

2. Breaker state data modified;

3. Modification of interlocking at substation;

4. Measurements (voltages and current) modified.

Consequences:

1. No supply for loads (possibly blackout);

2. Safety maintenance crew;

3. Overcurrent, fault current;

4. Overvoltage damage to equipment;

5. Damage to substation equipment.

Subscenario 2: Adaptive protection setting

Attack: Protection setting has been modified.

Extended actors:

1. DSO (control room, operator)

2. Customer

3. Maintenance staff

4. Device supplier

5. Telecommunications infrastructure operator

6. Attacker

Attack targets:

1. Substation

2. SCADA system

1.5.3.3 Smart metering scenario

Adverse events and attack goals

The adverse events in the scenario include the following:

- Loss of supply (at household / customer premises). E.g., due to false curtailing signals sent to the customer(s).

- Damage to customer equipment. E.g., due to a major power quality deviation.

- Privacy violations. E.g., an outside adversary (e.g., an Internet hacker with a specific harmful intent) compromises the confidentiality of customer power consumption records and other customer data (e.g., payments etc.).

- Market manipulation (destabilization, fraudulent/harmful bidding).

- Damage to goodwill (reputation, trustworthiness). E.g., incorrect billing of customers (e.g., due to compromised database records), can result in unwanted publicity and damaged goodwill.

- Reduction of operational decision capability due to bad information (e.g., planning, forecasting). E.g., compromised information lines between equipment and systems handling the inventory and management of equipment (e.g., smart meters, data concentrators), can undermine operational decision-making at the DSO.

- Loss of revenue. E.g., poor power delivery, degraded operations or excessive need to replace equipment due to adversarial incidents including IT attacks, can result in loss of revenue for the DSO.

- Suboptimal operation of the grid (specifically increased losses). E.g., in case the DSO relies on metering data for the purposes of power grid optimization (especially on the lower voltage levels, parts of the grid close to the customers), missing or inaccurate such data might result in suboptimal grid operation.

- Legal damage. For similar reasons than those leading to damage to goodwill (see a point above), the DSO might even face legal damage.

- Espionage. An adversary might use an IT attack to extract sensitive information regarding customers (e.g., high-resolution power consumption data, billing information, contact information), or used technology.

1.5.4 Reference models and templates

The scope of the threat analysis has been the assessment of cyber vulnerability in future smart grid architectures. Its purpose was to estimate how difficult it is to compromise smart grid ICT architectures - architectures consisting of a large number of various ICT components, interconnected within a complex system-of-systems, which includes units for distributed control and monitoring (e.g., sensors, actuators, controllers), control applications, data concentrators, communication infrastructure (e.g., routers, switches), enterprise IT systems, databases and more. Unfortunately, all of the constituents of smart grid ICT architectures alone tend to be cyber vulnerable.

Some such cyber vulnerabilities are widely known and remediable to an extent, while many others are yet to be found, or worse, are already known to a potential antagonist, but not to the general public and the electrical utilities and other organizations that operate the systems. The difficulty to secure the systems and architectures stems even from their large extent, high complexity and diversity, heavy interconnectedness, and the unavoidable presence of the human factor. In spite of all this however, smart grid ICT architectures need to operate in a safe and reliable manner, for which an adequate level of cyber securement is a necessity. Since the proliferation of skill and organization enabling people to perform sophisticated cyber-attacks continuously rises, as well as the availability of various means and tools that enable and simplify the processes of cyber reconnaissance and attacking, so

also new means (e.g., mechanisms, approaches, methods and tools) for defending cyberspaces must, too, advance.

This work package addresses the challenge by applying a method of probabilistic assessment of cyber vulnerability to smart grid ICT architectures, moreover, with the vision to constitute a part of a more comprehensive framework for the assessment and detection of vulnerabilities of smart grids, which incorporates the cyber-physical aspects of smart grids as a functional whole rather than their ICT architectures alone.

The first step includes the process of identification and prioritization of candidates for reference models (also called templates) capturing important cohesive parts of smart grid ICT architectures. A reference model denotes a model that describes a generic piece of some architecture, according to a given syntax (in this case the Cyber Security Modeling Language introduced further below), carrying a volume of information about the modeled piece of architecture, which has a high potential of being reused - both within the same architecture, and between two architectures (e.g., two architectures of two different electrical utilities).

1.5.4.1 Assumptions and delimitations

The work related to formulating the smart grid reference models, including the prioritization of them, relies on a few rather general assumptions. A failure to meet the assumptions would make the validity of the work's results questionable. The assumptions are as follows:

1. The existence of a reliable framework for modeling and assessing cyber security (e.g., Cyber Security Modeling Language), which is applicable to the task of evaluating cyber security in a partially automated way;

2. The possibility to capture a typical smart grid ICT architecture, which can be found in most smart grid environments, with some degree of variation from one to another, and treat it as a generic such architecture;

3. The relatively little amount of volatility of the smart grid ICT architecture, due to different technological and other changes that continuously take place. In other words, the relatively high stability of the structure and configuration of the parts of the smart grid ICT architecture over time. To date, this appears to be the case for older, more mature and well-adopted parts such as SCADA, while unfortunately less so for the newer ones such as market automation systems, or systems aggregating the control of multiple distributed energy resources (DERs).

The subsequent cyber vulnerability analysis is delimited to supporting and performing evaluation of cyber security through automated processing of ICT-architectural models.

1.5.4.2 Approach to modeling architectures and assessing cyber security

The approach to assessing cyber security applied in the project makes use of the concept of metamodeling, which is further extended by the capability to perform automated analysis. This section attempts to provide a brief introduction to the concept.

The concept of metamodeling is based on the distinction between two types or rather layers of models: a metamodel (sometimes also called a class model) and an instance model (sometimes also called an object model). A metamodel describes the syntax of instance models, and their permitted structure (i.e., classes with their properties, associations with their cardinality constraints). A corresponding instance model can then describe instances of the classes (i.e., objects with their property values), and instances of the associations (i.e., connections between objects).

To extend the above described distinction to an arbitrary number of such planes, a metamodel relates to instance models in the same way as a meta-metamodel relates to metamodels. An example of such extension is the Meta Object Facility (MOF) [6], which uses

four distinct levels (M0-M3). Thus far, the metamodeling concept could only serve to model or document something - pieces of reality or something hypothetical. In order to more fully exploit the concept's potential; it has been extended into a powerful foundation for automated analysis: Looking back at the distinction between a metamodel and an instance model, a metamodel might not only describe the syntax and the permitted structure of instance models, but also arbitrary computation of the value of an attribute of a class.

For an illustrative example, let us consider a metamodel describing two classes: company and employee. Further, let the class employee have an attribute called unavailability days a year; let the class company have an attribute called full operation probability; and let there be an association called key employment between company and employee. Even further, let the company's full operation probability have a derivation (calculation) defined such that the value [of the property of an object of the class company] reflects the need of having all key employees (i.e., all objects of the class employee that are connected to the company) available at the same time, for all companies (i.e., all objects of the class company).

This results in that whatever structure of companies and employees we model, the companies get their derived attributes (here, full operation probability) calculated in an automated way, according to the above. When modeling the instance model, one can specify the yearly days of unavailability for each employee. In a simplest way, this could be done using a scalar number. In this example however, the unavailability days are specified in a stochastic manner, as a probability distribution - normal distribution with its parameters (mean and variance). Below, the example is depicted in figure 13 (metamodel) and figure 14 (instance model). Put into the context of this project, the concept is applied to cyber security analysis, with the already indicated extension of stochastic attribute values, and described in a subsequent section of this report.

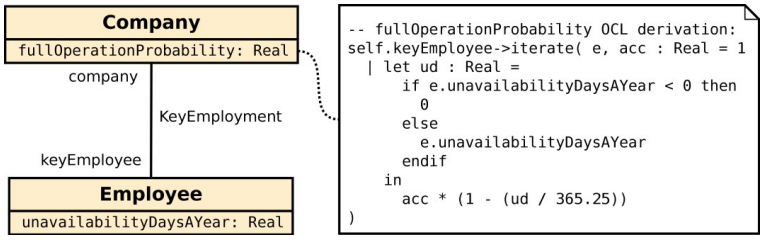

Figure 13: Illustration of an example metamodel

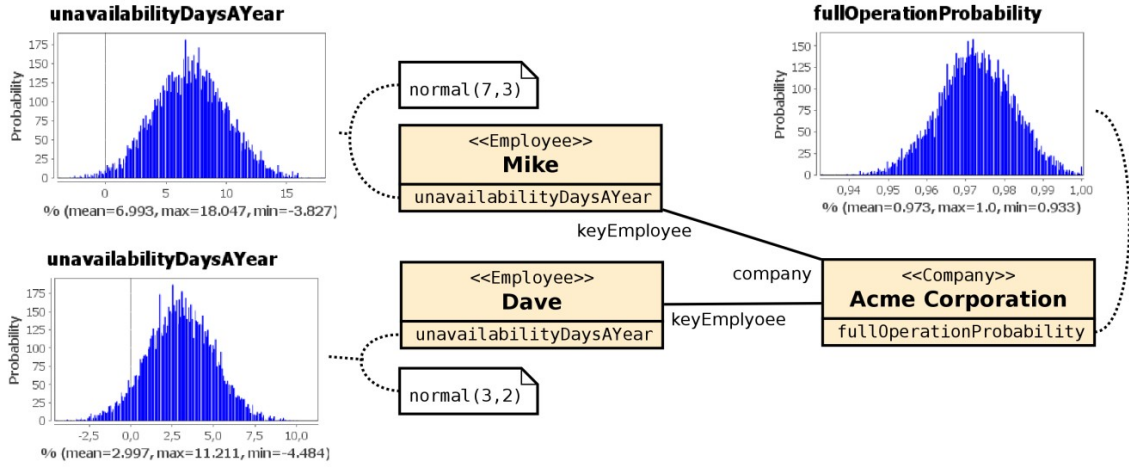

Figure 14: Illustration of an example instance model

Finally, in order to implement the concept of metamodeling extended by automated analysis, a computational engine (inference engine) is needed, typically in form of a piece of software. One such engine has been used in the project, and will be discussed later in the report.

1.5.4.3 Significance and representativeness of reference models for smart grid ICT architectures

A reference model (or template) is a reusable piece of instance model, which contains content as an instance model, and just as an instance model, corresponds to a given metamodel. This is depicted in figure 15. The use of reference models offers two major benefits, described below.

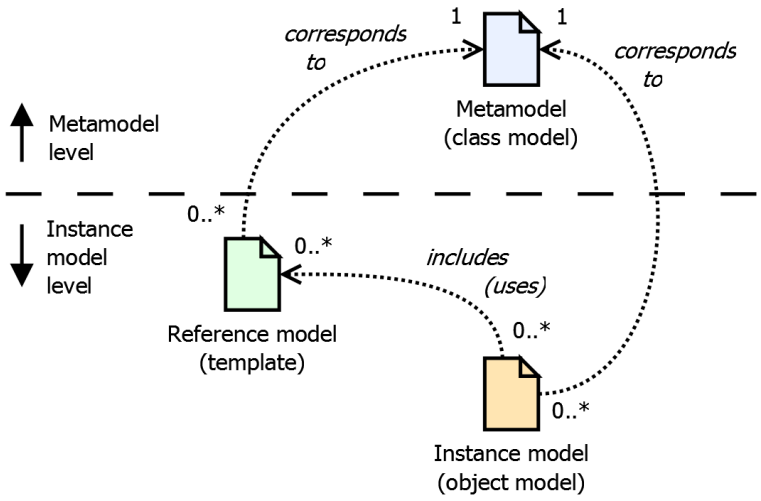

Figure 15: Reference model (template) in relation to a metamodel and an instance model

Firstly, it offers the reuse of knowledge instead of the need to elicit information and build models from scratch. Building models from scratch can be an expensive affair, and typically even more so the process of eliciting all information needed for that purpose. Using a suitable reference model offers savings in terms of the effort needed for elicitation of information and modeling - through providing a typical or generic model, which can be detailed or adjusted rather than modeled from scratch, hence requiring less effort to arrive at the same end result.

Additionally, reference models offer the possibility to describe architectures not yet present within an organization. For example, an electrical utility that has not implemented a specific system (e.g., advanced metering infrastructure), might use a reference model to see how its implementation would look like, and perform analyses over it, without the need to laboriously study documentation and perhaps use consultancy services for the purpose.

However, most importantly in context of this project, reference models offer the possibility to store numerous domain specific attributes (e.g., those of significance to cyber security), which relate to the entities in the reference model (e.g., services, systems, their usage, communication paths, etc.). Unlike the entities typically describing ICT architecture, such domain-specific attributes are considerably further away from common knowledge, and hence, even more difficult to elicit and model correctly. This is where the use of reference models offers most notable aid.

Secondly, the use of reference models eliminates or reduces necessity to repeatedly model repeating parts of a given architecture. On a smaller scale than that of SCADA or advanced metering infrastructure architectures, even within architectures such as these, models can show high repetition of some of their parts. Let us for instance consider a commonly used desktop operating system with a typical set of preinstalled software, such as Microsoft Windows 8; or a server operating system such as Red Hat Enterprise Linux. Typically, it is needed to model these multiple times across a larger model of a whole enterprise. The use of reference models as templates offers the possibility to simply refer to such a reference model (template) - instead of modeling all its details anew, or even worse, omitting the details due to the excessive effort needed to model them.

Alternatively, as described above, a reference model (template) might be used to instantiate a baseline piece of a model (e.g., an operating system instance), which is to be altered

subsequently (e.g., by removing some services installed on the operating system, adding some others, or changing some configuration parameters). The use of reference models (templates) in an embedded fashion is illustrated in figure 16.

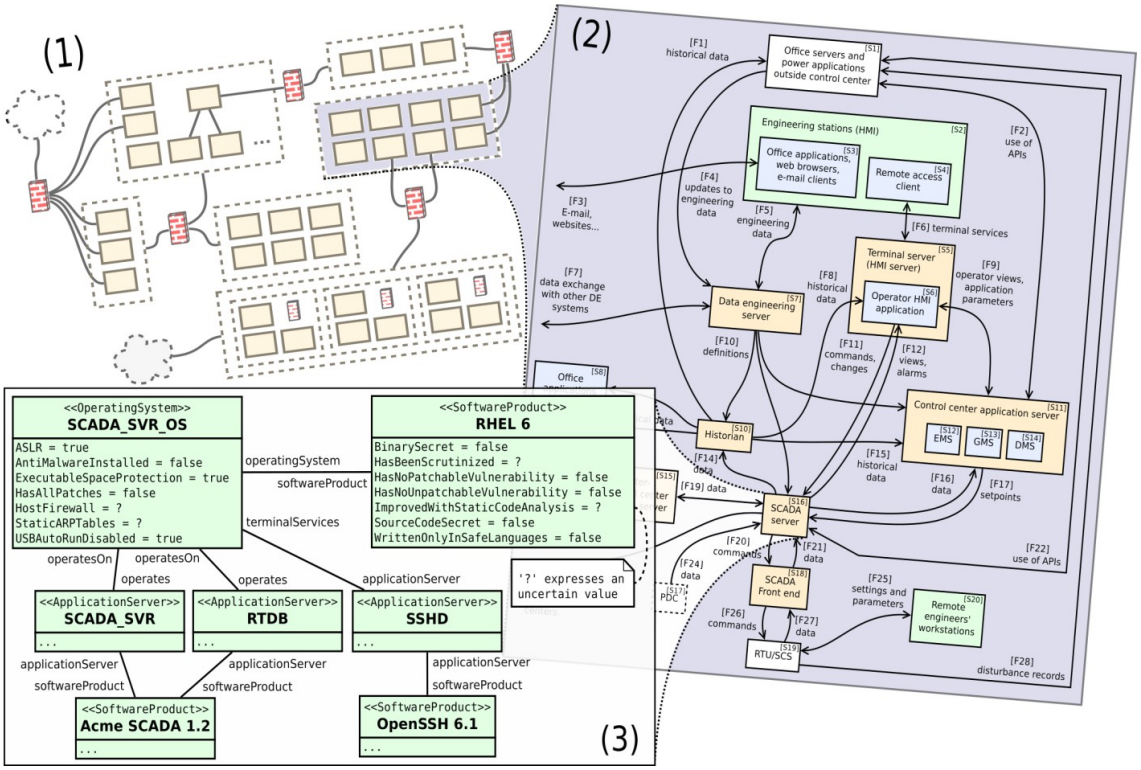

Figure 16: Illustration of the use of reference models (templates). Part (1) illustrates an overall ICTarchitecture, part (2) illustrates the SCADA reference model, and part (3) illustrates a reference modelof the SCADA server, part of the SCADA reference model.

The project aimed at developing a set of reference models for smart grid ICT architectures which show a high degree of representativeness. The representativeness of the reference models is supported by the indications of maturity and stability of the different architectural parts over time, as well as studies and sources providing both concrete and generic models of certain such parts' implementations.

1.5.4.4 Framework for modeling and automated analysis

In order to assess the cyber security (including the picture of cyber vulnerability) of an architecture, we use a framework that defines a set of concepts (vocabulary) together with their relations and assessment logic for the purposes of automated assessment. The framework consists of three parts:

1. a formalism for probabilistic inference (P^2AMF [7], [8])
2. the Cyber Security Modeling Language [9], [10]; and
3. a software engine, in which the previous can be implemented, and which performs the probabilistic inference on top of architectural models.

These three parts are tightly coupled - the second builds on and depends on the first, and the first is implemented by the third - together forming a framework that can be used to calculate models, even in a simulation-like fashion, and so produce results.
All the above mentioned parts are briefly described in the subsections below.

Predictive, probabilistic architecture modeling framework (P^2AMF)

P^2AMF [7], [8] is a formalism for arbitrary predictive, probabilistic inference, built on the bases of the Unified Modeling Language's (UML) [11] class diagrams, the Object Constraint Language (OCL) [12], and extended by the possibilities to define stochastic derivation of attribute values, among others.

P²AMF is also the formalism followed by the most recent implementation of the Cyber Security Modeling Language.

Cyber Security Modeling Language (CySeMoL)

CySeMoL [9], [10] is both a modeling language for capturing architectures (ICT and slightly beyond that) with entities and their attributes having significance to cyber security, and a logical tool for automated evaluation of cyber security (once an architecture is modeled). CySeMoL has been developed at KTH Royal Institute of Technology during the last decade, and has been chosen as the core of this project's approach to cyber security evaluation.

Although approaches, methods and solutions comparable to CySeMoL exist, their design optimizations make them less suitable for application in the context of this project. Namely, most of them suffer from being either too vague and hence overly subjective [13], too little automated and so requiring large amounts of human effort (e.g., Common Criteria [14], OCTAVE [15], CORAS [16] and the model by Breu et al. [17]), or too limited in terms of scope (e.g., MulVAL [18], [19], NetSPA [20] or TVA-tool [21]). CySeMoL is designed for evaluating cyber security of systems-of-systems-level architectures.

CySeMoL contains the following content:

- A set of classes, called assets, corresponding to the different entities modeled in the architecture under evaluation (e.g., network zone, operating system, application server, data flow, protocol, network management process, etc.). In total, CySeMoL defines 23 assets.

- A set of associations between all these assets, which allow the modeler to connect them together in order for them to represent the architecture under evaluation. In total, CySeMoL defines 51 such different associations.

- A set of defense mechanisms defined for each asset (e.g., cryptographic authentication, port security, host firewall, exclusive use of type-safe programming languages, security audit etc.). In total, CySeMoL defines 58 defense mechanisms.

- A set of attack steps defined for each asset, through which an attacker can carry out attacks (e.g., obtain own IP address, execute arbitrary code, social-engineer credentials, etc.). In total, CySeMoL defines 59 different attack steps.

- Derivations of attribute values to calculate default values for certain defense mechanisms (unless explicitly specified).

- Derivations of attribute values of attack steps, which simulate the propagation of attacks (i.e. probability and/or effort-constrained transitions from one attack step to another over the architecture under evaluation), following the logic of attack and defense graphs.

An illustrative overview of the CySeMoL metamodel's structure is provided in figure 17.

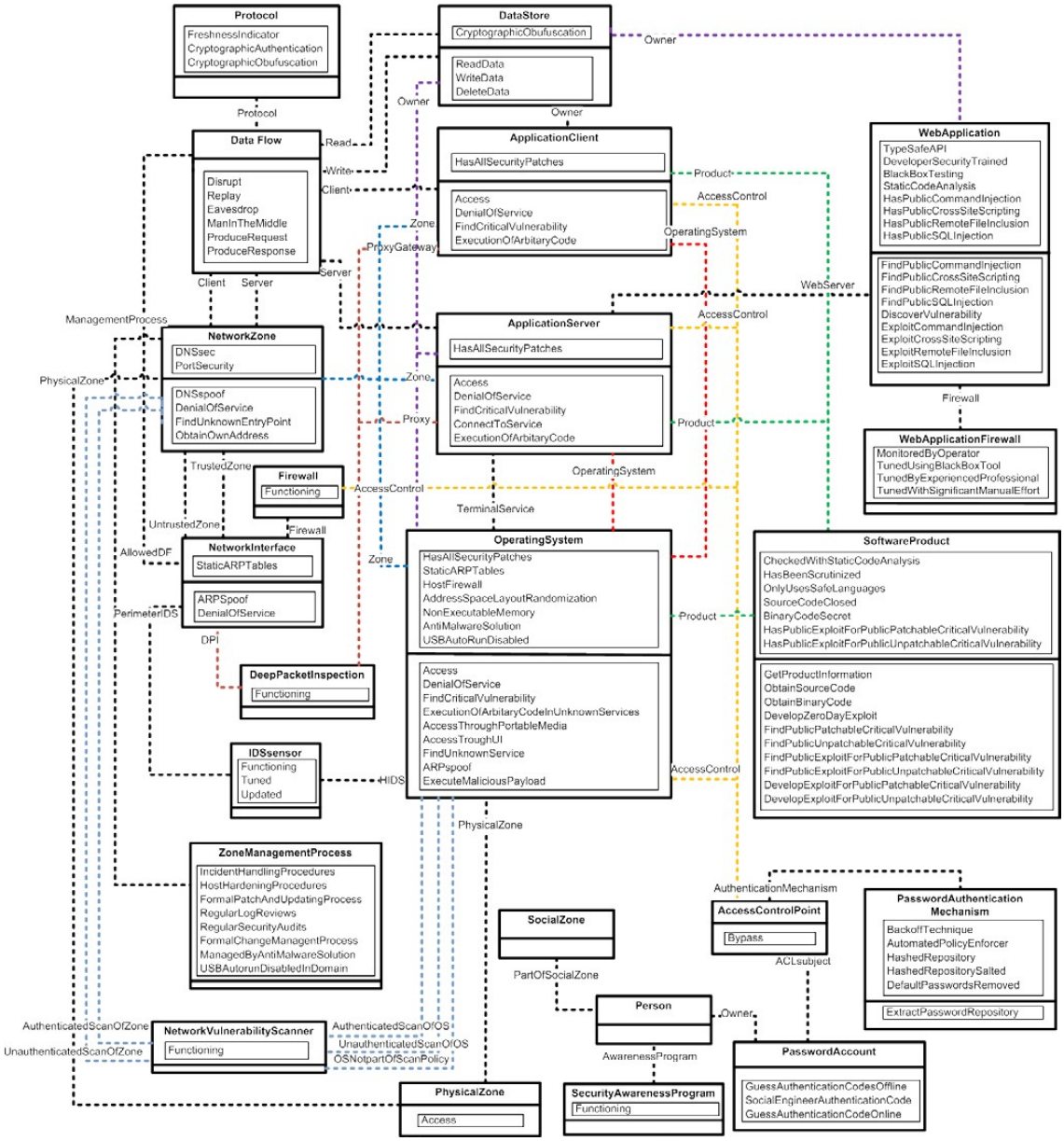

Figure 17: Overview of CySeMoL's structure

Software engine for probabilistic inference

Finally, in order to automate the process of cyber security evaluation, which is done through the calculations of the CySeMoL metamodel, an underlying inference engine, which implements the P^{2}AMF formalism or an equivalent of it, is necessary. At the present day, only one such solution is fully available for the work in the project: the Enterprise Architecture Analysis Tool (EAAT) [22], [23], previously developed at KTH, ICS. While the solution is heading toward the discontinuation of its software development lifecycle, two more related and applicable solutions are on their rise, one of them its successor in form of an upcoming commercial product, the other an inspired research prototype.

The reference models (templates) developed in the project can be applied to whichever of the three above mentioned solutions.

1.5.4.5 Collection of reference models (templates)

The collection of reference models (templates) for modeling and evaluating smart grid architectures in CySeMoL to be developed contains the following:

1. Macro-level reference models. This group aggregates reference models covering IT/OT landscapes or environments consisting of a group of systems, often distributed, as opposed to individual systems that usually are placed in one physical box.

    a. SCADA. Covers systems related to the SCADA infrastructure of a typical DSO, including inter control center interoperation systems and maintenance personnel workstations.

    b. DSO substation. Covers systems placed in the DSO substations. The reference model is partially overlapping with the DSO SCADA reference model, or seen differently, the latter uses (includes) the DSO substation reference model.

    c. DSO AMI (advanced metering infrastructure, i.e., smart metering). Covers systems constituting the infrastructure of AMI at the DSO level (from backend servers through network infrastructure to smart meters), as well as a number of supportive information systems related to it.

    d. WAMPAC. Covers the infrastructure for wide area monitoring, protection and control, which is today mostly used on the level of TSOs, although some of the concepts may be applicable at the level of DSOs, as well.

    e. Aggregator. Covers assumed IT architecture of an organization, which aggregates distributed energy resources, and provides services such as flexibility management. The systems included herein mostly fall into the category of IT.

    f. Market system. Covers assumed IT architecture of market systems and clearinghouses.

    g. Smart building. Covers the systems environment of a smart building, be it a smart home, other building, or industry.

    h. Distributed energy resource. Covers the most immediate control infrastructure of typical distributed energy resources (e.g., photovoltaic panels, wind turbines, batteries).

    i. Electric vehicle infrastructure. Covers the infrastructure of an entire electric vehicle system, which covers the electric vehicle, electric vehicle supply equipment (EVSE), the infrastructure of EVSE operators, electric mobility service providers up to a clearinghouse.

    j. Enterprise and office IT environments. Covers a few environments typical and hence assumable for office and enterprise IT (e.g., office clerk workstations, typical company IT systems such as e-mail system, web server, etc., common industrial packages such as SAP, etc.).

2. Micro-level reference models. This group aggregates reference models that describe different individual systems, which often operate in a single physical box (e.g., a RTU or a server computer).

    a. Substation automation systems. Covers individual systems that can be found in a substation, such as different RTUs, IEDs and PLCs.

    b. Common operating systems. Covers individual operating systems that have a number of services and other software pre-installed. Examples are maintenance workstations or laptops, different variants of server systems, etc.

An overview of the entire IT/OT landscape which the above listed reference models are a part of is provided in Figure 5 (section 1.4.4, above).

Content of a reference model

Figure 18 illustrates the content of a reference model, in a format suitable for human consumption rather than for processing by software. The illustration describes the SCADA server of the DSO SCADA reference model. It does not include interconnections with its environment (i.e., networks, other servers through data communications, etc.). Additional textual comments have been omitted for clarity.

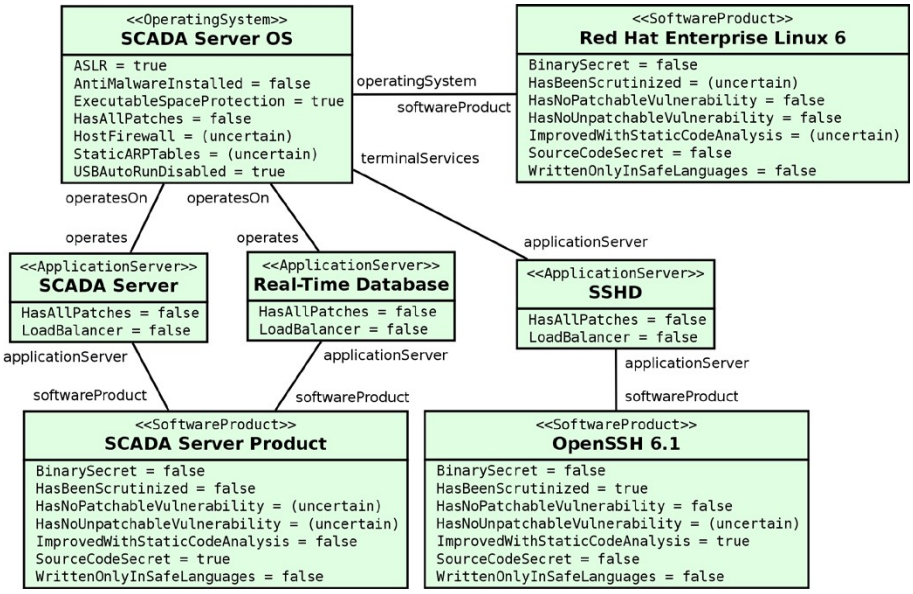

Figure 18: Illustration of a piece of a reference model

1.5.4.6 Prioritization of reference models (templates) to be modeled

Different parts of IT/OT architectures have different properties with respect to e.g. size/extent, age/maturity, commonality/degree of adoption, volatility over time, etc. Eliciting data to build a reference model can be demanding in different ways, much dependent on the previously mentioned properties of the piece of architecture under consideration. In some cases, there is much detail available, as well as many variants of implementations, which tends to increase the effort needed in the process of information elicitation. Such is the case for SCADA systems, for instance.

On the other side of the spectrum, there are rather novel infrastructures such as the automation of electrical markets for smart grids, aggregator architectures, or systems for electric vehicle infrastructure. Some of these or parts of these have not yet reached the point of maturity of development and adoption that would allow a reasonable consideration of typical implementations or typical architectures. Hence, such infrastructures are more demanding to obtain valid and reliable information and assumptions about, which implies both less detail reasonable to capture for their reference models, as well as a higher degree of speculative nature of these reference models (which may diverge heavily from future reality). The demands of eliciting information and formulating reference models together with the above mentioned motivates the prioritization of reference models.

Criteria of prioritization and their importance

In an attempt to perform a systematic prioritization of the candidates for reference models that are seen relevant to capture, we identified a set of prioritization criteria. However, rather than being exact metrics, all the criteria have the nature of estimates.

The prioritization criteria are as follows, in the format "acronym: description and comments":

1. NEC [NECessity]: The necessity/importance of the piece of architecture under consideration in the overall smart grid ICT architecture and ultimately the smart grid

as a functional whole. For example, the presence of a SCADA system is arguably more necessary and important in smart grid architectures than the presence of an electric vehicle infrastructure (which is more optional with regards to making the smart grid architecture function). NEC has a high importance to the prioritization.

2. KNW [KNoWledge]: The relative amount of detailed knowledge available or obtainable about the piece of architecture under consideration within the project. For example, we have more detailed knowledge about the structure and function of a SCADA system than about a market automation system. KNW has a medium/high importance to the prioritization, since it highly impacts the project team's capability of arriving at a satisfactory reference model.

3. SIZ [SIZe]: The absolute size of the piece of architecture under consideration in terms of the amount of elements to model. For example, the DSO AMI contains considerably higher amount of ICT elements than a distributed energy resource. SIZ has a low/medium importance to the prioritization (although the bigger the model, the more valuable its availability to whoever needs to model that piece of infrastructure; the importance of that parameter is limited in this project).

4. COM [COMmonality]: The assumed commonality of the presence of the piece of architecture under consideration in a real smart grid setup. For example, substation automation systems are extremely common in any smart grid architecture, as opposed to WAMPAC, which mostly exists on transmission level and by far not all transmission environments. COM has a high importance to the prioritization, since there is a clear need for reference models for what is used most and thus has the outlook to be modeled most.

5. TS [Temporal Stability]: The temporal stability (i.e., the opposite of the volatility over time) of the piece of architecture under consideration. For example, SCADA is a very mature piece of architecture, which changes at a minimal pace over time, unlike the architecture of a market system or a clearinghouse, which is presently young and subject to much shaping and development. TS has a high importance to the prioritization, since the value of the reference models deteriorates at a pace directly related to the amount of their volatility.

6. NTM [Need-To-Model]: The need/importance of modeling the piece of architecture under consideration with regards to the project objectives. For example, distributed energy resources need to be modeled within the project (w.r.t. its objectives), as opposed to WAMPAC, which is at the periphery of interest to the project. NTM has a high importance to the prioritization.

7. POS [POSsibility]: The possibility of arriving at a satisfactory reference model of the piece of architecture under consideration; For example, it is relatively unhindered to arrive at a model of a substation or a SCADA system, while the little public knowledge and legacy of aggregator systems architectures prevents one from arriving at a model that could reliably be used as a reference. POS has a high importance to the prioritization.

8. EAS [EASe]: The ease of formulating a reference model of the piece of architecture under consideration. For example, it is easy to model a simple architecture, such as the ICT part of a distributed energy resource, as opposed to a complex one such as DSO AMI or WAMPAC. EAS has a low importance to the prioritization.

9. SRO [Significance of ROle]: Significance of the role that the piece of architecture under consideration plays in the cyber-physical scenarios studied within the project. For example, WAMPAC, which is at the transmission level, relates little to the cyber-physical aspects studied in this project, unlike DSO substation, which is very immediate to the electrical process. SRO has a medium importance to the prioritization, since cyber security aspects are intertwined and full of indirection,

which makes the closeness to a physical process of a piece of architecture not as decisive as the cyber aspects.

10. SIM [Significance of IMpact]: Significance of the impact of cyber-physical incidents that can materialize through the piece of architecture under consideration. For example, enterprise IT systems and office systems tend to stand for little direct exposure to cyber-physical threats (unlike cyber-security threats in general though), as opposed to DSO SCADA, the compromise of which can cause much distortion to the electrical process and even physical harm; SIM has medium importance to the prioritization.

11. CPD [Cyber-Physical Dependencies]: Possibilities for inclusion/emphasis of cyber-physical dependencies rather than cyber-security dependencies alone. For example, it is possible to include and/or emphasize cyber-physical relations and dependencies in industrial control systems (e.g., DSO SCADA, substation automation systems) much better than in operating systems in general, or common enterprise systems. CPD has a low importance to the prioritization, due to its nice-to-have nature (as opposed to the nature of a necessity), as well as the overall alignment to the objectives of the project.

For all of the criteria identified, the higher value a reference model candidate scores, the generally more desirable it is to create and formulate the reference model, within the project.

Evaluation

The results of the evaluation of the candidates for reference models (templates) on the criteria listed earlier are presented in figure 19. Each criterion is evaluated on an ordinal scale {L < L/M < M < M/H < H}, in which the letters stand for low, medium and high, respectively. Quantified, the scale translates to $score_{criterion} \in \{0 < 0.5 < 1 < 1.5 < 2\}$, from which the total score is calculated, together with the importance of each single criterion, which uses the same scale as above, and which translates to $weight_{criterion} \in \{1 < 1.5 < 2 < 2.5 < 3\}$. Finally, the numerical score of each reference model is calculated as follows:

$$\forall\, i \in \text{ReferenceModels: } score_i = \sum_{j \in criteria} weight_j \cdot score_{j_i}$$

Reference model	NEC (H)	KNW (M/H)	SIZ (L/M)	COM (H)	TS (H)	NTM (H)	POS (H)	EAS (L)	SRO (M)	SIM (M)	CPD (L)	Score
DSO SCADA	H	H	H	H	H	H	H	L/M	H	H	H	**48.5**
DSO Substation	H	H	M	H	H	H	H	H	H	M/H	H	**47.5**
DSO AMI	M	M/H	H	M	H	H	H	L/M	H	M	M/H	**38.75**
WAMPAC	L	M	H	L	M	L	M	L/M	L	H	H	**18**
Aggregator	L/M	L	M	L/M	L	M/H	L/M	M/H	H	M	M	**19**
Market system	L/M	L	L	M	L	M/H	L/M	M	M	L	L/M	**14**
Smart building	L	M	L	M	M	M/H	M	M	M/H	L/M	M/H	**22.5**
Distributed energy resource	L	M	L	M	H	H	M/H	M/H	H	M	H	**31.5**
Electrical vehicle infrastructure	L	M	M	M	M	M	M/H	M	L/M	L/M	M/H	**22**
Enterprise and office IT environments	H	H	M/H	H	H	H	H	M	M	L	L/M	**40.75**
Substation automation systems	H	M	M	H	H	H	H	M	H	M/H	M	**43**
Common operating systems	H	M/H	M/H	H	H	H	H	M	M	M	L	**41**

Figure 19: Evaluation of criteria for prioritization

Prioritization

The prioritization builds on the scores presented in figure 19, and is shown in figure 20, ordered according to its prioritization score.

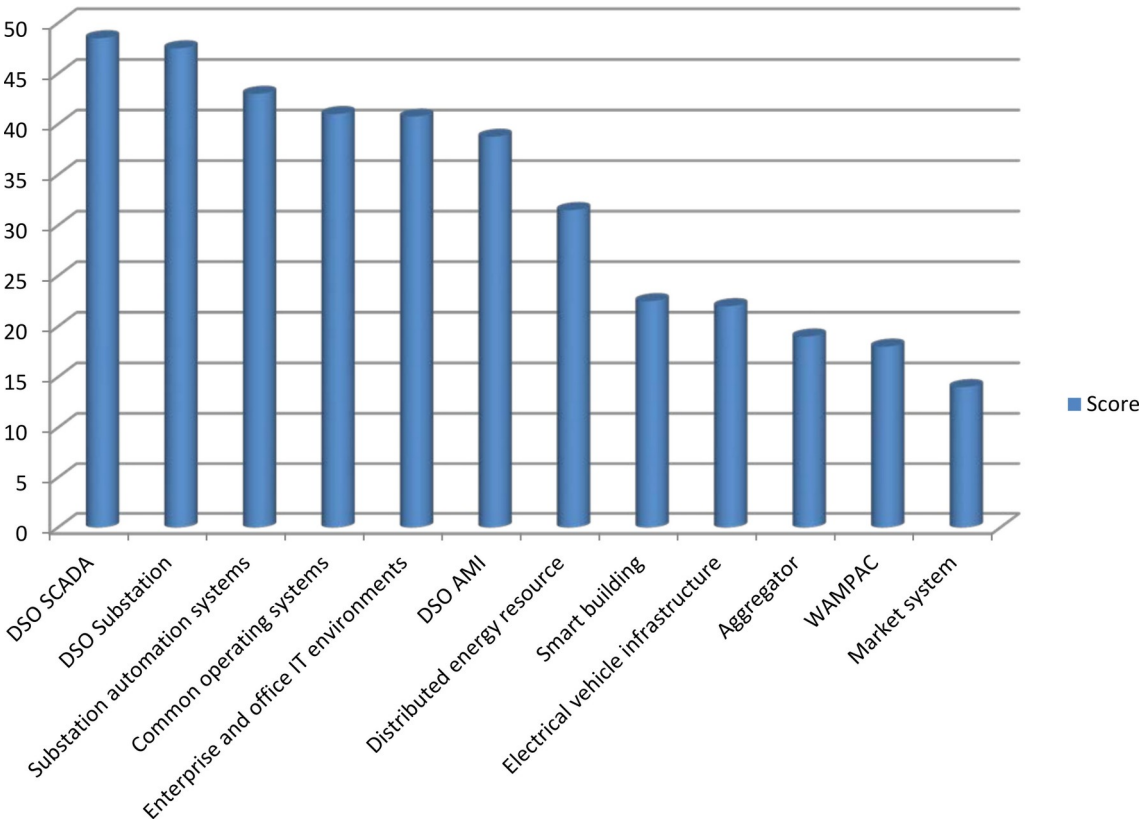

Figure 20: Prioritization of reference model candidates

According to the prioritization criteria, their weights, and the estimations done in the prioritization process, the traditional architectures such as SCADA, substation automation systems, common operating systems, enterprise IT tend to be the most desirable ones to model, together with the rather extensive AMI. DER falls between this cluster and the one including the rest of reference model candidates, which render less desirable to model. Although the ICT architectures of smart buildings, electric vehicle infrastructure, DER aggregators and market systems are also interesting in context of the project, the prerequisites for formulating quality reference models for them are rather low, and hence their priority is low compared to the others.

The chosen priority list of reference models to formulate is the following, including a brief motivation for each:

1. DSO SCADA. SCADA system seems to be present in any smart grid architecture, as well as it seems to be the single most cyber-dangerous component when taken control over by an antagonist with a destructive intent.

2. DSO substation. Similarly to SCADA although at a less aggregated level, a substation is where the electrical process is most immediately observed and controlled at the level of DSO.

3. Substation automation systems. This reference model is a micro-level one according to the categorization further above, based on which it can be seen as a support reference model for DSO substations.

4. Common operating systems. This reference model is also a micro-level one, and can be seen as a support reference model for all other reference models listed here.

5. Enterprise and office IT environments. This reference model, although not directly and immediately related to smart grids, is inevitably a part of each modern organization (company servers, office workstations, maintenance computers, etc.). Beyond the general case of an organization, each utility, including the electrical ones, is here assumed to have an implementation of the well-known and de-facto standard enterprise resource planning system by SAP. Moreover, since cyber-attacks and infections often spread themselves through corporate IT environments, it is most often of importance to also model this part in context of smart grids, together with the parts that are closer to the electrical process.

6. DSO AMI. Although smart metering might mostly be sensitive in other means than the traditional observability and control of the electrical process, it is one of its potential uses on a sub-substation level, which also can be advantageous to observe for the purposes of control and optimization. Moreover, this project will study certain scenarios related to this use of AMI. Similarly, beyond the scenarios of normal operation, functions that AMI offers can be misused in a compromising fashion (e.g., denial of service or electric supply), which will also be a subject of study in the project.

7. Distributed energy resource. Although typically a very simple piece of ICT architecture, DERs are a notable part of smart grids. Moreover, compromising multiple DERs in a welltimed fashion might have consequences to the stability or operation of larger parts of the smart grid, which will be studied in the project.

These models have been developed, formulated and modeled; and are described below in this document. The information necessary for the development and formulation of the reference models have been gathered from documentation available in the public domain, as well as interviews with experts on the subject matter.

1.5.4.7 Reference model: SCADA infrastructure

SCADA systems play an important role in the infrastructure of a power utility. The SCADA reference model described here aims to represent a generalised IT architecture of a real-time process control centre solution for the electrical grid. Presented is a geographically distributed system that is controlled from a central location. The model depicts an installation of a SCADA system in a typical industry setting and includes backup services and common data exchanges between system elements. The reference model shown here is designed with redundancy, but not multiplicity. By redundancy is meant that if it is likely that there are two servers providing a service in a fail-safe manner, they also exist in the model. By multiplicity is meant that if there is no fail-safe reason to have more than one element, they are not included. No network redundancy is modelled.

General description

A typical SCADA setup consists of a master station, Remote terminal units (RTUs) and a communication system [24]. A master station (SCADA server) is responsible for communicating with the geographically dispersed field equipment. Data from the field is acquired through the system part called Front End (FE). Human users operate the SCADA system through human machine interface (HMI). There might be several consoles for users. Operator consoles are used by system operators for controlling the supervised process. Engineering consoles (alternative name: maintenance consoles) are used for control system maintenance; database and picture maintenance.

RTUs are the sources of measurements and states for a SCADA system. Real-time system measurements are continually overwritten in a SCADA system and that is why old values need to be stored separately. Old values are stored in a separate historical database (Historian) for future use [25], [26]

Zones

Figure 21 describes a SCADA system setup with a simple substation. There are altogether seven different network zones shown. The reference model covers five zones. These zones and the scope of the modelling are described below. The network topology as modelled in securiCAD is shown in Figure 22. The reference model contains 50 views, but here we will show the most important ones. Data flows and attributes have been omitted for readability reasons.

- The SCADA LAN is the central part of the SCADA model where the main services are run and data is processed. SCADA LAN is part of the reference model. The SecuriCAD depiction of the SCADA LAN is shown in figure 23.

- Process LAN. This is the gateway to the process WAN where the measurements are collected from and new configurations saved to. The process LAN is part of the reference model. The SecuriCAD depiction of the data acquisition is shown in figure 24.

- The SCADA demilitarized zone (DMZ) is a duplicate of the main SCADA system that is accessible from the organisation`s office network. The purpose of this zone is making the information collected and processed by SCADA and application servers accessible in the office environment without exposing the control system directly to security threats and attacks. The SCADA DMZ is part of the reference model. The SecuriCAD depiction of the DMZ and Office LAN is shown in Figure 25.

- Office LAN. An important part of any SCADA configuration is the interconnection to the office network where customer information is processed and workflows planned among other things. Office environments differ significantly, so most of it is outside the reference model's scope. Only one intranet workstation and one office application server are part of the reference model.

- Maintenance (engineering) LAN. This is where the substation system maintenance is done from. Substation maintenance has not been fully modelled in the SCADA reference model.

- Substations. Geographically dispersed substations with varying configurations. Only a single connection to a simple substation is modelled. The internals of the substation are modelled separately and presented in Section 1.5.4.8. Figure 24 contains a representation of a simple substation.

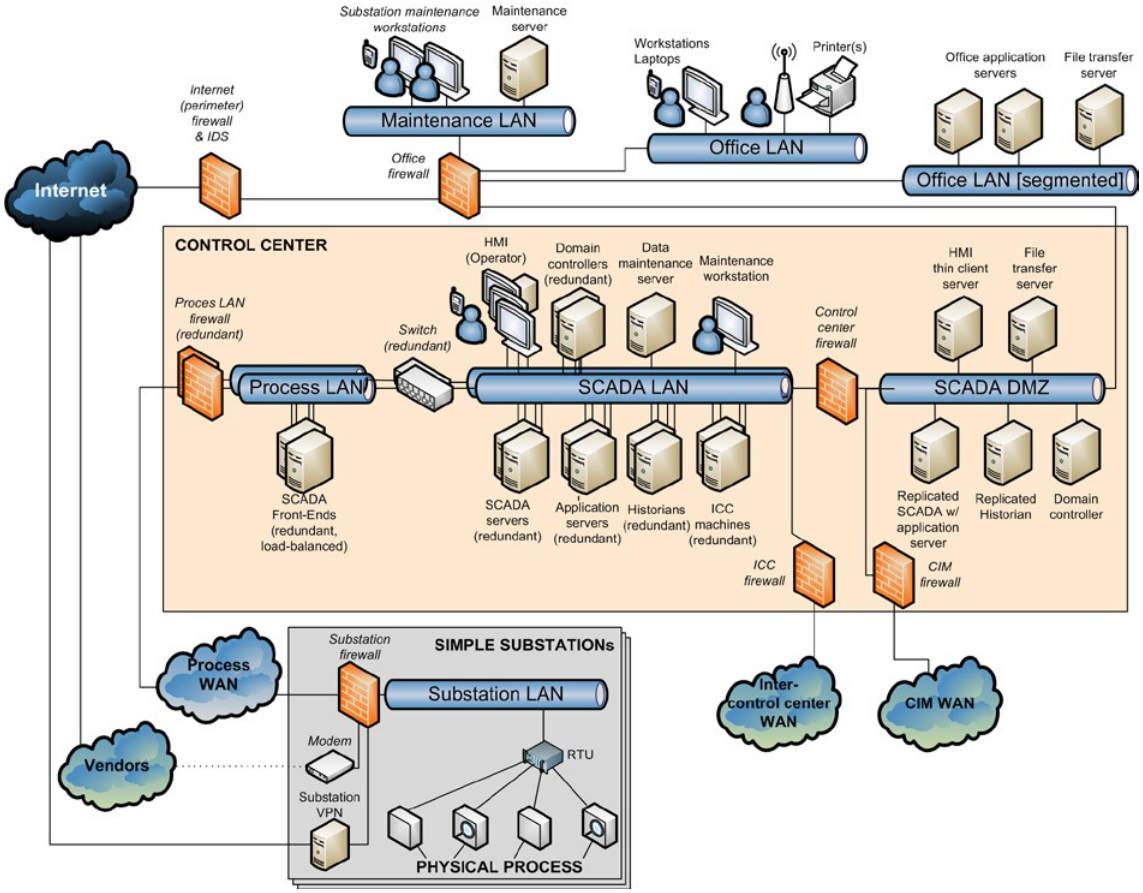

Figure 21: Overview of the SCADA setup

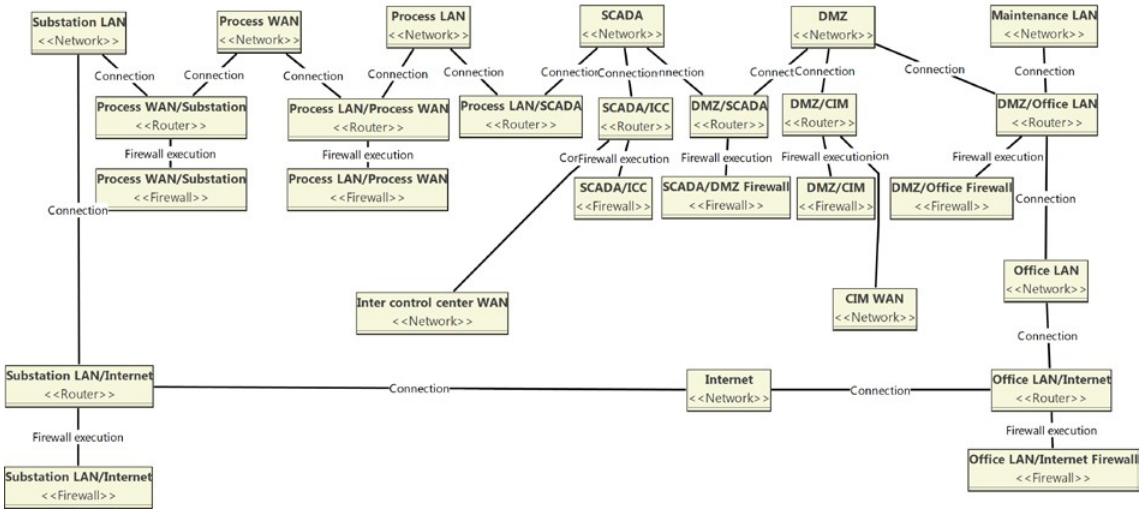

Figure 22: Network topology of the SCADA reference model

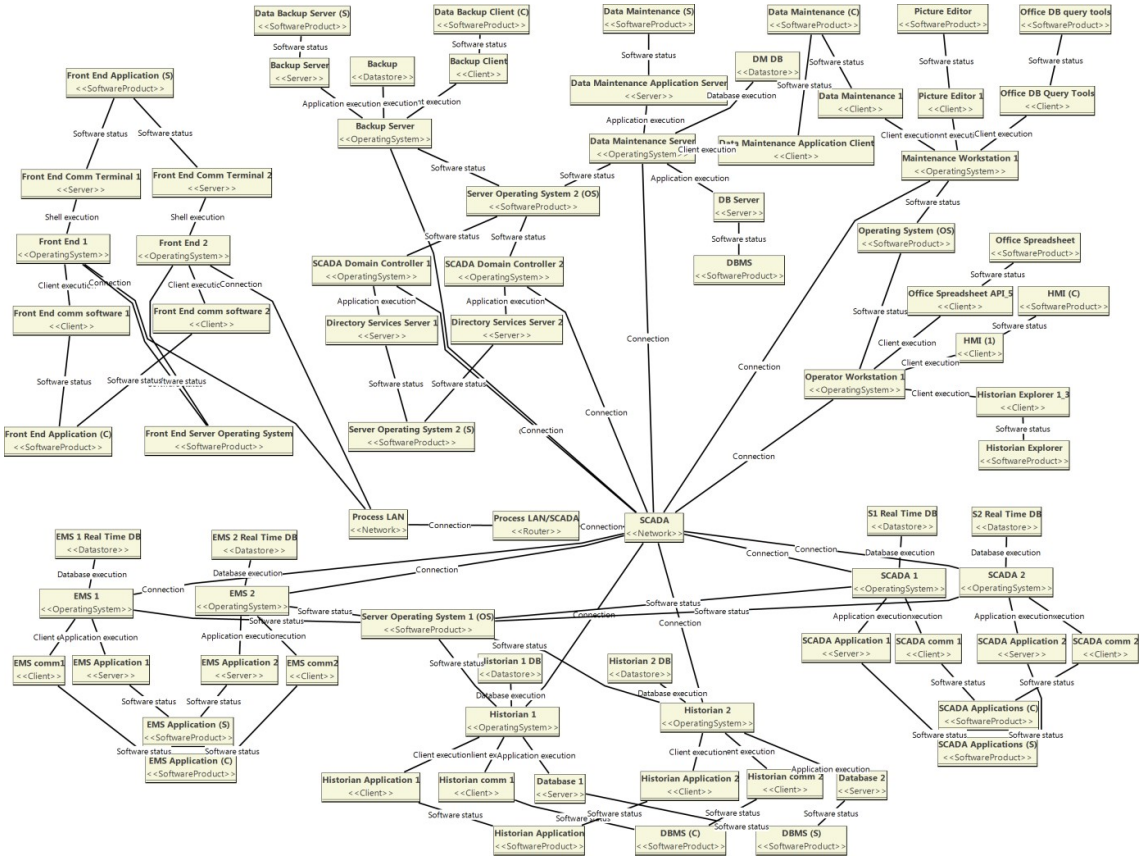

Figure 23: SCADA LAN and connected elements

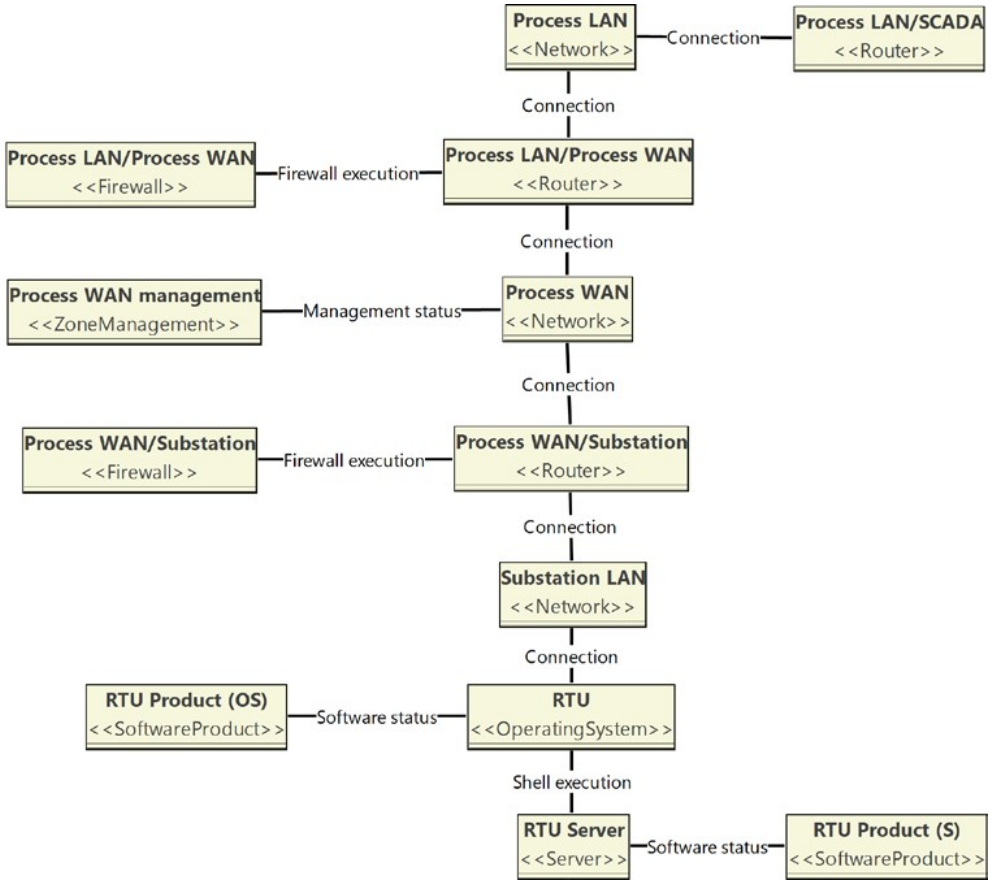

Figure 24: Data Acquisition

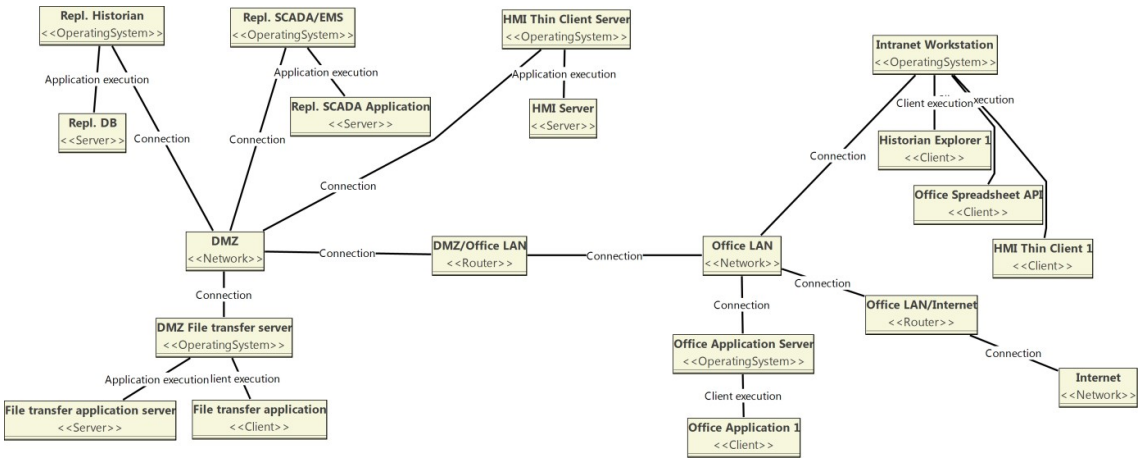

Figure 25: SCADA DMZ and Office LAN

Reference model elements

The SCADA reference model contains several types of elements of which system and data exchange elements as the core parts of the model are described below.

The following network zones are separated using routers and firewalls.

SCADA LAN

- SCADA servers in a redundant setup. A computerised control system that includes functions for remote real time data acquisition and remote control of process devices.
- Application servers in a redundant setup. Electric grid specific tools for monitoring, control, and performance optimisation of the grid.
- Historians in a redundant setup. Database that contains historical process data that is used for future planning and incident handling.
- Data maintenance server. Enables manual data entry into the SCADA topology database and is used for defining the different data that are being stored, exchanged and processed across the entire SCADA system.
- HMI. Human machine interface for communication between the operator and the machine.
- Domain controllers in a redundant setup. User authentication and authorisation management in the SCADA LAN.
- Engineering workstation. For control system maintenance, database and picture maintenance.
- ICC machines in a redundant setup. For communication to other control centres using the ICC protocol.

Process LAN

- SCADA Front End servers in a redundant setup. To manage communication with geographically distributed field devices.

SCADA DMZ

- Replicated SCADA and application server. A copy of the control network SCADA server for access from the office network. The server is also running a copy of the control network application server for access from the office network.

- Replicated Historian. A copy of the Historian from the control network for access from the office network.

- HMI thin client server. A terminal services server that mediates access between the DMZ and the office network.

- Domain controller. Domain controller with a separate user database for the DMZ LAN.

- File transfer server. The file server is used to by SCADA and EMS systems in the SCADA LAN to fetch data from outside the SCADA LAN.

Office LAN

- Intranet workstation. Workstations for users in the office network.

- Office application server(s). Systems that are used for managing enterprise services using the data from the SCADA DMZ LAN.

Data exchanges

The reference model captures likely data exchanges between the different zones and systems. These data exchanges are called data flows in the model. The reference model contains three categories of flows. The first category describes data flows which are critical for the functioning of the control system. In the second category there are data flows which are likely, but not necessary for the functioning of the control system. Backup flows make up the third category, showing which servers are likely to be backed up and how. All the data flows here have been organised according to the network zones where they exist. Figure 26 shows some of the data flows inside the SCADA LAN.

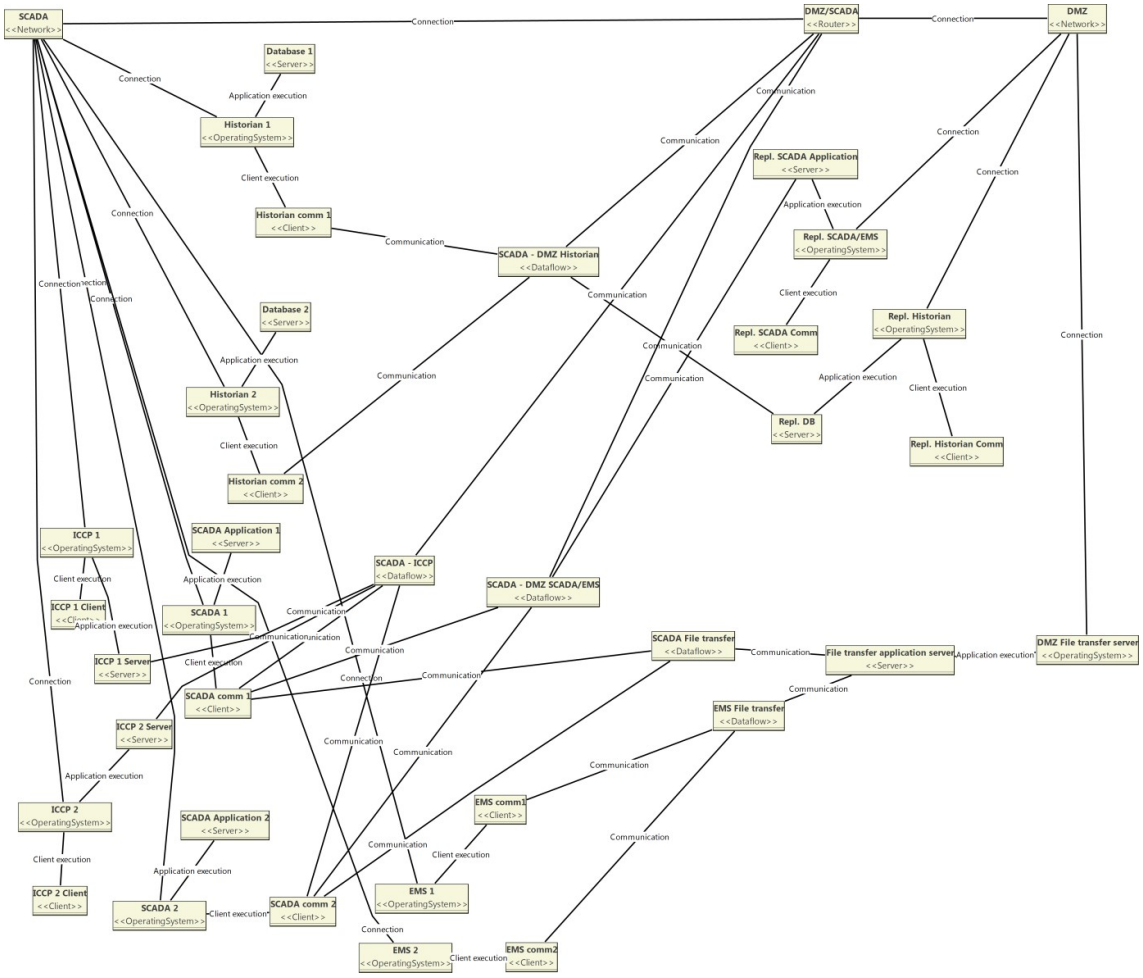

Figure 26: SCADA LAN data flows

Control system critical data flows:

1. Data flows within SCADA LAN between servers:
   a. SCADA-Historian data flow: SCADA server to Historian server
   b. Historian replication: Between two Historian servers
   c. SCADA replication: Between two SCADA servers
   d. EMS replication: Between two EMS servers
   e. EMS-SCADA: EMS server to SCADA server
   f. SCADA-EMS: SCADA server to EMS server
   g. Historian Data Maintenance: Data Maintenance application client to Historian
   h. SCADA-Data Maintenance: SCADA server to Data Maintenance application server
   i. Data Maintenance-Historian: Historian to Data Maintenance application server
   j. Data Maintenance-SCADA: Data Maintenance server to SCADA server
   k. Authentication through redundant domain controllers
   l. ICCP servers to SCADA server

2. Data flows within SCADA LAN between servers and operators:
   a. SCADA-HMI data flow: Operator workstation to SCADA server
   b. Picture Editor-Data Maintenance: Engineer Workstation to Data Maintenance application server
   c. Historian Explorer-SCADA: Operator workstation to Historian servers
   d. Spreadsheet-SCADA: Engineer Workstation to SCADA server

3. Data acquisition flows between SCADA LAN and Process LAN:
   a. SCADA-Front End: SCADA client to Front End server
   b. Front End-RTU: Front End to RTU

4. Data flows between SCADA LAN and SCADA DMZ:

a. Historian-DMZ Replicated Historian: Historian client to Replicated Historian server
b. SCADA-DMZ Replicated SCADA: SCADA client to Replicated SCADA/EMS server
c. EMS-SCADA: EMS client to Replicated SCADA/EMS server
d. SCADA-ICCP: SCADA application client to ICCP server
e. SCADA file transfer: File transfer from File transfer application from DMZ
f. EMS file transfer: File transfer from File transfer application from DMZ

5. Data flows within SCADA demilitarized zone between servers:
   a. SCADA-Historian data flow: SCADA server to Historian server
   b. HMI Thin client server to SCADA server
   c. Authentication through domain controller

6. SCADA zone and inter control centre network:
   a. ICCP - Inter control centre WAN: Inter control centre WAN to ICCP servers

7. SCADA demilitarized zone and office network:
   a. Workstation HMI: Intranet workstation HMI client to HMI Thin client server

8. 8. SCADA zone and office network:
   a. Office-SCADA: Office application server client to SCADA server
   b. Office-EMS: Office application server client to EMS server

Non critical data flows for system management and administration:

The following three services may be enabled on any server and workstation located in the SCADA and SCADA demilitarized zones:

- Secure file transfer (SFTP),
- Terminal services between clients (RDP),
- Secure shell (SSH).

Backup data flows:

The following data flows originate from a central backup server that resides in the SCADA LAN and backs up systems there:

- Backup Retrieve: Getting data from Data Maintenance Server, Historian, SCADA, EMS, Operator Workstation, Engineer Workstation, Front End.
- Backup send: Sending data to Data Maintenance Server, Historian, SCADA, EMS, Operator Workstation, Engineer Workstation, Front End.

Security assumptions

The following assumptions have been gathered with interviews with two SCADA experts and with the help of literature [24], [25], [26]. These assumptions describe the security properties of the model elements and are set as default values in the reference model.

1. Firewalls: Firewalls are used to protect against intrusions. In most cases the firewall rulesets are well known. However, not all the traffic between the demilitarized zone and the office environment has been documented, so the ruleset between these zones is known with the likelihood of 80%.

2. Network: It is assumed that no Domain Name System Security Extensions (DNSSEC) and no port security (restrict a port's ingress traffic by limiting the MAC addresses) in switches and routers are being used throughout the network.

3. Operating systems: The utility uses well-known commercial-off-the-shelf operating systems like Red Hat Enterprise Linux and Microsoft Windows, which have protections against buffer overflow attacks and with 50% probability host firewalls. Antivirus and antimalware measures have only been installed in the office environment. There is an

active vulnerability management program for the software used in the office environment, but the software in the control environment (Process LAN, SCADA LAN, SCADA DMZ) is updated only occasionally for service stability reasons. The operating system software does have 75% likelihood of vendor support, because old unsupported versions might still be used. The office software codebase has been most likely well tested and there might be active bug reporting going on.

4. Client and server software: There are different programs used from different vendors to run the SCADA system. The software used in the Office environment has been most probably patched while the software in other zones has been patched with 50% likelihood. There is only small likelihood (10%) that the software in substations has been patched. The software used has very likely vendor support. However the codebase of SCADA system specific software might only have been partly statically and dynamically tested during development.

1.5.4.8 Reference model: Substation automation infrastructure

Substation automation consists of a number of different components such as substation-level control systems, substation engineering systems, IEDs (relays) for protection and control, merging units, engineering and testing systems. Firstly we describe a reference model for a substation automation system, i.e., a generic overview of a typical substation setup that is vendor-independent. Secondly, a securiCAD model of the generic substation automation system is described.

As a note, this reference model is attempting to describe a fully automated IEC 61850 substation, to date typically found in the high voltage and/or complex power transmission domain. Many substations use simpler systems with less automation, redundancy and technological sophistication, e.g., on the level of sub-transmission and distribution of electrical power.

General description

An overview of an IEC 61850 substation automation system (consisting of a number of networks and devices) is presented in figure 27. Detailed descriptions of its network structure, systems and data flows follow below.

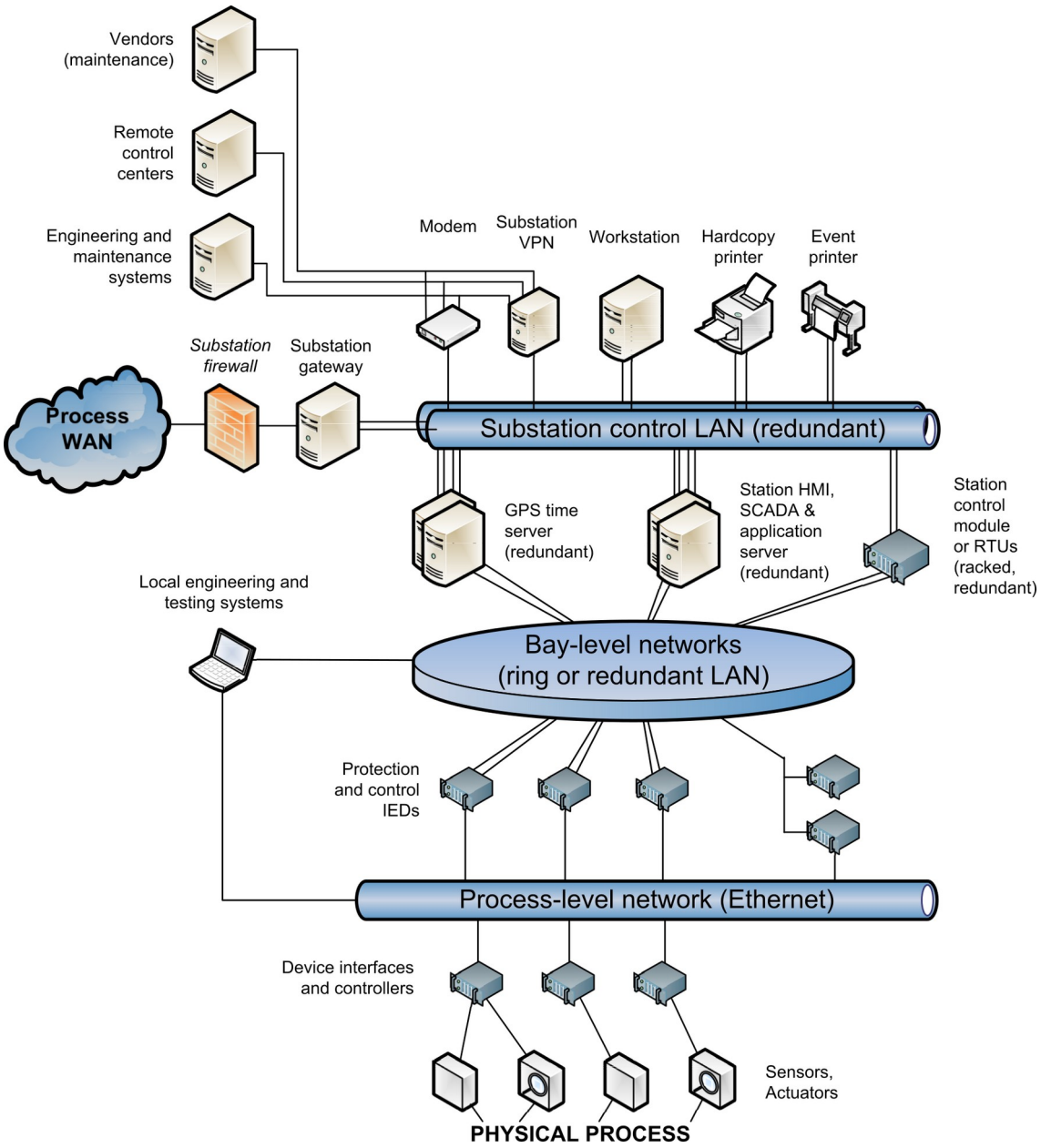

Figure 27: Overview of the reference model of a substation control system

Networks

There are three types of networks that can typically be found inside an IEC 61850 substation:

1. Substation control network, which mostly hosts network appliances/systems, time synchronization systems (e.g., GPS time server), printers and alarm systems, IT and OT systems running on general-purpose computing hardware (e.g., PC) such as a station-level control system (essentially a small SCADA system). Some of these systems are co-hosted on the bay-level network described below. Due to high availability and reliability requirements of certain functions dependent on this network (e.g., alarm and event printing, interconnection with control centre SCADA system), this network is redundant. A commonly assumable topology is a dual LAN using wired Ethernet.

2. Bay-level network, which mostly hosts IEDs for bay-level protection and control purposes, as well as co-hosts certain systems also hosted on the substation control network, namely the GPSbased time server(s) and the station-level control system or device. There can be multiple baylevel networks in a substation, and also can feature redundancy. An example of such redundancy and a common topology chosen for this

network is double ring (decentralized in larger substations). Both optical and copper cables can be used as the physical medium on this level.

3. Process-level network, which hosts merging units and controllers that enable the connection between higher-level protection, supervision and control functions with physical appliances (sensors and actuators). There can be one or more process-level networks per bay-level network. Due to the real-time requirements of data communications on this network, much communication on this network runs directly over Ethernet (as a data link protocol according to the ISO/OSI Basic Reference Model) rather than TCP/IP (transport/network level, respectively). Also, as this network operates in close vicinity of the electrical process that generates strong levels of electro-magnetic interference (i.e., noise that degrades the performance and reliability of electronic communications and equipment) as well in open fields with the risk of lightning strikes. Therefore, it is increasingly more common to use optical media for the physical layer of network communications than copper.

The networks are also shown in figure 28, marked as "NZ_Station...", "NZ_BayLev..." and "NZ_Process..." respectively.

There are at least two other networks external to the substation, which need to be introduced:

1. The process WAN (cf. SCADA reference model, figure 6; and figure 28, "NZ_Process..."), which is interfacing the substation control LAN, and through which the substation is connected to the control centre SCADA system(s), and possibly also the rest of the power utility. The process WAN can be owned by the utility itself, or it can be leased from an external telecommunications operator.

2. An engineering network segment (see figure 28, "NZ_UtilityE...") at the power utility, from which power engineers access and update configurations of power systems (e.g., protection IEDs) in the substation.

3. A network environment at an external vendor (see figure 28, "NZ_Vendor..."), from which a vendor-side maintainer or troubleshooter is connecting, through a modem interface and/or a VPN placed directly in the substation.

Systems

The systems typically present and used in the substation are categorized according to their primary placement in the substation networks.

1. Systems in the substation control network (see figure 29):

    a. Station-level HMI, SCADA and application server. While at the central SCADA level (cf. SCADA reference model) these components operate separately on different physical or virtual machines due to performance and scalability. They are often operating on a single machine in substations, thanks to only a small portion of the entire electrical process, and thus the small amount of devices, monitored and controlled. The different components (HMI, the user interface; SCADA, the real-time database; and application server, the advanced logic for substation-level protection, control, analysis, and optimisation) all operate as software applications or services, although some consisting of several actual pieces of software. All of the components are redundant for more important substations such as the type of substation described in this reference model.

    b. Station control module or RTUs. In case the SCADA supervision and control is not realised directly through the station-level SCADA, there is a separate station control module, or RTUs that interconnect the substation equipment outlined below (e.g., IEDs) with the central SCADA. The RTUs used here are redundant and typically racked. In simpler types of substations, such as those described in the

SCADA reference model (see section 3.1), only RTUs would be used, without station-level SCADA or separate HMI.

c. GPS time server. Since the substation control system and the entire SCADA system requires precise time synchronisation (synchronisation of clocks on the different devices), even more so in case of the presence of WAMS components such as phasor measurement units, a reliable time server is needed. Since substation can be placed in areas with less reliable communication lines, the time server typically obtains time from the GPS system. The time server is also redundant.

d. Workstation. This is an engineering workstation from which it is possible to manipulate with devices in the substation as well as perform other engineering tasks locally in the substation. One of the tasks that can be performed on this level is security configuration, e.g. access control across the different devices and systems in the substation.

e. Substation VPN. The VPN allows secured remote connections from other networks of the power utility, as well as outside it (e.g., vendors). Sometimes, connections to the substation are made through VPN over a public Internet connection rather than VPN over the process WAN, through which control system communications flow.

f. Modem. A modem is often needed to allow the connectivity directly from the public Internet - in case it is not possible or feasible to connect to the substation using the process WAN. This type of connection is still frequently used for maintenance purposes, both by engineers at the power utility and vendors.

g. Event printer. The printer is used to print notable substation events (as a form of physical log).

h. Hardcopy printer. This is a typical office printer, supporting local engineering operations in the substation.

i. Substation gateway and firewall. The gateway and firewall (possibly operating in a single device) interconnect the substation control network with the process WAN. They also guard the perimeter security of the substation from the process WAN, except eventual modem-based entry points such as the one mentioned above.

2. Systems in a bay-level network (see figure 30):

   a. Protection and control IEDs. There is typically a range of different types of protection IEDs, and a bay control IED. These systems are embedded, physically rugged, and designed for high availability and reliability. IEDs communicate with each other (e.g., using IEC 61850-8-1 [27]), depending on their functions and configuration, as well as the devices they monitor and control.

   b. Local engineering and testing systems can be connected at this level and used, although their permanent presence is not assumed. They use proprietary protocols.

3. Systems in a process-level network (see figure 31):

   a. Merging units essentially function as analog-to-digital converters of measurements taken by sensors (e.g. voltage and current transformers) into sampled values (IEC 61850-9-2 [28]), which are further communicated to other systems such as IEDs for control and protection.

b. Controllers (sometimes also called IEDs) allow the control of switchgear such as circuit breakers, disconnectors, on-line tap changers, etc. They also communicate with IEDs for control and protection, typically using IEC 61850-8-1 (GOOSE/GSSE).

4. Networks external to the substation:

   a. Systems from remote control centres. The most notable in this category is a central (or simply higher-level) control system such as the SCADA, which has the authority to monitor the process at the level of the substation, as well as issue control commands and update set points.

   b. Engineering and maintenance workstations from the engineering and maintenance network segment at the power utility. The engineering and maintenance operations are aimed at substations, and so a part of the reference model, although they are physically located outside of substations. The engineering systems enable the analysis, development and testing of protection and control schemes and substation configurations that are further implemented and/or uploaded to individual devices (e.g., protection and control IEDs).

   c. Maintenance workstations from vendor(s) of substation automation devices. As is often the case, vendors require access to substation due to the need of non-trivial maintenance of the systems present in the substation, potentially including the management of capacity and performance; to be able to guarantee a reliable operation of the whole substation control system or its parts.

Data flows

The data flows described below are divided according to networks, for more clarity.

1. Data flows between the process WAN and the substation control network

   a. Data communication between the substation control system and SCADA system(s) in control centre(s). The protocol can be IEC 60870-5 (-101/104) [29], [30] or DNP3. A more legacy setup could also use Modbus over IP. Typically, the industrial systems used support a number of different protocols, including proprietary ones, which are beyond the scope of this reference model.

2. Data flows within the substation control network

   a. Time synchronisation (NTP).

   b. Printing.

   c. Proprietary communications between engineering systems running at the workstation and the substation control system; eventually between the substation control system and other systems placed in the network, such as a substation alarm device.

   d. Modem and VPN connections from outside the substation; from which further connections are initiated, such as remote desktop, remote shell (SSH), eventually control system communication (e.g., IEC 60870-5, DNP3 or ICCP), or proprietary communications between systems of a single vendor (e.g., maintenance systems).

3. Data flows within the bay-level networks

   a. Communication between the substation control system and IEDs for protection and control, typically according to IEC 61850-8-1 (MMS).

b. GOOSE (IEC 61850-8-1) for information exchange between IEDs (for protection and control).

c. Eventual testing and maintenance communication (during test operations), according to proprietary protocols.

d. Time synchronisation (IEEE 1588 or SNTP).

4. Data flows within a process-level network

   a. Sampled values (IEC 61850-9-2) (measurements) flowing from merging units to IEDs for control and protection. This is actually a bidirectional communication.

   b. Control communication between the IEDs and the switchgear (e.g., circuit breaker IEDs), according to IEC 61850-8-1 (GOOSE/GSSE).

   c. Eventual testing and maintenance communication (during test operations), according to proprietary protocols.

5. Data flows from engineering network segment of the utility

   a. Read access and uploading of configuration files (including protection and control schemes) related to the substation and the different devices in it (e.g., protection IEDs). FTP or a proprietary protocol can be used for this purpose.

6. Data flows from an external vendor

   a. Communications for the purposes of maintenance and eventual troubleshooting. For this purpose, both proprietary and standard protocols can be used (e.g., remote desktop, remote shell, FTP, TFTP, proprietary protocols, etc.).

The reference architecture model

As the reference model implemented in securiCAD contains over a dozen different views while it has conceptually been introduced above, this section only presents a subset. Also, the model offers hundreds of parameter specifications (e.g., whether an operating system uses hardware-based data execution prevention) according to the securiCAD metamodel, which this section does not describe about the model.

An overview of all networks together with network equipment, firewalls, intrusion detection systems and information about network management processes, is given in figure 28.

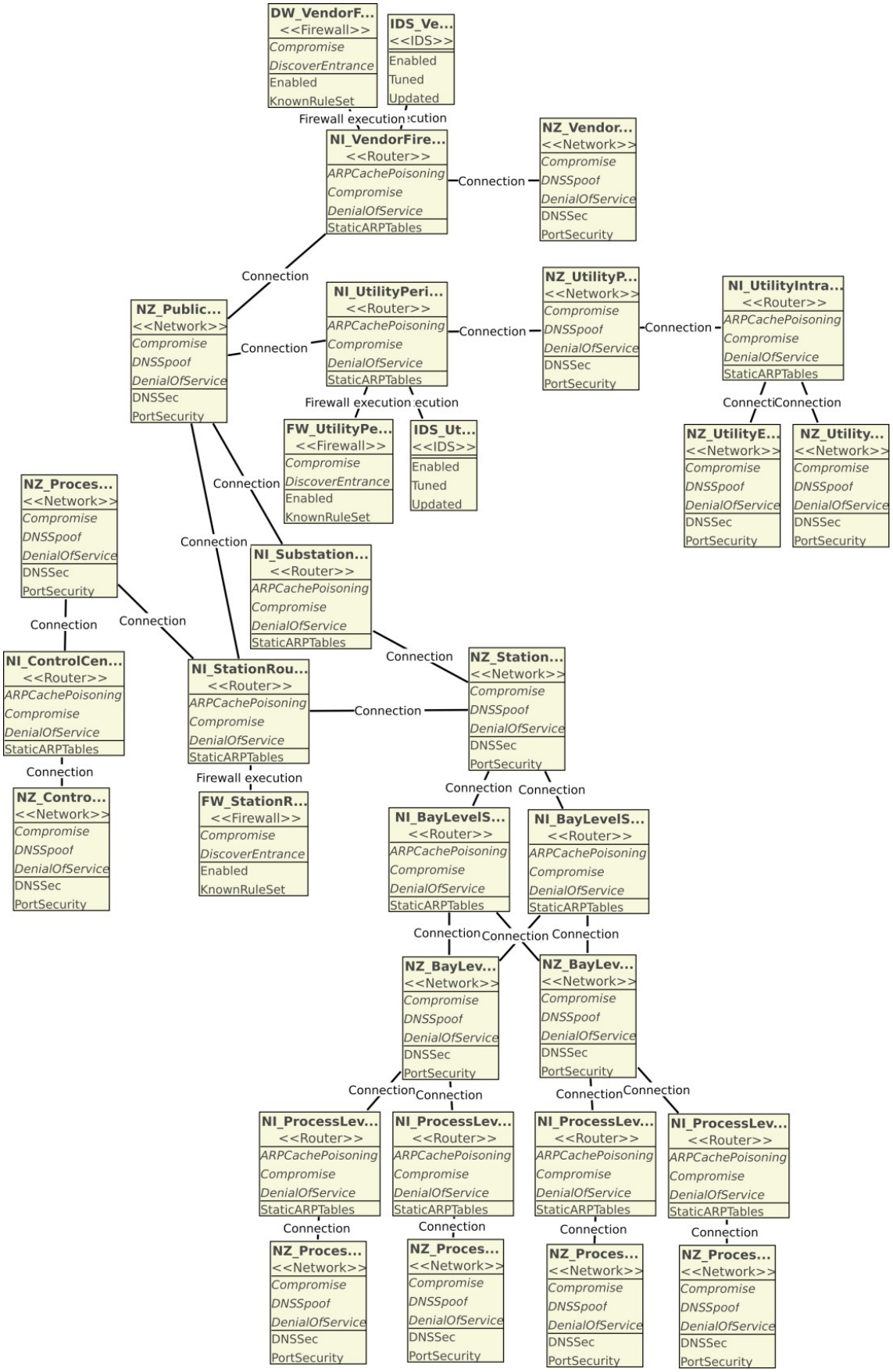

Figure 28: Overview of networks together with network equipment, firewalls, intrusion detection devices, and network management processes.

In the following, securiCAD models relating to the three types of networks in an IEC 61850 substation (i.e., station-level network, bay-level networks and process-level networks) are described. Figure 29 shows the station-level network, figure 30 shows the bay-level networks (specifically two in this instance), and figure 31 shows the process-level networks under the

first bay of the two modelled. As can be seen the different data flows within a single network zone does not need to be modelled for the purposes of security analysis performed by securiCAD. Modelling these for documentation purposes is possible in securiCAD, although it increases the computational demands of the security analysis process, and can even reduce the comprehensibility of the view if a single view is used for capturing all these relations.

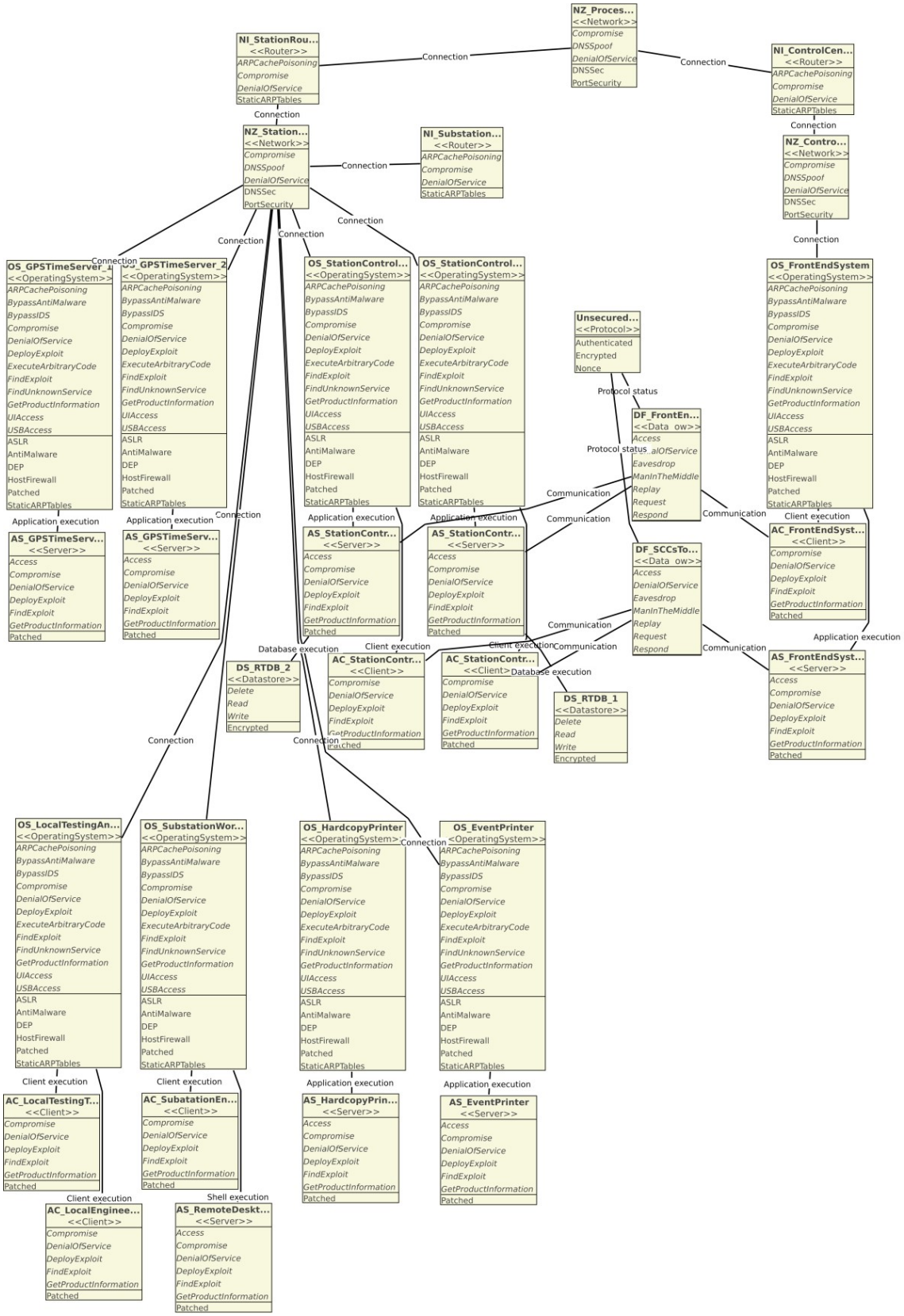

Figure 29: View of the systems, services and a network-crossing data flows related to the station level network in a substation.

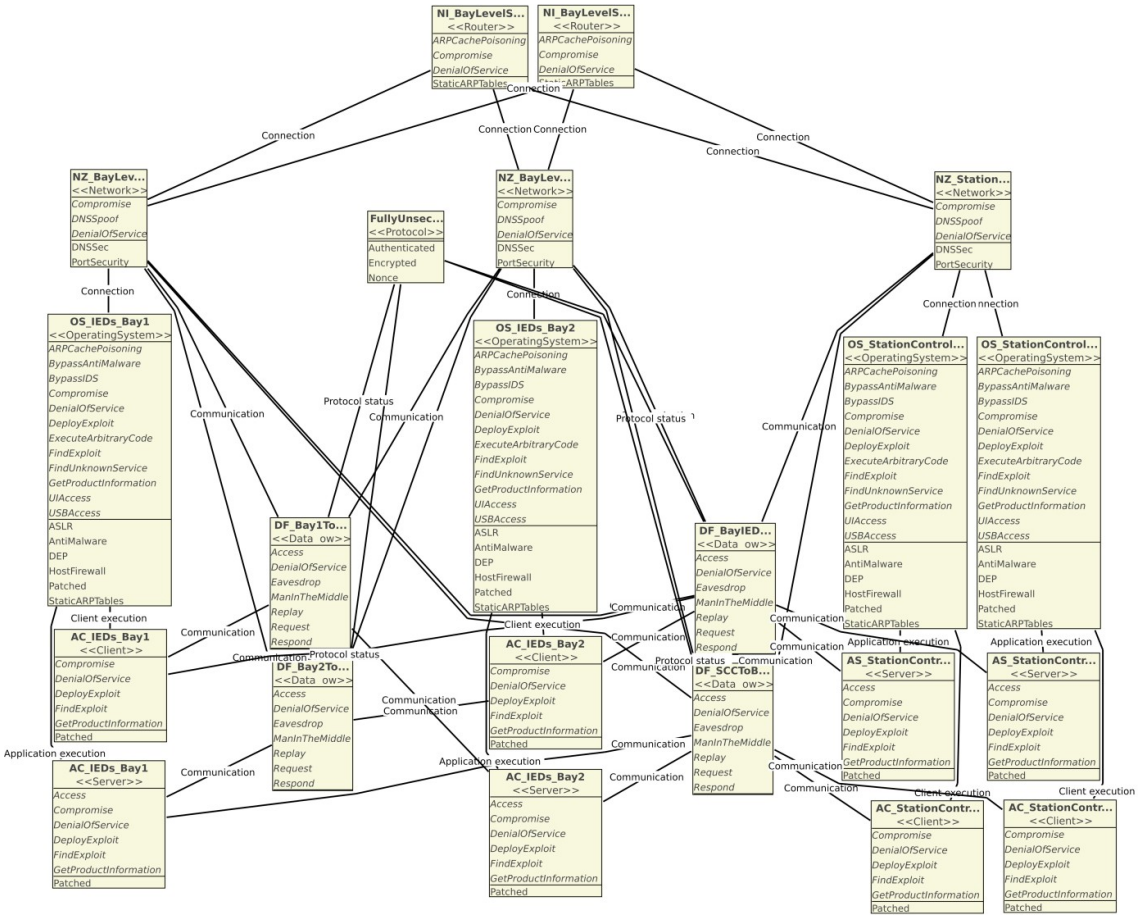

Figure 30: View of the systems, services and network-crossing data flows related to the bay-level networks in a substation.

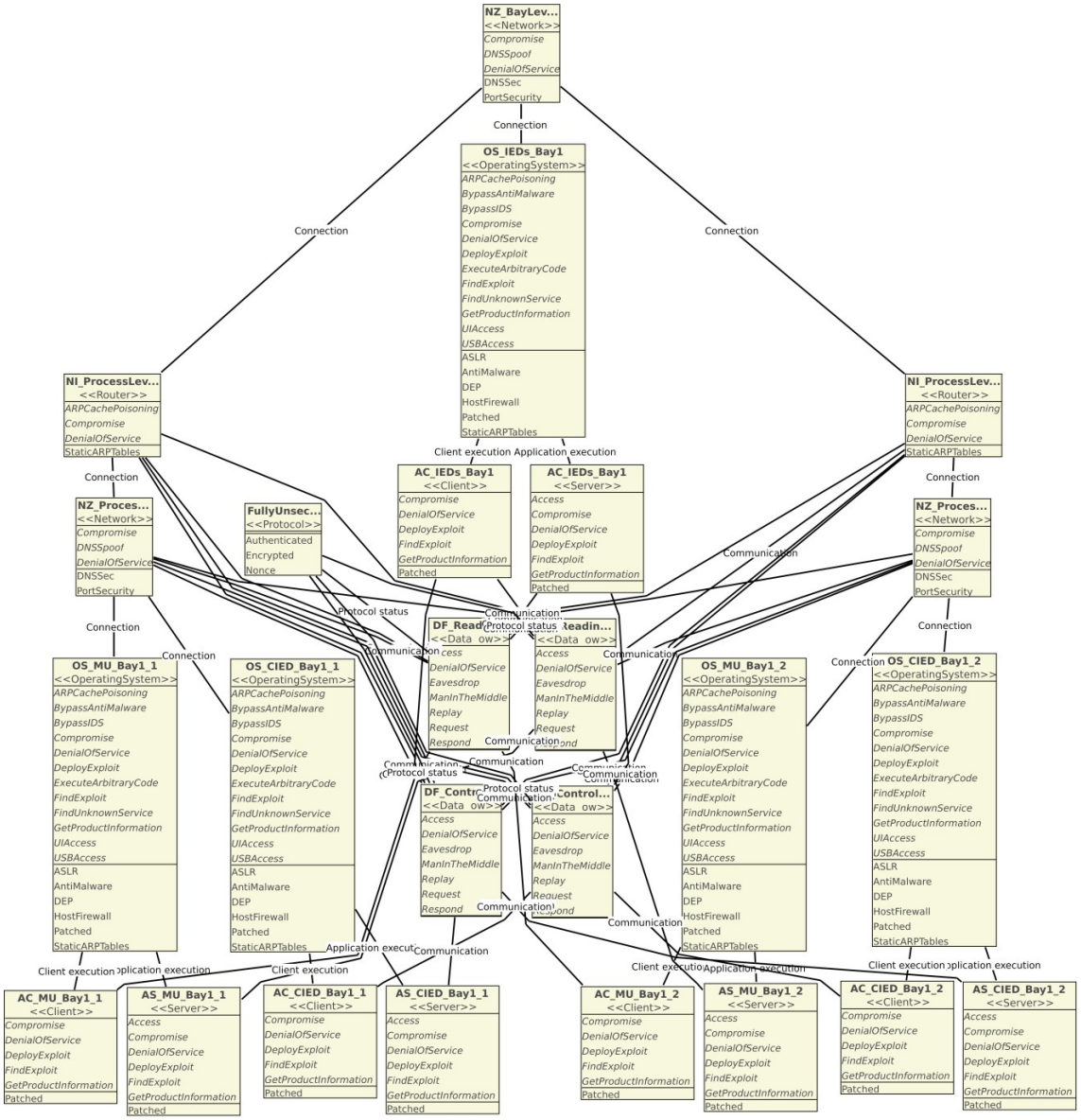

Figure 31: View of the systems, services and network-crossing data flows related to the process-level in a substation.

Finally, most of the assets modelled (displayed as boxes) contain a number of items inside themselves. The items are of two types - attack steps and defence mechanisms. While the attack steps are typically left entirely to securiCAD's calculation when constructing the model, the defence mechanisms (perhaps a more generic term is simply properties) are assumed to as high degree as possible across the entire model. Specifying these properties (defence mechanisms) decreases uncertainty of the model and thus makes is more accurate, making better outlook for the security analysis process to yield accurate results.

Last but not least, a very important step in the modelling process is to place an attacker in the architecture from which the simulated cyber-attacks originate. The attacker placement is shown in figure 32 - a view showing both the attacker and his/her tools, and an arbitrary system placed on the Internet, such as a popular news or e-mail service. The latter of the two systems mentioned is modelled to make the model as close as possible to the reality. Namely, a real cyber attacker will be able to even misuse (e.g., compromise in different ways) public sites and the content they provide, in order to reach the attack goals.

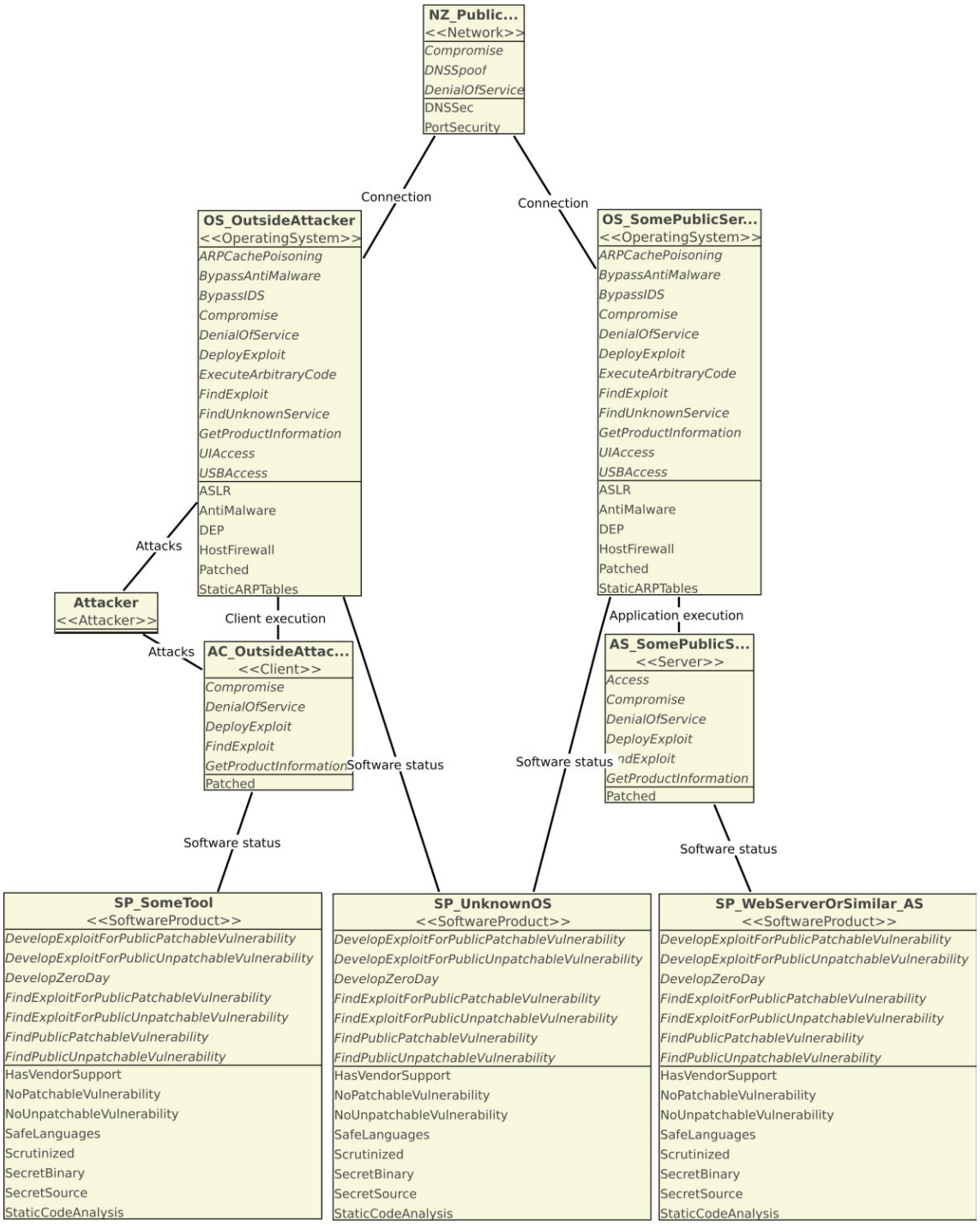

Figure 32: Placement of the attacker in the architecture together with his/her tools, and an arbitrary Internet server.

1.5.4.9 Reference model: Advanced metering infrastructure

By studying existing reference models of AMI (see e.g. [31], [32], [3], [4]) together with two real AMI setups, we have both developed an elaborated conceptual model of the AMI architecture, as well as a model in securiCAD. The former is outlined in figure 33.

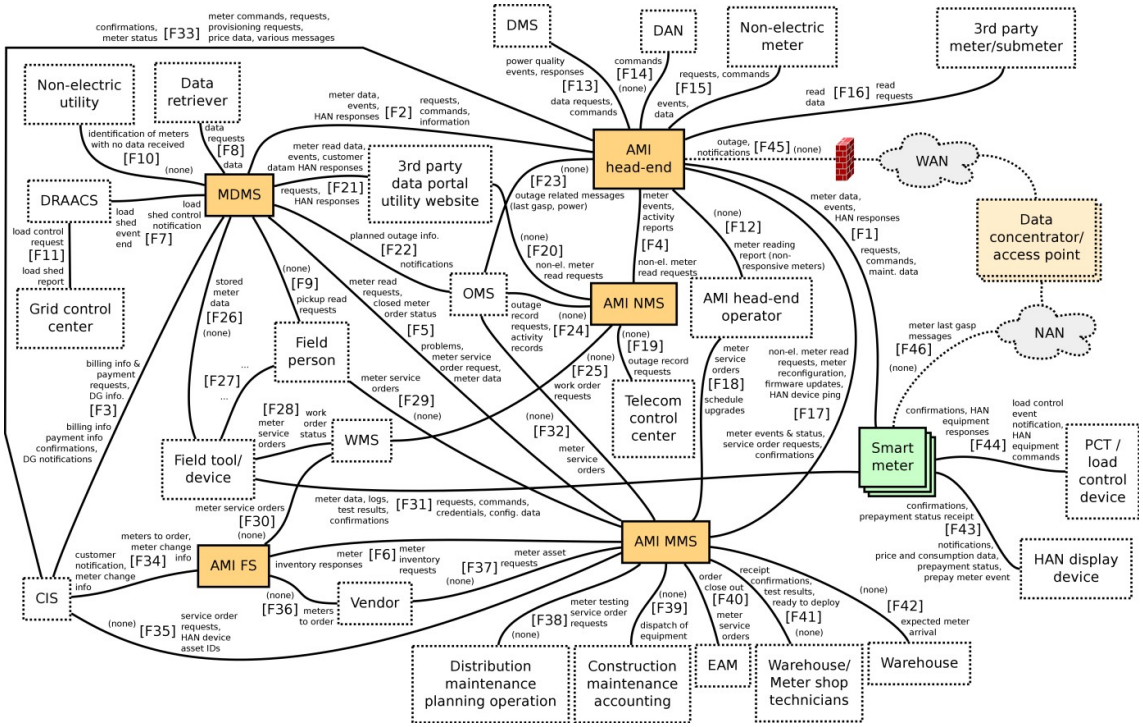

Figure 33: An overview of the AMI reference mode

The securiCAD model consists of 439 objects divided into dozens of views, and can only be presented briefly here, using a few key views.

Figure 34 presents the various main communication networks of the model. It shows all the network zones with associated routers, firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS). From the right (the customer side) to the left (the utility's back-office) in figure 34:

Each home area network (HAN) router is connected to its respective HAN, which is further connected to the smart meter (not shown) and public Internet. Next is the AMI-neighbourhood area network (NAN), in which a data concentrator aggregates the link to a number of smart meters. In this model there are four different NANs of different type of media such as power lines (using power line carrier technology), public and private lines (wired), mesh and point-to-point radio frequency (RF) networks (wireless). The reason is that a variety of data communication technologies are used between the data concentrators and the actual smart meters (not implying that all of these solutions are implemented in a single actual case). Further, one or more wide area networks (WAN) extend between the data concentrators and the AMI head end (HE). The WAN networks also typically have a variety of forms, such as wired infrastructure (e.g., optical), cellular telecommunications network (e.g., EDGE/2G) or other RF network; and can be owned and operated by the utility itself, or by a separate telecommunications company. HE is the first system on the utility side that interfaces with the customer side, which among other processes metering data before they reach further into the utility's server landscape.

The WAN networks are connected with the utility through an AMI demilitarized zone (DMZ), to increase the amount of network separation and so increase security - through ensuring that only the desired field systems can communicate with systems inside and beyond the AMI DMZ. The AMI DMZ has become a part of the reference model despite the fact that not all instances of AMI implement DMZ. Inside the IT landscape of the electrical utility (DSO, distribution system operator) there are several networks such as OT (i.e., operational technology -- such as the AMI meter management system or the AMI head end), IT (i.e., [ordinary] information technology such as the meter data management system or the workforce management system), backbone (i.e., eventual communication lines owned or leased by the utility), a network segment for software update infrastructure, an office virtual

private network (VPN), and a system administrator network; connected using a set of corresponding routers.

Some of the networks and the systems residing in these do not necessarily belong to the AMI itself. However, these systems have tight interconnections with parts of the AMI, which makes them important to model, since security analysis requires having an architectural picture as complete as possible. The reason is that attackers tend to penetrate enterprises using the easiest and cheapest means that are accessible, often proceeding through the least secured zones, which often include front office networks of companies, and other places with a large variety of IT assets, lesser administrative control, and potentially low security awareness at people interacting with the IT assets. The existence and the content of all of these networks vary with different specific instances of AMI. The reference model describes a typical but somewhat extensive case.

Zooming into the domains of the smart meter, the view spans across several networks, consists of a number of systems (the models of which include operating systems, software clients, software servers), and data flows between the systems with corresponding data communication protocols. The view defines defence mechanisms and other properties of the classes (assets) modelled. Defences are specified as either being present, absent, or present with a specific probability (e.g., 0.3, meaning 30% chance of the defence mechanism being present). The latter allows for expressing uncertainty about the defence mechanism's presence, in case a generic architectural picture does not specify whether a particular property or defence mechanism is present or absent (i.e., it cannot be said exactly and certainly for the generic case), or specifies this in a probabilistic matter, e.g., based on how common it is to have a specific defence mechanism present.

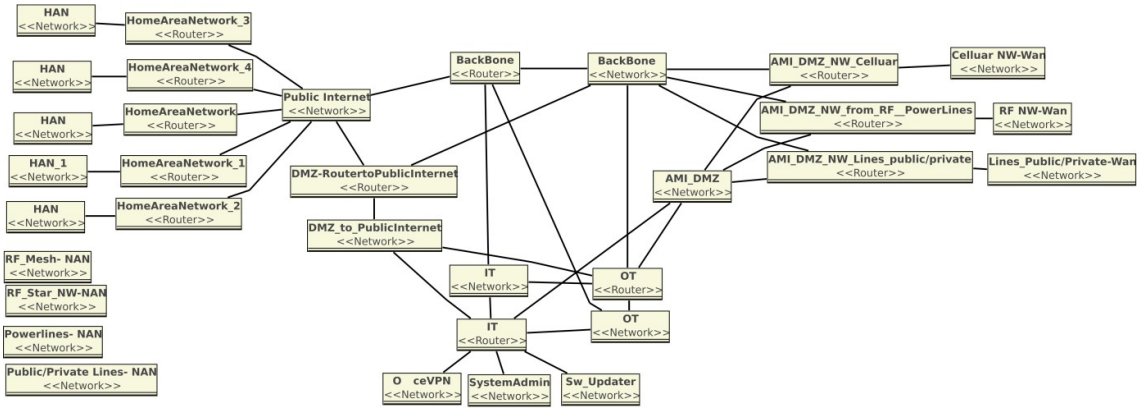

Figure 34: Network zone model of the AMI reference model in securiCAD.

Figure 35 provides a zoomed-in view on some of the systems located at the utility premises, with the MMS in its focus. The MMS system is located in the centre of the figure, with its database connected to its left side. Below the MMS-system there is an access control mechanism attached to the system, which represents access restrictions like log-in authentication with associated user accounts and users further connected to the accounts. The user accounts and users whose access is defined include different technicians, the AMI operator and a vendor, who needs to be given access at times, too. Other systems that are connected to MMS include the outage management system (OMS), AMI forecasting system (FS), customer information system (CIS), and the enterprise asset management system (EAM).

The AMI head end and meter data management system, to which the MMS is connected, are not a part of this particular view.

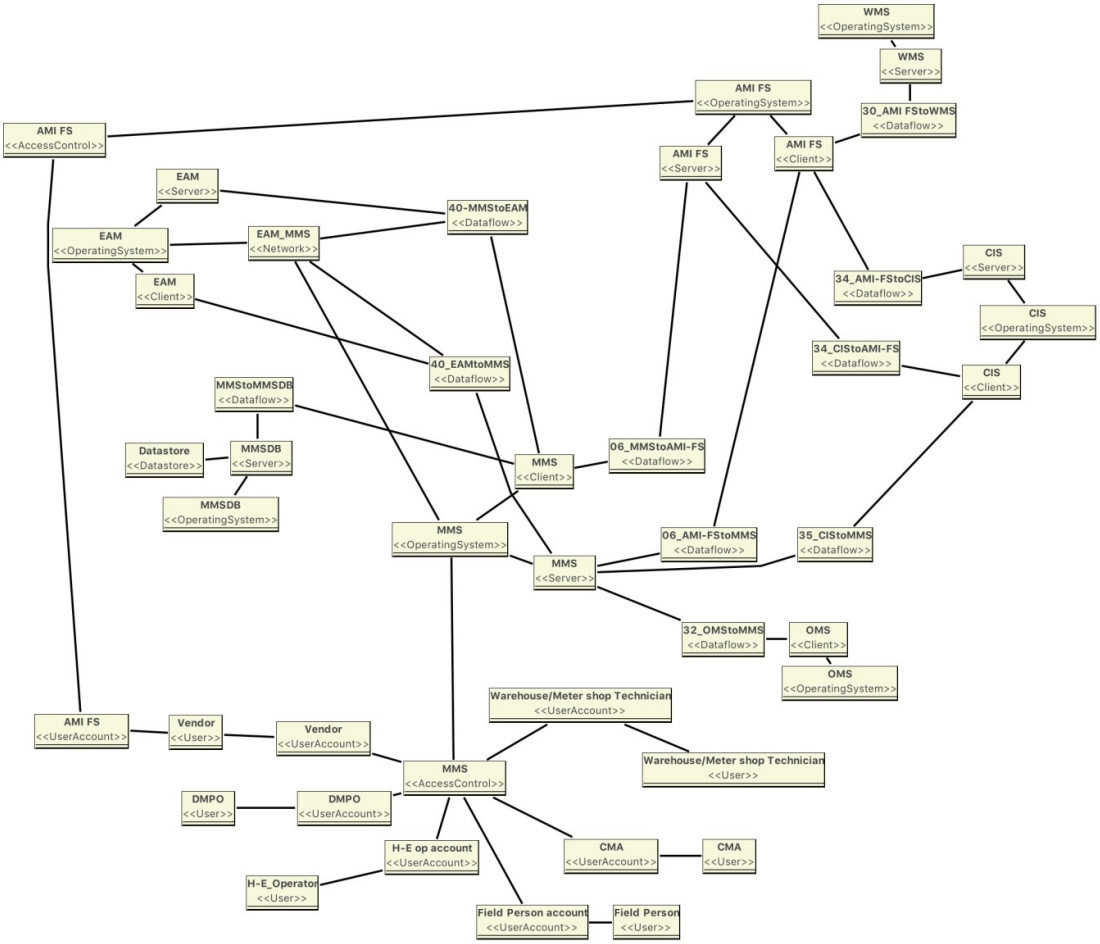

Figure 35: Partial model of the meter management system (MMS).

Similarly, figure 36 provides a zoomed-in view at the MDMS system and its closest surroundings (closest interfacing systems).

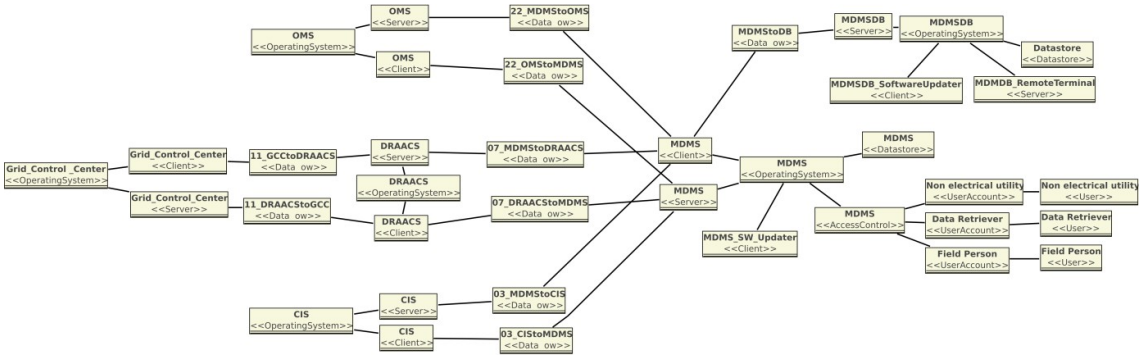

Figure 36: Partial model of the meter data management system (MDMS) and its surroundings.

Figure 37 provides a detailed view of the MDMS system itself. Unlike the views described previously, it also shows attack steps and defence mechanisms (properties) for the different entities modelled. In this specific view, the following entities are modelled:

- MDMS's operating system. It has five defence mechanisms (properties), as follows. The use of address space layout randomization (ASLR), the use of an anti-malware solution, the use of data execution prevention (DEP - a security feature spanning across hardware (processor) and software), the use of a host firewall, the state of being fully patched, and the use of static ARP tables. Each of these properties can be set to a real value between 0 (false) and 1 (true), describing the probability of that

particular property being true. In a scenario of a rudimentary protection (further referred to as "in this instance"), both ASLR and DEP are set to true, since all modern general purpose operating systems running on an Intel x86 based platform support these. The rest of the parameters are left uncertain, as they highly depend on the specific set-up and maintenance of the system.

- MDMS's database (datastore). It has one defence mechanism - the use of encryption on the database level, which is set to false in this instance.

- The IT network zone, within which the MDMS operates. It has two defence mechanisms - the use of DNS security extensions, and the use of PortSecurity, which only allows whitelisted network cards to communicate over the network.

- The access control point of the operating system (i.e., login). It has several defence mechanisms - the use of a backoff technique (e.g., upon several failed login attempts), the state of being enabled, the use of hashed password repository, the use of password hash salting, the absence of default passwords, and the use of password policy enforcement. In this instance, the access control is enabled and has a hashed password repository; however, the use of salting is uncertain, as well as the enforcement of password policy (e.g., minimum acceptable password complexity, or maximum password validity period).

- There is a host-based intrusion prevention system (IPS) integrated with the access control point, running on the MDMS's operating system. The IPS only has one defence mechanism (property) - the state of being enabled.

- Furthermore, there are four pieces of software modelled - two client and two server applications. Each of these applications only has one defence mechanism (property) - the state of being fully patched.

- Each piece of system software (i.e., operating system, client or server application), has a software product entity bound to it. A software product (e.g., Microsoft Office 2010) has eight defence mechanisms (properties) - the state of being supported by its software vendor (and thereby e.g. receiving updates), the absence of patchable vulnerabilities (those for which an applicable patch exists), the absence of an unpatchable vulnerability (those for which there is no applicable patch), whether the software has been written in a (set of) safe language (e.g., memory-managed and strongly typed languages), whether the software has undergone significant security scrutiny during its development, the state of the software having a secret binary (e.g., as a purely in-house developed application, for which the binary has never been shared outside a well-controlled circle and so cannot be obtained by a third party), the state of the software having a secret source (i.e., the opposite of being open-source or having the source code publicly leaked), and finally whether the software has undergone a static code analysis during its development.

- Finally, the model contains an example data flow (automatic software updating), which follows a specific communication protocol (HTTPS). The data flow itself has no defence mechanisms; however, the protocol has three - the use of cryptographic authentication, the use of encryption, and the use of a nonce (i.e., [cryptographic] freshness indication). In the case of the HTTPS protocol, all three defences are set to true.

Additional information can be found in [33].

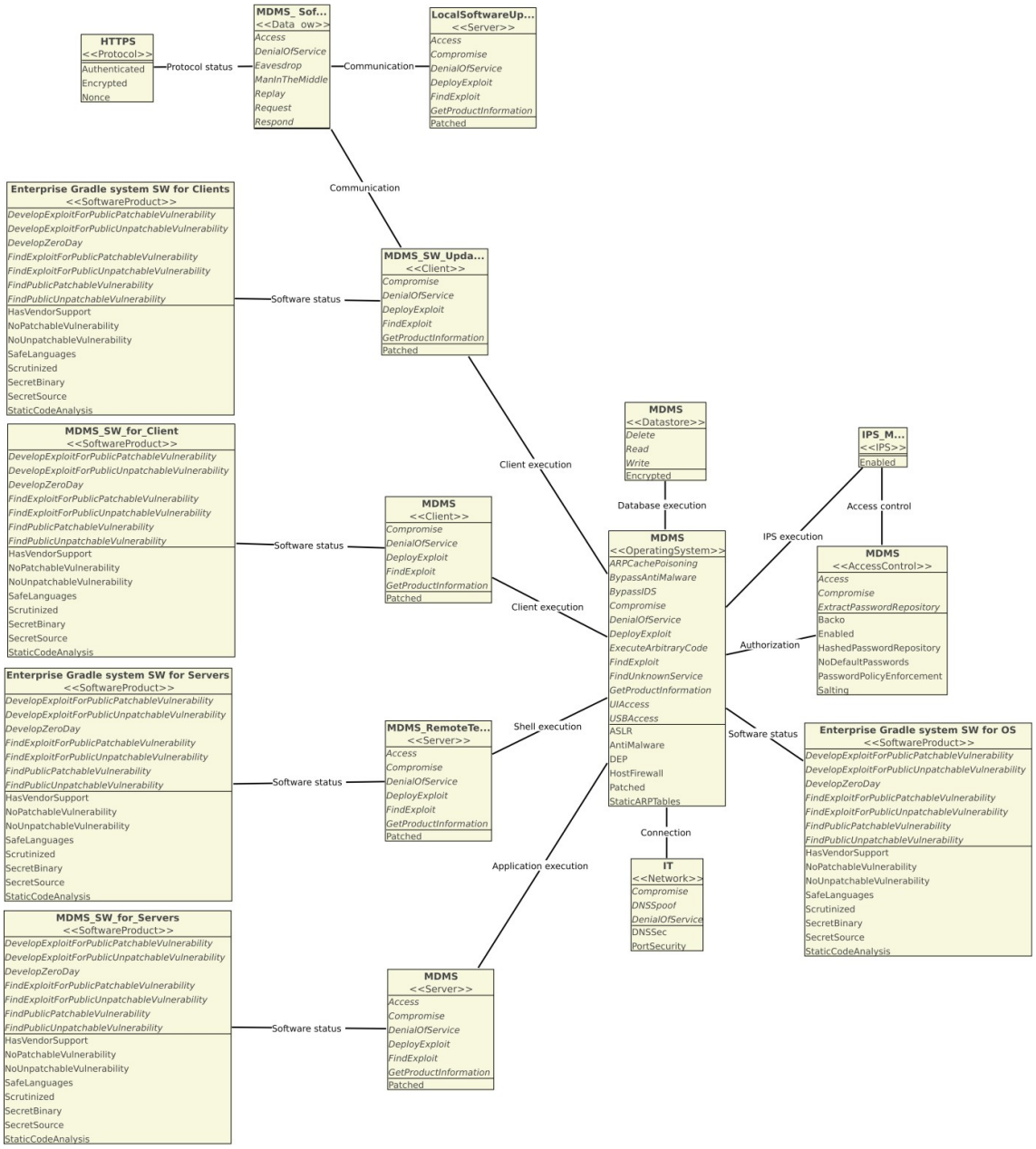

Figure 37: Detailed model of the meter data management system (MDMS).

1.5.4.10 Reference model: Distributed energy resource

A distributed energy resource (DER) is a small power source or sink that can be remotely controlled in order to provide services to the power grid. Examples of DERs include photovoltaic systems, wind turbines, diesel generators, biomass power sources, energy storage systems, electric vehicles as well as various kinds of loads.

The IT architecture of the solutions for controlling and supervising DERs is typically simple, consisting of the DER unit's controller, some form of integrated controller (an intelligent electronic device, IED), and an external controller system, which can be connected to it. This external controller system is typically proprietary, using proprietary protocols for data communications, through which the DER control and supervision is realized. DERs are, however, starting to be built with support for standardized protocols such as DNP3 and ICCP (see [34]), which enables their easy integration with a distributed control system such as SCADA. Whether a specific DER is of the former or the latter type is likely going to depend on the purpose of the DER. The simplicity and highly customized, consumer centric control functions of a DER might suit households better than the highly standardized interfaces necessary for a direct centralized control from a power utility.

An overview of the DER control architecture in the two above mentioned variants is shown in figure 21.

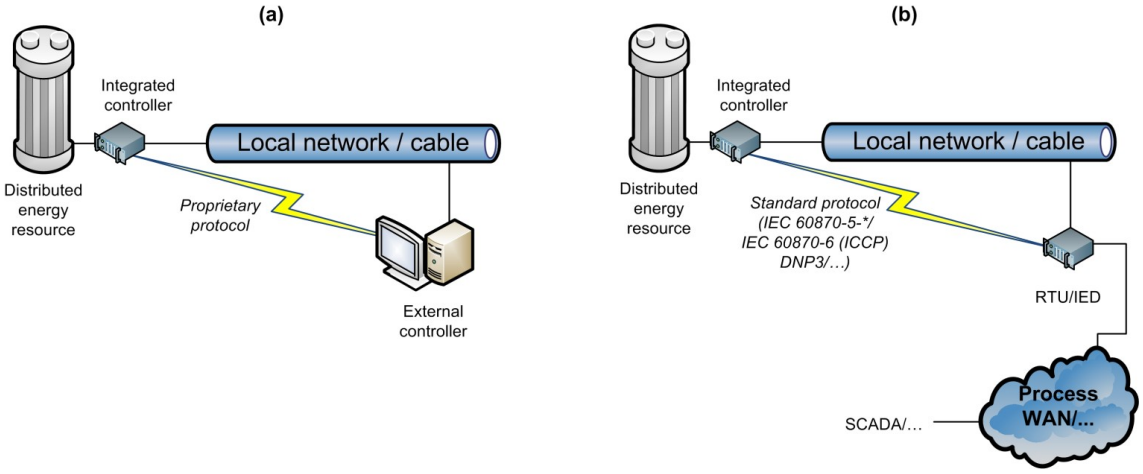

Figure 38: Overview of DER control architecture - using proprietary (a) and standardized (b) approach.

The external controller is assumed to be an ordinary [office] personal computer or workstation. Details on these devices are discussed in sections 1.5.4.11 and 1.5.4.12.

1.5.4.11 Reference model: Enterprise and office IT environment

Given a sensible configuration of networks and systems at a power utility, the Office IT and enterprise networks are arguably distant from the critical control networks, and/or their interconnection with these is well-protected. However, such a belief does often not hold well, mainly for two reasons. Firstly, the "office/personal IT world" tends to creep into the control networks as time flows, through new interconnections needed due to business optimization or simply due to convenience for employees (e.g., bringing their own devices such as smartphones, tablets, laptops, soon wearables to premises where critical control networks span; browsing the Internet or receiving e-mails on computers used in control networks; or the ability to access control systems from an office network to obtain live or fresh operational data - to name a few).

Secondly, the "office IT world" is indeed very close to the control networks anyway, since a modern day utility needs interconnections and automated data exchanges between these network spaces in order to achieve the business efficiency and thus competitiveness that at least allows the utility to stay on the market. From the attacker's perspective, this makes the office network space a highly viable pathway into the critical control networks of a power utility, without mentioning that employees often tend to circumvent IT security policies due to convenience, even if such policies exist and are effectively communicated. Put simply, enterprise and office IT environments are both threateningly close to the control networks of a power utility, and threateningly insecure due to the complexity and the broad variety of different systems operating inside, as well as the generally low level of access restrictions and high diversity and unpredictability of all IT activity that needs to take place inside these networks for legitimate business reasons.

A brief overview of an enterprise network is described in the rest of this section. Firstly, the reference model of the enterprise and office IT environments is divided in four main segments - public DMZ, office network, engineering network, and intranet network - as depicted in figure 39.

The overall assumption related to the enterprise network architecture described in this section is that the company maintains a decent level of security rather than a "sub-standard" one. It is not assumed that the utility has invested large sums of money in advanced and

sophisticated IT security solutions; however, basic, relatively cheap and simple defenses, such as appropriate network segmentation and an enterprise-grade firewall at the network perimeter, are counted with as the typical case. Hence, the division of the enterprise IT environment into four segments has been made.

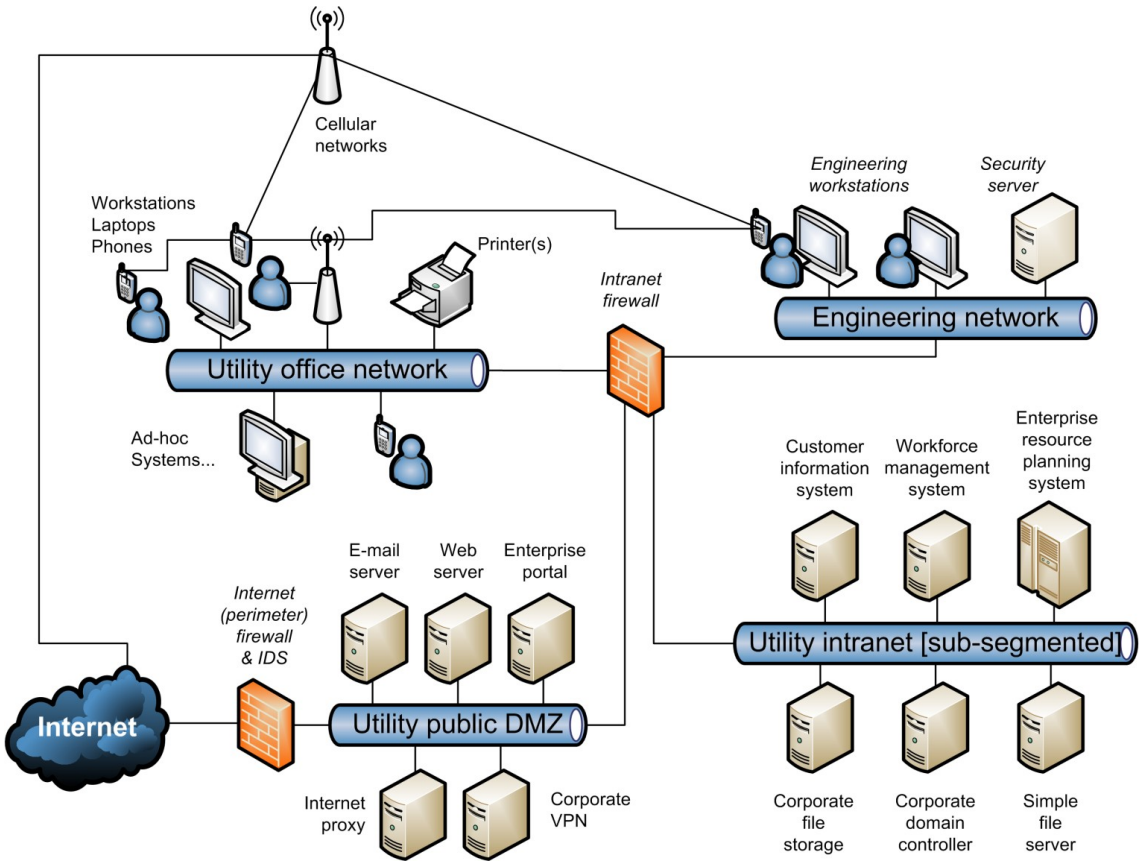

Figure 39: Overview of enterprise IT networks

Utility public DMZ

The public DMZ network is one that hosts servers that require network access initiated from the outside (e.g., the public Internet) or are highly likely to become compromised. This is typically the company's web server and an eventual customer portal, e-mail server, corporate VPN, and the like. This network can even host an enterprise Internet proxy, through which all web communication can be routed, both for performance, efficiency and security reasons.

The generally assumed layout of this network is that all the mentioned servers are hosted on the single network. However, in a more advanced case this network can be segmented further into smaller segments, for example to have a separate segment for each logical function such as the corporate web (a separate network for only hosting corporate web servers, their databases and eventually related load balancers), or e-mail.

As described, the public DMZ is the network that resides virtually closest to the network perimeter; however, this does not imply that no part of the other networks can be closer - namely, there are office computers and cell phones used by users that have access to public networks, may lack security awareness and appropriate cautiousness, which can make them more prone to become victims of a malware attack that further spreads and infects whatever it happens to within the company's IT environment.

Utility office network

The utility DMZ is the network, in which most employees' computers are hosted, and from which the employees work. The employees are the human operators of their computers and

commonly differ in terms of their user activities and behaviour, as well as their security awareness. Additionally, the office spaces might also use wireless networks for convenience, to which other devices (e.g., private smartphones, tablets, wearables and the like), can connect. Not only can these devices connect and operate over these networks; they can in parallel use Internet over cellular networks, and can already be compromised.

To sum up, the utility office network is one of those with the most diversity in network behaviour, the least control and enforcement of this behaviour, and the highest risk of assets being compromised by a piece of malware.

The office network can also be sub-segmented for security reasons, e.g. according to the different organizational units or departments present. This practice would normally depend on the size of the power utility as a company - the larger company (and/or the more geographically distributed one), the more likely the sub-segmentation of the office network.

Engineering network

In an electrical utility, a specialized form of the utility office network is the engineering network. This specific network can be used to host engineering workstations and servers related to substation-level equipment (substation automation systems), from which the engineers can review, design, simulate/test and maintain electrical protection schemes, substation automation devices, etc.

Unlike the general office network, the engineering network can be more restricted, e.g. in terms of not having direct Internet access allowed, or having it heavily restricted to what is needed for the business.

Utility intranet

The intranet is a network that hosts company-internal servers that need to be accessible from inside the company (e.g., some of all the networks of the company), but not from the outside (e.g., from the public Internet). Systems that can typically be found here are those that hold critical/important information in support of the core business (e.g., customer information system, enterprise resource planning system, workforce management system), those that provide common IT infrastructure for sharing information among employees, organizational units/departments, etc. (e.g., corporate file storage, simple file server such as a FTP server), and those that provide basic infrastructure services such as domain name resolution and other domain services including authentication and authorization (e.g., enterprise-level domain controller).

As is the case for some power utilities, certain intranet servers need to be accessible from the control system network (or need to access this network); while they also need to be accessible from other utility-internal networks such as the office network. Such interconnections can be misused by attackers to penetrate deeper into the IT infrastructure and especially sensitive and commonly well-protected networks such as a central control system network (where e.g. the SCADA system resides).

Unfortunately for the ease of assuring IT security in an IT landscape, the trends develop in the direction of having more interconnections among systems; and we are certainly going to see more of it in the coming years and decades. Similarly to how the public DMZ and the office network can be sub-segmented, the intranet can also be divided into several separate network segments to further increase the IT security by adding more barriers between the different systems that normally do not need to interoperate, or do but in very specific and well-controllable manners (e.g., only unidirectional requests over TCP protocol on a specific port, to a specific hostname or IP address). Such steps limit the possibilities for attackers to maneuver around inside the utility's networks, and can be found in the more security-aware and hardened enterprises.

More details regarding the types of systems hosted in the above described enterprise networks are described in the following section related to common operating systems.

1.5.4.12 Reference model: Common operating system

In CySeMoL/securiCAD models, each operating system (OS) of a computer could be modeled separately together with its parameters. However, if greater amount of detail such as server applications (e.g., remote access service) and clients (e.g., web browser) running/used on the computer need to be modeled, this approach can be demanding and time-consuming. The aim of this section is to describe a set of operating systems commonly found in specific parts of a power utility, together with common software packages that are both used and relevant to CySeMoL/securiCAD modeling.

The descriptions of the common operating systems below are based on an assumption that the power utility resides in Europe or in the US; as well as that the utility has some formalized processes for IT and IT security management, and a decent level of these achieved. It is here admitted that the descriptions and might not be representative for all power utilities in the world.

Lastly, the rest of this section mentions security-related attributes used in CySeMoL, which can be read more about in [35].

Enterprise office workstation

Perhaps the easiest one to imagine, the enterprise office workstation, is typically a personal computer (stationary computer or laptop), running an operating system with graphical interface. For today's power utilities, the most common such system is a reasonably recent version of Microsoft Windows (e.g., Windows 7, Windows 10, eventually Windows XP; however doubtfully older systems for common office use). Alternatives such as Apple's Mac OS or Linux might be used by some individuals (particularly the former), however, are far from the mainstream. All of these systems normally have vendor support including regular security updates, and support for protection techniques such as address space layout randomization (ASLR) and data execution prevention (DEP). Furthermore, most of these systems have undergone thorough security scrutiny in their development, including static analysis, and their source code is not open to the public domain (the last with the exception of Linux). All of these operating systems including their basic applications are written in "non-safe" languages such as Assembly language and C/C++. The installation of the office workstation operating system normally has an anti-malware solution installed and enabled, and has a rudimentary host firewall (in Windows 7 and newer). These systems do not feature static ARP tables and the degree to which they are fully patched depends on the company policy and enforcement of software patching - typically high, but less than 100%.

Apart from the operating system itself (the kernel and a set of basic programs and libraries), the enterprise office workstation also commonly features a number of other software packages that are both commonly installed and used:

- Web browsers are used for much of the work and computer-related leisure activity done by the office staff - from using internal systems, much of which have web interfaces, to private web browsing. In many cases the web browser comes with the OS vendor (Microsoft), and has the same security properties as the operating system itself. So can be assumed about other proprietary web browsers (e.g., Safari, Opera). In the open-source cases (Firefox, Chrome/Chromium), the security properties are equal except their source code being open to the public domain. Lastly, web browsers that are different than the OS vendor have a slightly lower probability of being fully up-to-date, which further depends on whether they feature an automatic updater program that installs an update whenever available.

- E-mail client (e.g., Microsoft Outlook, Mozilla Thunderbird) is commonly used for accessing corporate e-mail, typically using Microsoft Exchange as the data communications protocol. When it comes to security properties, those of web browsers can be applied.

- Office suite (e.g., Microsoft Office, OpenOffice/LibreOffice) is used for most office work that has to do with reading and writing/editing documents. Unlike the web browser and e-mail client, the office suite does not frequently communicate over network, however, it is often used to open received file attachments from e-mails and files that are downloaded from the corporate intranet or the public Internet. The security properties are similar to those of the web browser, except that the security scrutiny is assumed to be lower, and "safe" languages (e.g., C# .NET and Java) are used.

- PDF reader (e.g., Adobe Reader or Adobe Acrobat) is used for reading PDF documents. The security properties are similar to those of the web browser.

- Media player (e.g., Microsoft Windows Media Player or VideoLAN's VLC) allows users to play media like music and video recordings, and is a piece of very commonly installed software. The security properties of the web browser can be assumed, except that the security scrutiny is low, among other due to a large variety of plugin-like codec libraries used.

- Corporate file access clients (using e.g., Microsoft SMB protocol) for accessing corporate file shares. The security properties of the OS can be used, as this package typically is a part of the OS.

- Remote access client (e.g., Microsoft Remote Desktop, Citrix, VNC, ssh) for accessing
- graphical interfaces of remote computers (typically servers) are used in certain cases, which highly depends on the IT systems landscape of the power utility company. For software from the OS-vendor (i.e., Microsoft Remote Desktop for Windows and ssh for Linux), the security properties of the operating system can be assumed; otherwise, those of the web browsers can be assumed.

- FTP client for accessing FTP shares, which is commonly only used by certain employees, if at all. For this purpose, a web browser or a dedicated stand-alone FTP client can be used.

- Software updater(s) for the operating system (a part of it), and eventual other software packages, which allows user notifications of available software updates and/ or automatic installation of these. The software updater normally uses a web protocol (e.g., HTTP, HTTPS, FTP, FTPS).

Server software is not typically installed on enterprise office workstations, with the exception of remote access services for allowing IT administrators to perform maintenance, or eventually allowing employees to log in to their workstations remotely through a VPN connection. See "Remote access client" above.

The presence of other software packages or eventual deviations from this description can depend on specific conditions and specific environment, in which the enterprise office workstation is used.

Engineering workstation or laptop

The engineering workstation or laptop is similar to the enterprise office workstation, except that the variety of software might be more limited, depending on the corporate policy. Except the software packages described for the enterprise office workstation, the following can be assumed present:

- A proprietary software package for simulation (of e.g., electrical protection schemes). This package can be assumed to have vendor support, not be open-source, be written in "non-safe" languages, not be security-scrutinized, but have static analysis. This package might communicate with a server over a local network using proprietary protocols.

- Proprietary software for work with configuration files of electrical power equipment like protection relays (IEDs) in substations. This package can be assumed to have vendor support, not be open-source, be written in "non-safe" languages, not be security-scrutinized, but have static analysis. This package will most likely not communicate over network at all; the configuration files will typically be exchanged separately as files.

It is very likely that the corporate file access client and even FTP client is used for exchanging engineering files and documents.

Mobile phone or tablet

Having smart personal mobile devices that essentially are powerful computers is a rather recent trend (of the past decade), however, a very far-gone and quite a ubiquitous one. Soon we are also having wearable devices like smart watches having wireless connectivity and general-purpose operating systems running on them. The specialty of these devices is that they both can be hosted on and communicate with the corporate office network, as well as an external cellular network. Moreover, they can contain arbitrary software applications and even malware infections from the owner's prior and arbitrary use of these devices. This makes them arguably risky to connect to and operate in corporate networks. It is, nevertheless, highly common to see employees having configured access to their corporate e-mail alongside all private data and accounts. What is especially worth noting is that smartphones and similar highly mobile devices are fairly common for people to lose (forget somewhere like in a taxi or at an airport), and/or get stolen with potentially valuable data on them (including corporate login credentials) and little protection of these.

Unlike personal computers, these devices use embedded flavors of operating systems like Windows IoT or Windows Embedded, Apple IOS, Android, Embedded Linux, and the like. These operating systems largely share the security properties of their personal computer or server counterparts, though. The typical selection of application software differs on mobile devices like smartphones, tablets etc. They rarely use any server applications, however, can literally feature a ton of different user applications installed. Some of the very common ones are web browser, e-mail client, office suite, PDF reader, and various social media applications that are largely based on web data communication.

Office device (e.g. network printer)

Office devices like network printers are in their nature embedded systems (a little like mobile devices described above) and often use well-known libraries and server applications (even open-source ones), however, they only feature a very limited set of functions. On the other hand, they very rarely if ever undergo software (firmware) patching and updating, which makes them rather constantly vulnerable to known attacks. These devices tend to use open-source operating systems like Embedded Linux or a variant of BSD.

Although these devices do not normally use clients applications, they use server applications to listen for ordinary requests (e.g., printing), as well as configuration requests, for which they often use a web interface (and therefore run a web server with a web application), or some proprietary interface. Similarly to the operating system itself, these applications remain normally also heavily outdated, since an update depends on both a release of a new firmware for the device by its vendor, and a manual firmware update operation.

Enterprise server

There is a broad variety of different enterprise server configurations, dependent on what they are used for. However, some typical patterns can still be captured. The type of enterprise server imagined here is one that hosts a web server, an e-mail server, a database, s file server, an application server, or some other enterprise server that often also uses some of the previous.

Enterprise servers use operating systems that are similar to those used in office workstations, however, in different flavors (suited for server use rather than a desktop). Typical operating systems are Microsoft Windows Server, an enterprise-grade Linux (e.g., Red Hat Enterprise Linux) or Sun Solaris. Unlike with enterprise office workstations, the distribution of Windows- and non-Windows-computers is much more even in the enterprise server world.

All of these systems normally have vendor support including regular security updates, and support for protection techniques such as address space layout randomization (ASLR) and data execution prevention (DEP). Most of these systems have undergone thorough security scrutiny in their development, including static analysis, and their source code is not open to the public domain (the last with the exception of Linux). All of these operating systems including their basic applications are written in "non-safe" languages such as Assembly language and C/C++. The presence of an antimalware solution and a host firewall is uncertain. These systems do not normally feature static ARP tables and the degree to which they are fully patched depends on the company policy and the diligence of the system administrators in software patching - typically high or very high.

Enterprise servers do not normally run client applications at all, and some servers do not feature a graphical user interface, either. Besides the server software providing the primary functionality of the server (e.g., a web server, e-mail server, database server, file server or some other similar server), the following software is commonly installed:

- Software updater(s) for the operating system (a part of it), and eventual other software packages, which allows user notifications of available software updates and/ or automatic installation of these. The software updater normally uses a web protocol (e.g., HTTP, HTTPS, FTP, FTPS).

- Remote access service (e.g., Microsoft Remote Desktop, sshd) for allowing IT administrators to perform maintenance. These server packages normally inherit the security properties of the operating system.

Control system server

The control system server can be seen as a special variant of the enterprise server (see above). The exact setup of the control system server depends on the specific implementation of the control system; however, the following can be noted:

- In case of a large-scale control system like SCADA and high availability requirements, the server is likely to run a Unix-like operating system like an enterprise-grade Linux or Solaris. In case of small-scale control system like a substation control system, the system is very likely to run a Windows operating system.

Unlike enterprise servers, a control system server is very unlikely subject to frequent software patching, for the reason of high availability and the need to thoroughly test products upon changes, even at the control system vendor's side. Updates to control systems exist; however, they are rather released a few times a year than a few times a week or month as it is for much enterprise software including common operating systems. Consequently, it can be assumed that control system servers are always vulnerable to a portion of known attacks.

Remote terminal unit (RTU)

A remote terminal unit is an embedded device that can act as an interface between IT/OT-infrastructure and a physical process, through translating logical commands into control actions on physical devices, and taking and translating analogue measurements from the physical process into digital data sent further. In short, RTUs provide telemetry data from a physical process, and mediate control of the physical process. RTUs can also be used as a bare communications interface between components like IEDs (e.g., for protection and bus control) and a more central piece of the control infrastructure such as the SCADA front end / master terminal unit (see further above in the description of the SCADA reference model).

Such setup is more typical for IEC 61850 based substation environments. Some RTUs also provide the functionality of a programmable logic controller (PLC). Modern RTUs can feature additional functions such as an integrated human-machine interface (HMI), even a web-based one.

There exist many RTU products in operation, which differ in terms of their hardware, operating systems they use, the features they provide, etc. Some RTUs have highly restricted feature sets and use special real-time operating systems like WindRiver's VxWorks or QNX; some use an embedded variant of Linux or Microsoft Windows. RTUs also differ in terms of what services they expose to the network; however, exposing some sort of interface for convenient overview and configuration by an engineer, is common. Such interface can be based on a proprietary protocol, or web technologies involving a web server program running on the RTU. RTUs also commonly support remote access through ssh or telnet, and time synchronization over SNTP (able to act both as a client and a server). RTUs communicate over Ethernet and IP (in addition to other interfaces), and support telemetry protocols such as IEC 60870-5-10x, IEC 61850, DNP3, Modbus, SPA etc.

RTUs normally have vendor support. However, support for protection techniques such as address space layout randomization (ASLR) and data execution prevention (DEP) is uncertain, likely unavailable. Most of these systems have undergone security scrutiny in their development, including static analysis. Their source code may or may not be open to the public domain. All of these operating systems including their basic applications are written in "non-safe" languages such as Assembly language and C/C++. It can be assumed that no anti-malware solution or host firewall is present. These systems do not normally feature static ARP tables. It is highly unlikely that these devices are fully patched due to their embedded nature and their most often availability-critical use.

Intelligent electronic device (IED) and programmable logic controller (PLC)

An intelligent electronic device is an embedded device similar to an RTU, except that it is more specialized for a particular function or a set of functions (e.g., protection of an electrical bus, line, transformer; or operation/control of an electrical device like a transformer, breaker etc.). It is essentially a sophisticated controller. An IED is typically connected using Ethernet and IP; configurable using a proprietary configuration protocol, and time-synchronized using SNTP. It typically supports protocols such as IEC 60870-5-103, IEC 61850, DNP3, Modbus, SPA, LON etc.; but also FTP for work with configuration files. Not all of these services need to be enabled even if supported, though.

A programmable logic controller is an embedded device that provides rather simple, but highly reliable and rugged industrial computing for controlling an industrial/physical process. Many PLCs are connected using Ethernet or a serial connection (in addition to other interfaces). Similarly to an IED, PLCs are also typically configured using proprietary software. Modern PLCs can communicate with RTUs or SCADA systems using protocols like Modbus or Profinet, and can even expose a web interface from which they can be controlled.
Both IEDs and PLCs use similar operating systems as RTUs (see above), and their security properties can also be assumed equal. Both IEDs and PLCs are commonly IP-enabled.

1.5.4.13 Reference model: Substation automation system (component)

The architecture of the substation automation infrastructure has been described earlier in the document. This section attempts to briefly describe its core components.

Substation control system (SCS)

A substation control system is a smaller version of a SCADA system that features a human-machine interface for personnel in a substation to be able to supervise and control the physical process. This system typically runs on a workstation computer running Microsoft Windows, and its characteristics correspond to a "Control system server" as described in the previous section.

Remote terminal unit (RTU)

Remote terminal units have been described in the previous section.

Intelligent electronic device (IED)

Several types of intelligent electronic devices can be found in a substation, including:

- Bus control IEDs
- Busbar protection IEDs
- Transformer protection IEDs
- Line distance protection IEDs
- Line differential protection IEDs

These different types of IEDs typically have the same basis (if they originate from the same vendor). Further, there are IEDs responsible for solely controlling advanced equipment such as transformers. Intelligent electronic devices have been described in the previous section.

Merging unit (MU)

Merging units are simple devices that converts analog measurements from a physical process sensor (e.g., from a voltage or current transformer) to a digital value, using IEC 61850-9-2 (sampled values) or IEC 81850-8-1 (GOOSE). They can also provide time synchronization and IEC 61850 access points.

Merging units are typically connected using Ethernet, and configurable over a USB interface. Operating systems and security properties of merging units can be assumed equal to those of IEDs and RTUs.

Time source unit

A time source unit is a specialized embedded device similar to an IED (see above), however, it only provides time (e.g., from GPS, DCF-77 or IRIG-B) using the SNTP protocol. It can additionally provide SNMP interface for diagnostic purposes. Similar operating systems and security properties than those of IEDs and RTUs can be assumed.

Engineering/diagnostics computer

In a substation, a mobile computer (laptop) for engineering and diagnostic purposes, is needed. This computer can be assumed running a Microsoft Windows system with an "engineering workstation or laptop" configuration (described in the previous section), however, there is a high likelihood that the computer is not fully patched.

Apart from the commonly installed software described earlier, the engineering/diagnostics computer is using pieces of proprietary software from vendors of the devices that support diagnostics and configuration in the substation.

1.5.5 Cyber vulnerability analysis

This section describes the results of the cyber security evaluation of the reference models which have been discussed above. It also presents an evaluation of a set of countermeasures in terms of their efficiency, and discusses their cost.

1.5.5.1 Materials and methods

The reference models formulated above were used as the models on which all security analysis was performed. They were modeled as a single large model representing a comprehensive IT environment of a DSO utility and some additional IT resources outside of

the DSO utility. Those additional IT resources include a few Internet hosts together with the attacker, and simplified IT environments of parties typically interconnected with the DSO such as a household customer, a TSO utility, an AMI data hub and an energy supplier company.

The method used during the security analysis consisted of three stages, which are described below.

Stage 1: Identification and modeling of scenarios

Two different sets of scenarios were used, to gain insight into two different issues:

1. The effectiveness of different protection strategies. There was one baseline scenario that reflects the assumed typical configuration for all of the reference models, and an attacker that resides purely outside on the public Internet. Additionally, for each reference model, eleven different protection scenarios were formulated. These protection scenarios were of the same type for each reference model, however, as the reference models contain different content (i.e. different computer hosts, services, data flows, etc., and the interconnections between these), the application of the scenario to each reference model resulted in an unique new model that had to be calculated separately (see stage 2, below). In this fashion, this process resulted in creating 44 models in addition to the baseline model - four [reference models] times eleven [scenarios].

   The protection scenarios applied to each reference model were the following:

   a. Use of anti-malware solutions on all hosts.

   b. Use of encryption on all data flows (e.g., using TLS or similar transparent technique).

   c. Use of firewalls on network boundaries.

   d. Hardening of hosts (i.e., modifications of hosts so as to reduce their attack surface, e.g. by removing unnecessary software, services, service features, etc.).

   e. Use of intrusion detection systems on all hosts.

   f. Strict network security configuration, abbreviated to "Network sec.", which implies whitelisting of hosts and network devices, the use of static ARP addresses on network devices and hosts, and DNSSEC extensions where applicable.

   g. Use of intrusion detection and prevention systems on all networks.

   h. Highly granular network segmentation (e.g., a separate network for a single type of host).

   i. Enforcement of a password policy.

   j. Assurance of diligent and effective systems patching;

   k. Assurance of software scrutiny in the software development process (from a software user's perspective, this can be understood as the choice of secuity-scrutinized software, or an assurance of a similar effect).

2. The impact of different attacker placement (i.e., different starting point of the adversary who aims to harm IT resources). In addition to the baseline scenario mentioned above, five subsequent scenarios were branched from the baseline scenario, each branching modifying it by adding to the outsider attacker the following:

a. Attacker inside the office network (e.g., a staff member).

b. Attacker inside the engineering network zone.

c. Attacker controlling an IT asset inside a substation.

d. Attacker controlling the administrator's computer (NB: not a malicious administrator).

e. Attacker inside a customer household.

The security impacts were evaluated for all reference models in each of these scenarios. This resulted in creating five models in addition to the baseline model and the models mentioned above.

Stage 2: Calculation of the scenarios for each reference model

In total, all 50 models were calculated using the tool securiCAD developed by foreseeti, a commercial tool partly based on the research results of the CySeMoL prototype. The following configuration was used for each calculation. The number of samples was set to 2000 in order to reach low variance between calculations of the same model (NB: none was detected using this setting), yet keep the calculation times reasonably low (NB: a single calculation took around two hours to complete). The time-to-compromise (TTC) infinity threshold value was set to one thousand 1000, causing any attack path that requires more than 1000 hours of work from a professional penetration tester to not be calculated further. The reason for this limitation is to control the algorithmic complexity and hence the amount of time needed for each calculation. The motivation for the specific value is that the range between 0-1000 of attacker hours provides sufficient resolution for seeing the attacker reachability or effectiveness of defenses in the studied architectures (NB: the results have shown that the limit has not been reached by the resulting TTC values for almost all evaluated attack steps with value expectation different than infinity, and hence the limit is considered confirmed sufficient).

Following each finished calculation, the calculation results were exported into a CSV file. The CSV files generated this way contain all attack steps and defenses across the entire modeled architecture, and their values - TTC-triples for attack steps and defense configuration for defenses. The values generated in each samples were not exported, as they were not intended to be analyzed, while their inclusion would vastly increase the export times and the storage space demands.

Stage 3: Analysis of the calculation results

For analysis of the calculation results, the statistical language and computation environment R was used, together with RStudio, an integrated development environment for work with R. First, an analysis script converting the exported results from a securiCAD model calculation into concise and human-intelligible figures was developed. Subsequently, the script was applied to the calculation results of all models resulting from applying the different scenarios to the reference models. Finally, the results were studied, conclusions drawn and described.

The results from securiCAD come in form of a vector of three real values, each representing the time-to-compromise (TTC), in workdays (person-workdays), that a certain top percentage of the overall population of professional penetration testers would require to reach the attack step in question. The three real values represent the TTC for 5%, 50%, and 95% of the professional penetration tester population, respectively. Put simpler, only the most elite or lucky penetration testers would assumedly have succeeded reaching an attack step until the time given in the first value. Around half of the penetration testers would assumedly have reached the attack step by the time given in the second value.

Next to all of the penetration testers would assumedly have reached the attack step by the by the time given in the third value. This means that the lower the numbers, the worse the security. In the graphs showing the results in the sections below, the colors signify the following. Blue indicates that only between 0% and 5% of the attacker population would have succeeded, meaning rather good level of security. Green color indicates that between 5% and 50% of the attackers would have succeeded. Yellow color indicates that between 50% and 95% of the attackers would have succeeded. Finally, red color indicates that all attackers would have succeeded [by the time in question], meaning a high level of vulnerability.

In order to gain more insight in the ease to perform different types of attacks, each portion of results (e.g., countermeasure effectiveness in SCADA) includes three distinct set of TTC values, according to three different attack steps on a selection of hosts, as follows:

1. "Denial of service", which is the prediction of how long time it would take attackers to bring down the host (including all systems running on it).

2. "User access", which is the prediction of how long time it would take the attackers to gain userlevel privileges to a system (not necessarily administrator-level privileges).

3. "Malware expoitation", which is the prediction of how long time it would take the attackers to compromise the system through sophisticated attacks using data with malicious software payload.

1.5.5.2 Vulnerability analysis: Advanced metering infrastructure

Overall cyber security posture

The results show that three systems appear to be notably more exposed to denial of service attacks than others - those are the meter data management system (MDMS), its database running on a separate host, and finally AMI data concentrators placed out in the field. The other hosts show highly similar pattern, except that home appliances and smart meters residing at the customer premises (or immediately close to it), are more exposed to denial of service from the most capable attackers.

User access appears to be rather easily achievable on the MDMS, the advanced meter management system (AMMS) and the AMI front end. The demand-response analysis and control system (DRAACS) also appears exposed, although to a slightly lesser degree. The customer-level devices (smart meters and home appliances) appear to show the highest dependency on the capability of the attacker. These systems appear to be most difficult to gain user access on by the least capable attackers, however, they are very easy to gain access to by the highly capable ones.

This stands in contrast with the rest of the systems in the AMI infrastructure, in which the distribution between the TTC between the most and least capable attackers is far narrower. When it comes to malware exploitation, the prediction patterns from user access (above) is largely followed, except that MDMS and its database show a higher degree of vulnerability towards the least skilled attackers. This is presumably the case due to the system's high degree of interconnection with the enterprise IT systems residing in the utility's intranet and office environment.

Countermeasures

The results regarding the effectiveness of the countermeasures are depicted in figures 40 (for denial of service), 41 (for user access), and 42 (for malware exploitation). It can briefly be concluded that highly granular network segmentation and diligent patching of system are the most effective security countermeasures. Granular network segmentation is especially effective at protecting database systems, which can to a greater degree be isolated from the rest of the hosts, with which they do not normally need to interact.

Further, firewalls and host intrusion detection systems also appear to have slightly elevated

effectiveness at protecting systems, especially from malware exploitation. Finally, it has to be noted that in all scenarios, only additional protection of a certain type are considered compared to a baseline architecture that is already protected to a certain degree - rather than comparing a protected architecture with a completely unprotected one. This, among other, explains the rather small differences in results towards certain protections being added, such as data encryption (since much data communications are already typically encrypted in an AMI infrastructure.

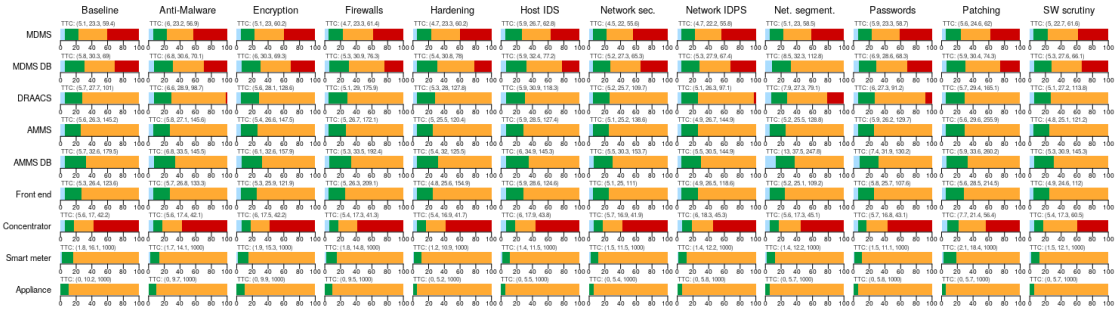

Figure 40: Overview of results from the evaluation of protection scenarios in AMI - denial of service.

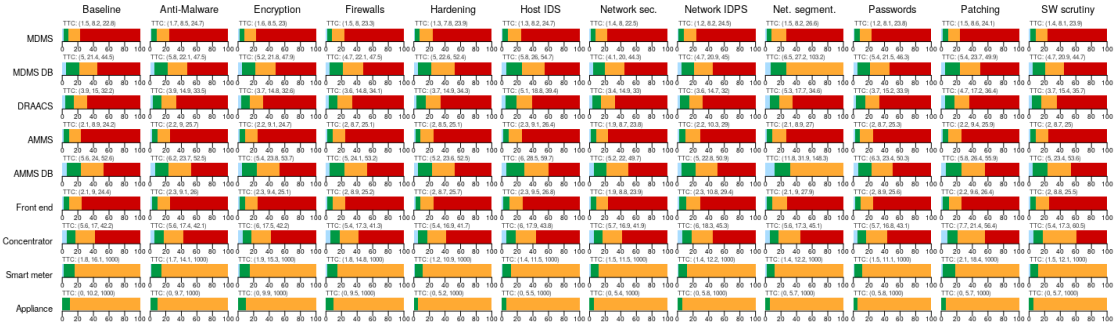

Figure 41: Overview of results from the evaluation of protection scenarios in AMI - user access.

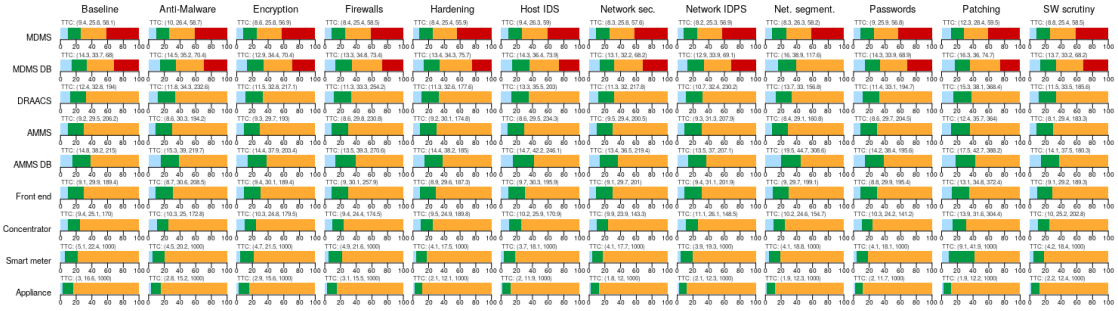

Figure 42: Overview of results from the evaluation of protection scenarios in AMI - malware exploitation.

Insider attacks

The results of evaluation of the impact from different placement of attackers is presented in figures 43 (denial of service), 44 (user access), and 45 (malware exploitation).

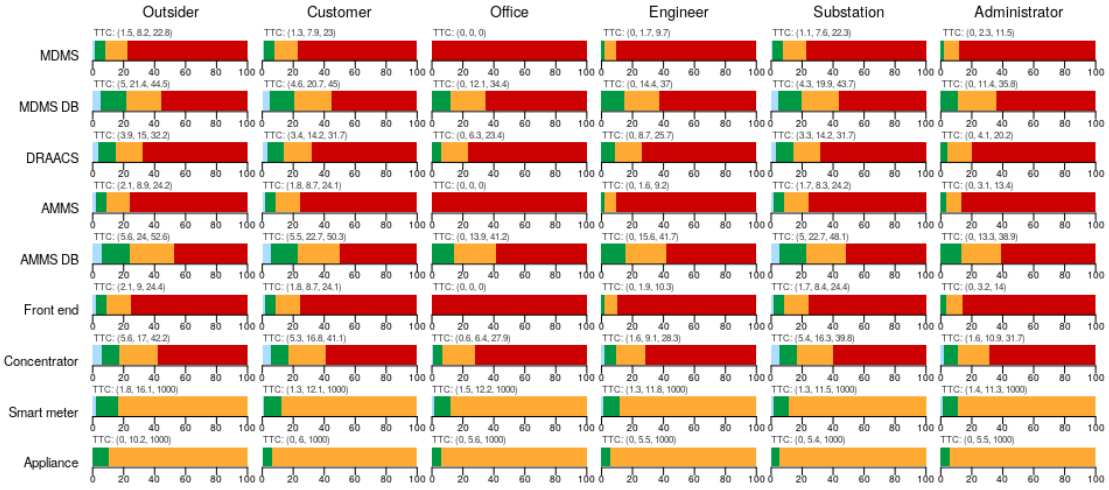

Figure 43: Overview of results from the evaluation of different attacker positions - denial of service inAMI.

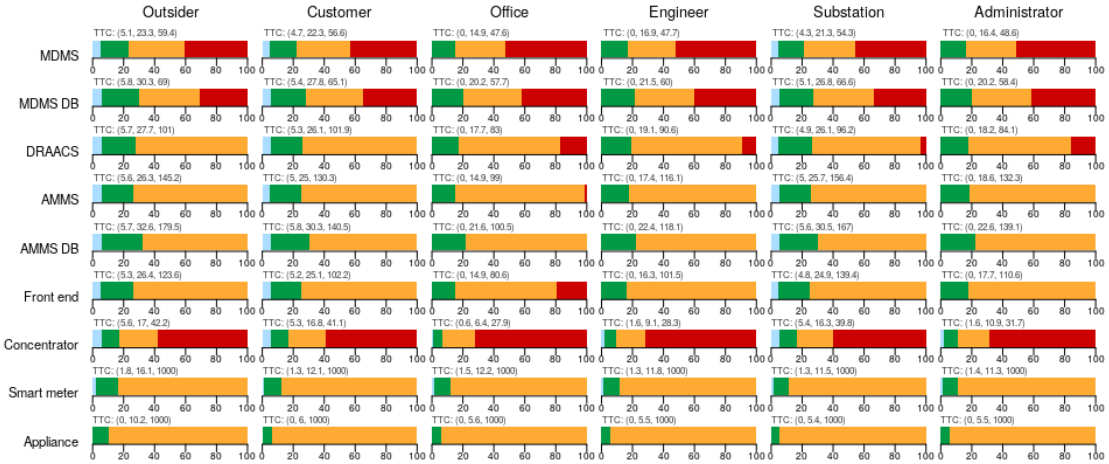

Figure 44: Overview of results from the evaluation of different attacker positions - user access in AMI.

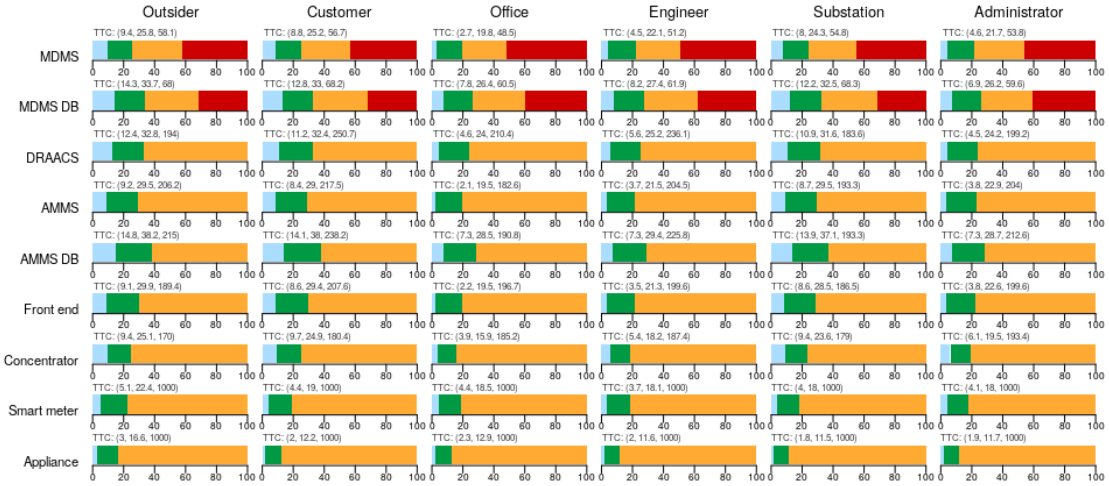

Figure 45: Overview of results from the evaluation of different attacker positions -malware exploitation in AMI.

1.5.5.3 Vulnerability analysis: SCADA infrastructure

Overall cyber security posture

The results indicate that the front end is the single most exposed component in the SCADA infrastructure, mostly owing to threats coming from the field wide area network (WAN) and substations. The front end appears to be especially vulnerable to denial-of-service attacks, compared to the other assets. The second most exposed asset is the SCADA server and then the SCADA application server (energy management system). The best protected assets appear to be the domain controllers.

Generally, the assets in the Scada DMZ network appear to be slightly less exposed than the ones in the SCADA network, which appears to be due to the high threat level coming from the field WAN and substations. The exposure of assets in both networks follows the same pattern.

Countermeasures

The results regarding the effectiveness of the countermeasures are depicted in figures 46 (for denial of service), 47 (for user access), and 48 (for malware exploitation). A brief analysis in text can be found below the figures.

Figure 46: Overview of results from the evaluation of protection scenarios in SCADA - denial of service.

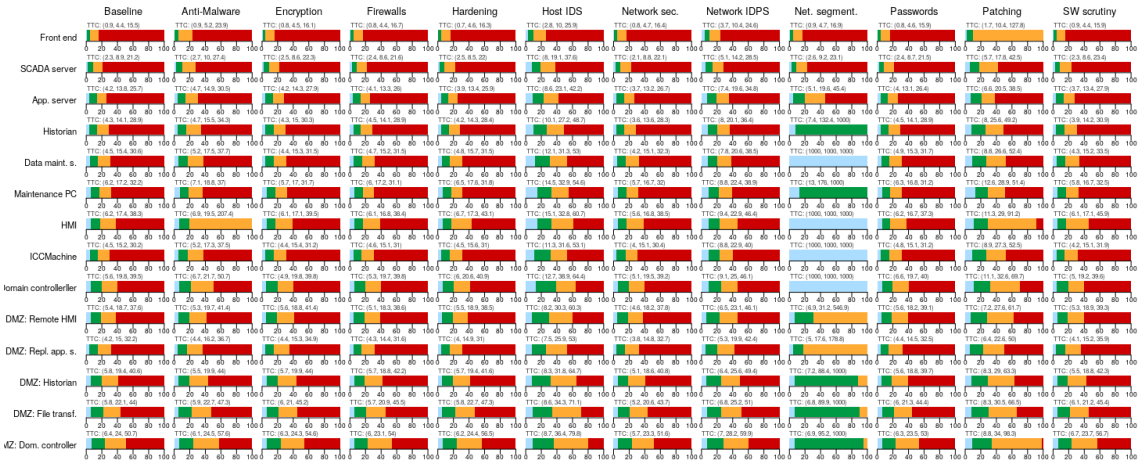

Figure 47: Overview of results from the evaluation of protection scenarios in SCADA - user access.

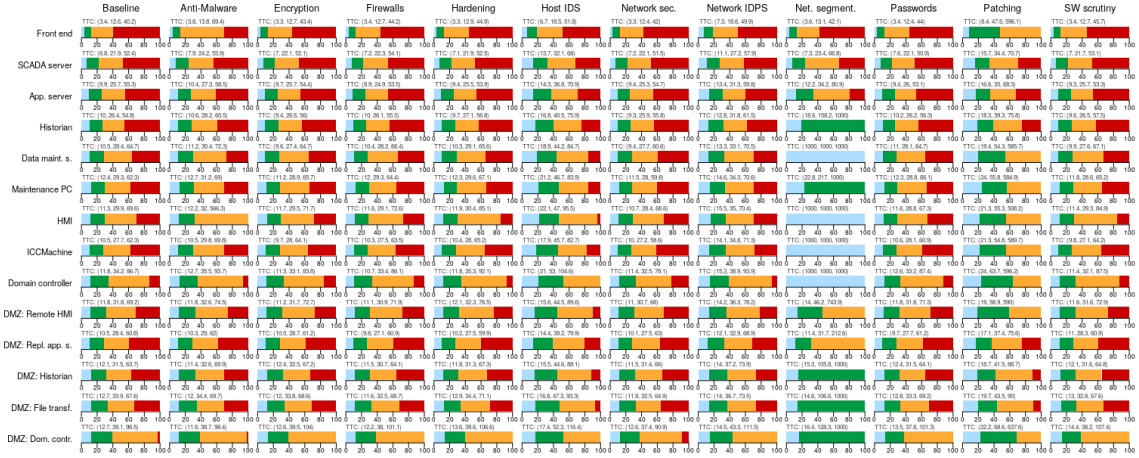

Figure 48: Overview of results from the evaluation of protection scenarios in SCADA - malware exploitation.

The results of comparing protection strategies show a somewhat varied picture for the different assets. While it can be said that network segmentation is the clear winner on overall, followed by diligent systems patching and host-level intrusion detection; certain assets show a different pattern. The results suggest that diligent patching is the most effective way of securing the front end, however, the the suitability of different protection strategies even depends on what type of attacks are to be guarded against. For example, network intrusion detection and prevention appears to protect well against highly skilled attackers using malware expoitation, while using an anti-malware solution appears to make it difficult for the least skilled attackers to succeed - even to a higher degree than the former protection strategy does.

According to the results, granular network segmentation is especially highly effective at protecting the SCADA HMI, the data maintenance (data engineering) server, the ICC machine and the domain controller in the SCADA network. Since segmenting the networks more granularly tends to be inexpensive compared to e.g. diligent systems patching and all testing implied; its choice should be out of question.

Insider attacks

The results of evaluation of the impact from different placement of attackers is presented in figures 49 (denial of service), 50 (user access), and 51 (malware exploitation).

Figure 49: Overview of results from the evaluation of different attacker positions - denial of service in SCADA.

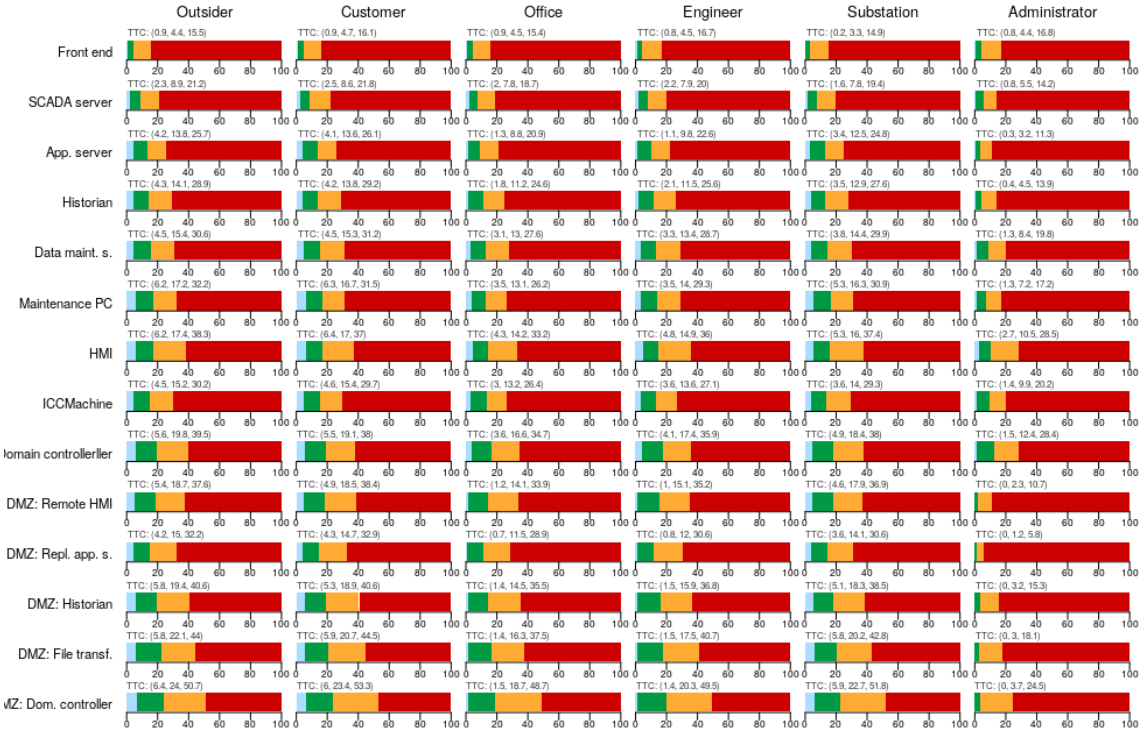

Figure 50: Overview of results from the evaluation of different attacker positions - user access in SCADA.

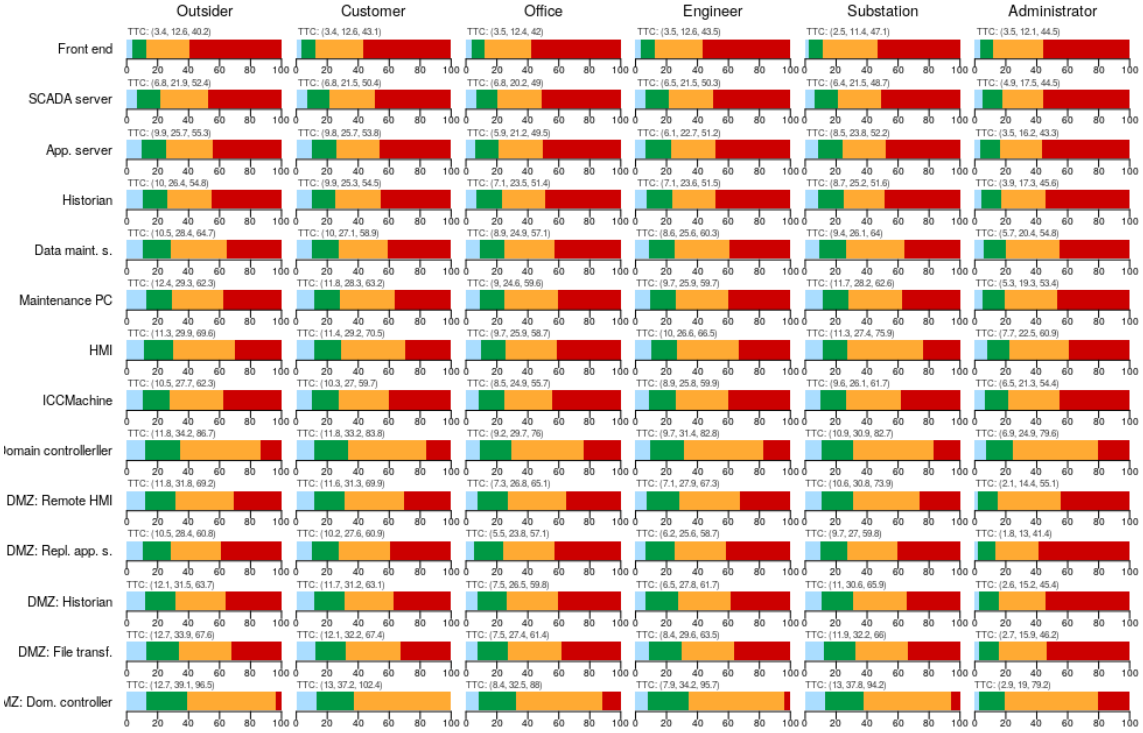

Figure 51: Overview of results from the evaluation of different attacker positions - malwareexploitation in SCADA.

1.5.5.4 Vulnerability analysis: Substation automation infrastructure

Overall cyber security posture

The results suggest that the substation [typically] is the single most exposed part of the entire IT environment of a DSO. One of the reasons is the presence of modem-based access to substations from the public Internet, in order for system vendors to be able to access the substations and perform equipment maintenance. It appears, unfortunately, not uncommon to find default or very weak login credentials (usernames and passwords) on the substation equipment, sometimes including the modem and/or VPN that allow access from the public Internet. The presence of such a situation implies a monstrous security hole into the entire architecture, the dangers of which overstep the boundaries of the substation itself, and even highly threaten the security inside centralized SCADA networks, as can be read from the results from evaluation of the different reference models put in a single large model of a DSO's IT environment.

From among the assets in the substation, the most exposed ones appear to be the workstation, the substation-level control system, and the remote terminal unit (RTU) device; as the malicious influence most likely spreads through the modem or VPN and thereby gaining full network access to the substation control LAN. This exposure pattern, however, seems to only apply to denial-of-service attacks and unauthorized user access. For malware exploitation, specialized embedded devices such as RTUs and IEDs appear to be an easier target for most of the attacker population; while the most capable attackers seem to have a rather easy way infecting any of the devices (around two manworkdays according to the results).

The entries (hosts) in the figures showing results, below, are shown from two separate substations. Their name labels (on the vertical axis) are prefixed by [S1] for one substation and [S2] for the other, which additionally features a connection to a distributed energy resource such as a wind turbine, or a wind turbine park.

Countermeasures

The results regarding the effectiveness of the countermeasures are depicted in figures 52 (for denial of service), 53 (for user access), and 54 (for malware exploitation). A brief analysis in text can be found below the figures.

Figure 52: Overview of results from the evaluation of protection scenarios in substation automation - denial of service.

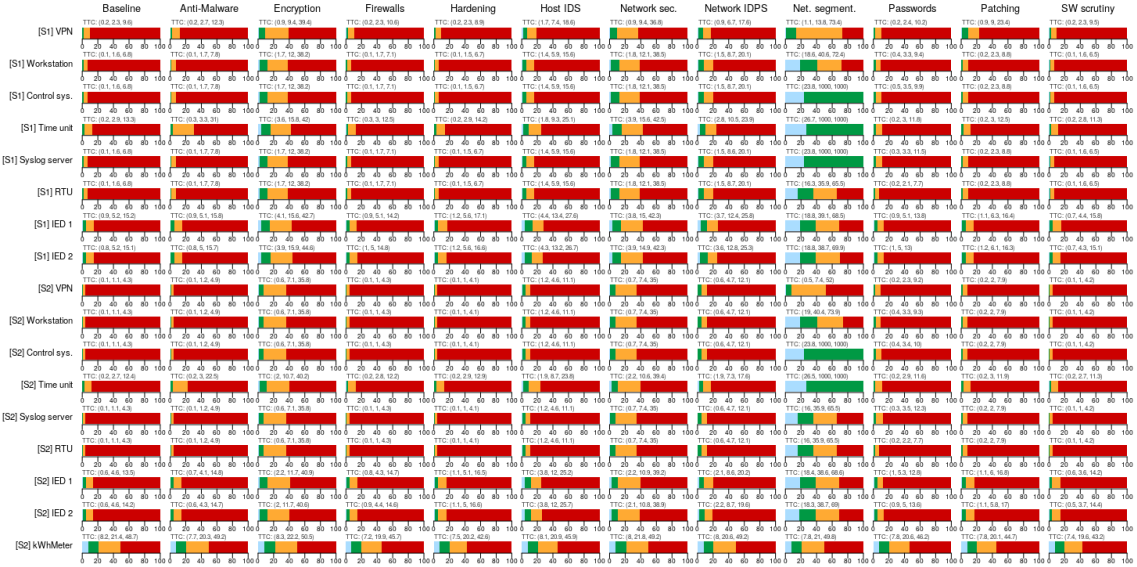

Figure 53: Overview of results from the evaluation of protection scenarios in substation automation - user access.

Figure 54: Overview of results from the evaluation of protection scenarios in substation automation - malware exploitation.

Similarly to the case in the SCADA network, granular network segmentation appears to be the most effective protection strategy by far. Furthermore, strict network configuration and data communications encryption also show a difference in terms of security, especially for malware exploitation, however, not as drastic one as granular network segmentation. Diligent systems patching appears to be similarly effective against malware exploitation, however both network intrusion detection and prevention, and host-based intrusion detection, surpass its effectiveness for user access and denial-of-service attacks; as they indeed do for any attack mode in case of the most capable attackers.

Lastly, the enforcement of password policy appears to be an effective protection strategy against denial-of-service attacks on the VPN, substation control system and workstation, second only to granular network segmentation.

Insider attacks

The results of evaluation of the impact from different placement of attackers is presented in figures 55 (denial of service), 56 (user access), and 57 (malware exploitation).

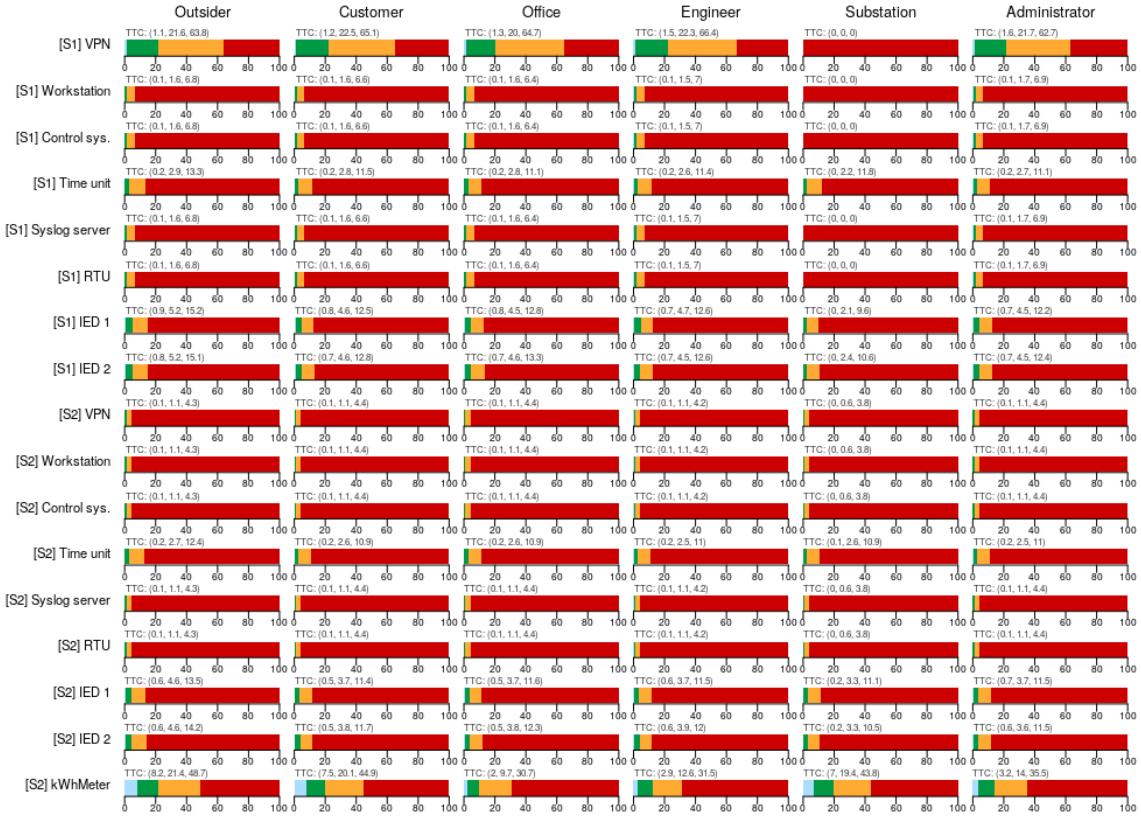

Figure 55: Overview of results from the evaluation of different attacker positions - denial of service in substation automation.

Figure 56: Overview of results from the evaluation of different attacker positions - user access in substation automation.

Figure 57: Overview of results from the evaluation of different attacker positions - malwareexploitation in substation automation.

1.5.5.5 Vulnerability analysis: Distributed energy resource

Overall cyber security posture

For a home setup (e.g., considering a household-grade solar panel or similar DER device), the level of security appears to be fair, keeping most of the less capable attackers out, according to the results (e.g., see the results for smart appliance, termed "Appliance", in the results of the advanced metering infrastructure). The setup of a household smart appliance is assumed to be similar to the setup of a household-grade DER.

For an industrial setup (e.g., considering a DER connected to a substation), the level of security appears to be rather poor, on par with the embedded substation automation equipment, namely IEDs (intelligent electronic devices). Interestingly however, the most capable attackers appear to have a similar chance of infecting the household-grade DER with malware, and even a greater chance of gaining unauthorized access to its control interface.

Countermeasures

Due to the simplicity of the IT infrastructure of the distributed energy resource itself, the effectiveness of countermeasures were not evaluated. Instead, our recommendation is to follow the conclusions from the other reference models alongside with up-to-date best practice for IT security.

Insider attacks

In both cases, the industrial-grade and household-grade DER setup, local insider attacks (in the substation and in the household, respectively), will lead to higher exposure of the DER unit and its operation. However, the difference is rather subtle, thirty per cent (30 %) and less shorter times to compromise.

1.5.5.6 Vulnerability analysis: Enterprise and office IT environments

Overall cyber security posture

According to the results, the web server in the public DMZ network, the engineer's PC, and several intranet servers - enterprise resource planning system (ERP), customer management system (CMS) and workforce management system (WFM) - are the ones most exposed to a compromise through unauthorized user access. The most capable attackers could compromise these systems in around (or slightly less than) two person-workdays.

Denial-of-service attacks seem to be most threating to the web server and customer portal in the public DMZ, as well as the Internet proxy. Finally, the exposure to malware exploitation is more notable for the Internet proxy - especially from the less capable attackers. The most capable attackers seem to be able to infect the web server, the engineer's PC and the three above mentioned intranet systems at easiest; however, it would still take them around 8-10 person-workdays.

Countermeasures

The results regarding the effectiveness of the countermeasures are depicted in figures 58 (for denial of service), 59 (for user access), and 60 (for malware exploitation). A brief analysis in text can be found below the figures.

Figure 58: Overview of results from the evaluation of protection scenarios in the enterprise IT infrastructure - denial of service.

Figure 59: Overview of results from the evaluation of protection scenarios in the enterprise IT infrastructure - user access.

Figure 60: Overview of results from the evaluation of protection scenarios in the enterprise IT infrastructure - malware exploitation.

The results indicate that granular network segmentation is most effective at some protecting assets in the public DMZ (web server, customer portal, DMZ database, and FTP server) and the intranet (data warehouse and file server), however largely falls short of providing effective protection to the most exposed systems in the intranet, like ERP, CMS and WFM. A highly effective protection strategy in all cases, actually surpassing granular network segmentation in most cases, is the use of host intrusion detection. Host intrusion detection appears especially effective against the most capable attackers. Network intrusion detection and prevention comes at the third place for most systems.

Regarding malware exploitation, diligent systems patching appears to provide protection comparable with host intrusion detection and network intrusion detection and prevention.

Insider attacks

The results of evaluation of the impact from different placement of attackers is presented in figures 61 (denial of service), 62 (user access), and 63 (malware exploitation).

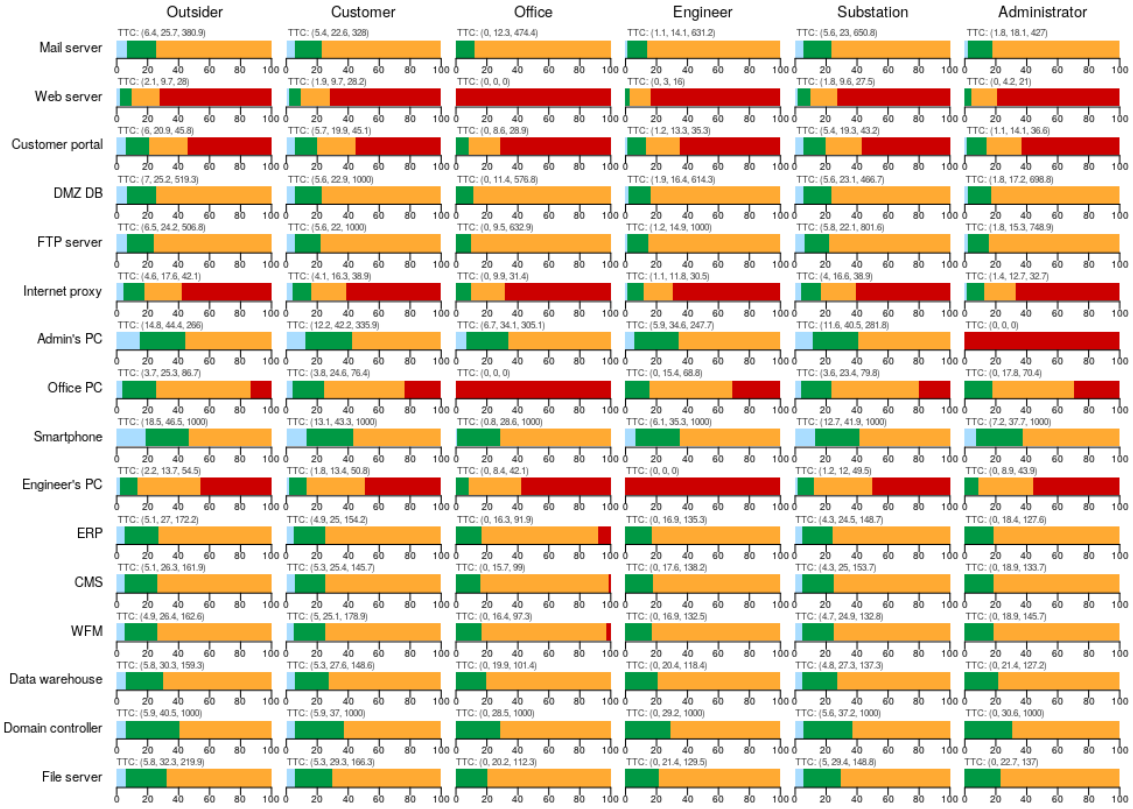

Figure 61: Overview of results from the evaluation of different attacker placement - denial of service in enterprise IT infrastructure.

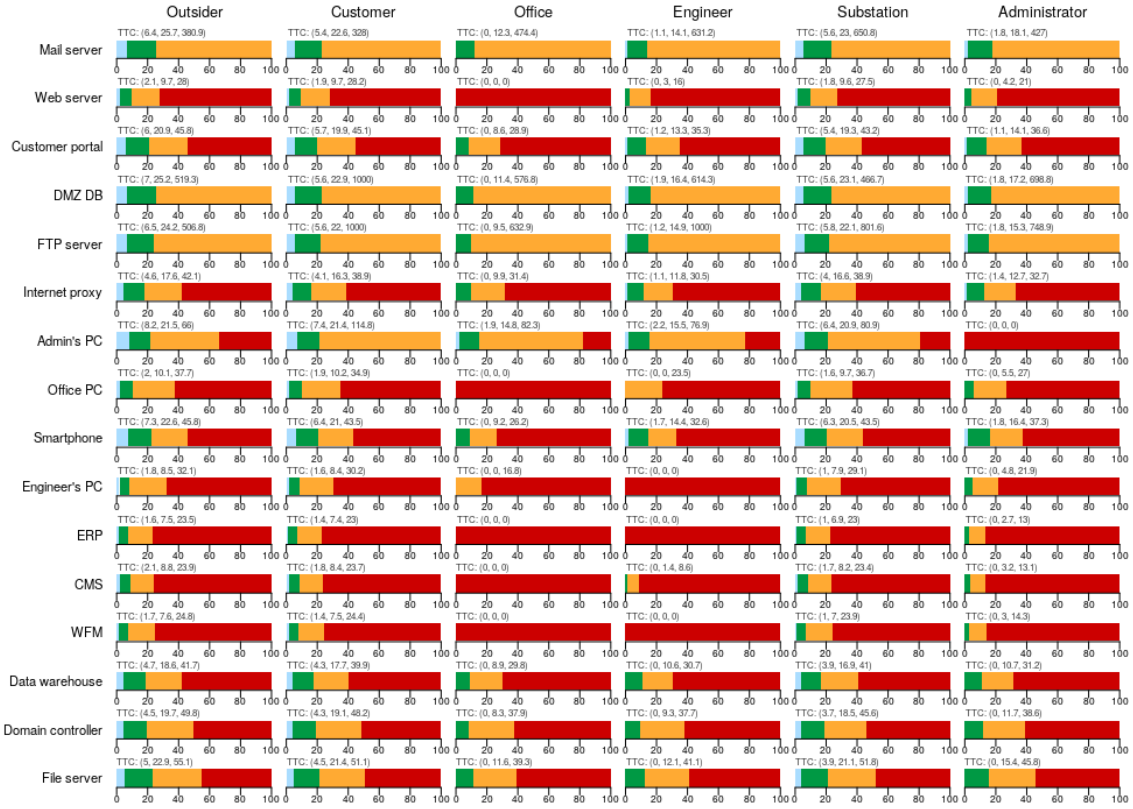

Figure 62: Overview of results from the evaluation of different attacker placement - user access in enterprise IT infrastructure.

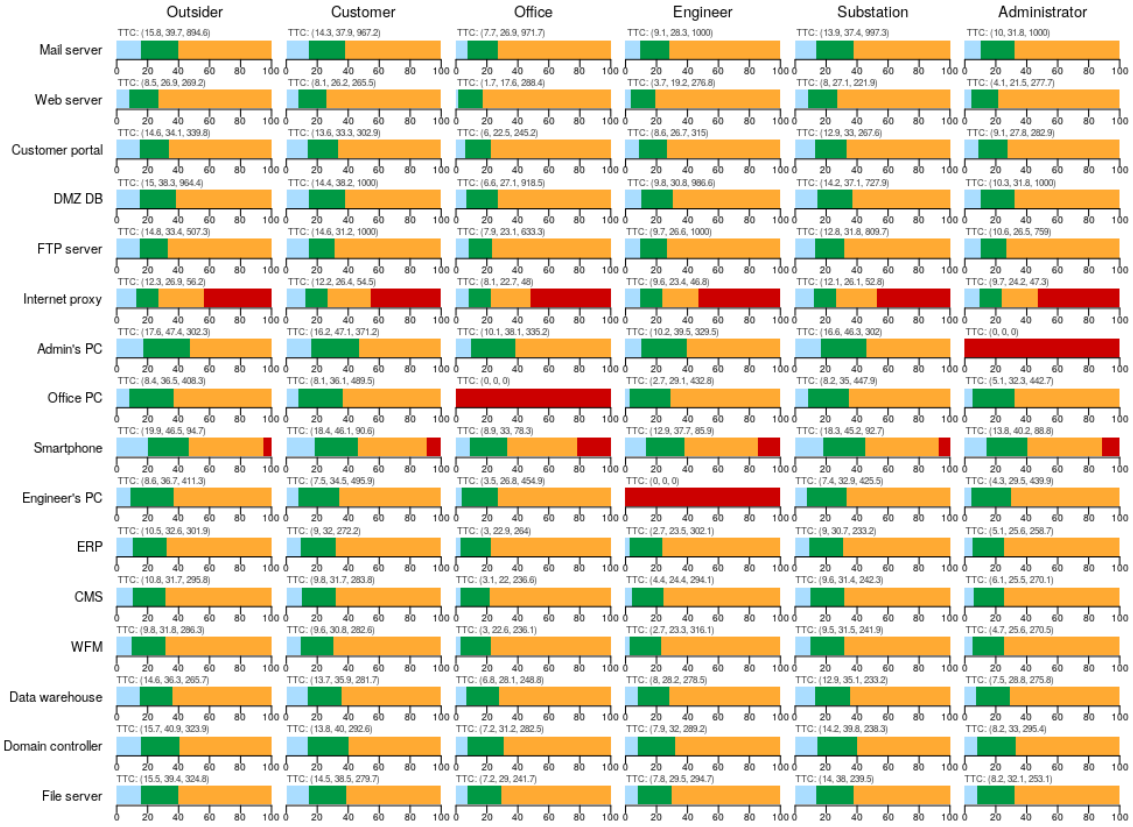

Figure 63: Overview of results from the evaluation of different attacker placement - malware exploitation in enterprise IT infrastructure.

1.5.5.7 Discussion

A vital element in correctly interpreting and understanding the results is the perspective from which the IT architectures were modeled and evaluated. Specifically, all the architectures were captured and modeled with a generally assumable amount of protection countermeasures already implemented and active. Based on this fact, valid conclusions can not be drawn about the total or absolute effectiveness of a certain type of protection (or protection strategy). Rather, these protection strategies only represent additional protection through the respective protection strategy. For example, if the underlying architecture (e.g., advanced metering infrastructure) already typically has its servers and other assets hardened, additional hardening yields little additional security. Hence, the protection strategy may appear inefficient from the results presented in this report; but such a conclusion would be invalid, since the results only indicate the effectiveness of additional protection through that protection strategy from a baseline scenario, which already has a certain degree of that type of protection in place. In order to provide insights regarding absolute protection potentials, the method would have to be different and the architectures would have to be modeled completely devoid of any protection countermeasures (a highly unrealistic case), and then compared with scenarios protected in the different ways.

Another issue relates to the effects of synergy between and among different protection strategies used in a complementary fashion. Such a protection would normally be the case (as is already reflected in the baseline scenario); however, the additional complexity and time demands of the evaluation process would be vast, since all possible combinations of the protection strategy scenarios would need to be modeled and evaluated for each reference model. Due to these demands, combinations of protection strategies and the synergies between these, were not studied. A previous study about the SCADA infrastructure [36] suggests that synergies from using multiple protection strategies at the same time, exist.

Finally, it needs to be mentioned that the calculation of the time-to-compromise values across the architectures, was an automated process subject to any and all limitations of the methods and tools used. Due to such limitations, for example, insider attacks were not modeled as carried out by actual malicious employees of the DSO. Rather, the insider attacks were modeled as an outside attacker having gained access to an insider's computer (however not the knowledge nor necessarily the insider's access credentials to other systems). Consequentially, the threat stemming from an administrator's computer being compromised, is fairly low compared to what many might expect. Having considered a malicious administrator with all credentials to the systems entrusted to him or her, would clearly have resulted in a much worse degree of security than what the results show. Fortunately, such scenarios are rather straight-forward to infer from the calculated results in case that type of analysis was of interest.

1.5.6 Model-based intrusion detection

Intrusion detection systems (IDS) discern observable events reflected in available data and metrics into information about possible cyber-attacks. In physical infrastructures, computational models of the system's behavior are often used for decision and control purposes, and are thus often readily available.

The idea of model-based cyber-physical intrusion detection aims to integrate such behavior models into a framework for detection of external interference (with assumed malicious intent). In this section we identify what type of information and knowledge can be inferred from behavioral representations of a system and how such models should be integrated support intrusion detection.

1.5.6.1 Intrusion detection

Intrusion detection systems (IDS), in computer science, gather and analyze the information from a computer network or system activities in order to discover malicious activities or violations of policy. IDS use one of two detection techniques:

- Statistical anomaly based IDS - where the anomalies are detected by comparing the system or the network behavior with the established baseline behavior. The baseline behavior is compared to a reference behavior. Intrusion is defined as an anomaly or a significant deviation from the baseline.

- Signature-based IDS - where known intrusions or attacks are recorded or defined using a signature, and the network or the system behavior is compared to these signatures in order to find a match to well-known malicious behavior and intrusions.

Anomaly based intrusion detection is explored for malicious control of DER in power system. The complex physical behavior of the unit ('normal' behavior) is obtained from the DER model and compared to the observed behavior in order to detect anomalies. This way the IDS can divide the observed behavior of the unit into 'normal' and 'suspicious'. In order to detect the reason of the suspicious behavior, signature-based IDS can be used to detect for example device failures.

Anomaly detection applied to physical behavior models

In order to verify if a physical device connected to the power grid is being controlled by an unauthorized entity, its behavior can be observed and classified into four categories:

- Normal operation: The unit behaves as expected and it is not controlled by any external setpoint.

- Faulty operation: The unit's operation is disturbed by a fault on the unit or in its electrical network environment.

- Verified control: The unit behaves as expected under a verified control regime, or according to issued setpoints.

- Malicious control: The unit is operated under an unverified control regime or unauthorized setpoints.

This proprosed categorization of unit behavior can be included in the process of detecting an intrusion affecting the operation of a DER. In order to discover anomalous behavior of the system, the "normal" behavior needs to be described as a reference. When detecting anomalies in the behavior of a DER, for example their power production or consumption, its physical model can be used to define their "normal" behavior.

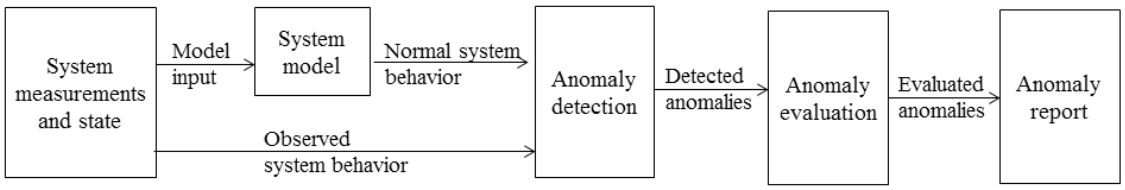

Figure 64: Anomaly detection based on physical behavior models.

The proposed IDS uses an anomaly detection method where the model output is compared against measured real-time data, and the difference is analyzed, as shown in Figure 64. In order to perform anomaly detection, the normal behavior of the DER needs to be defined as a DER model. This model takes measurements and DER state and output data associated with the normal behavior (for example an active power value of a power producer). The difference between the normal and observed behavior is weighted in order to detect anomalies. Subsequently, the anomalies are evaluated and the IDS generates a report about anomalies discovered.

1.5.6.2 Intrusion detection in the SALVAGE context

The work on intrusion detection in the context of the SALVAGE project has been motivated by the following observations:

- Cyber-attacks aimed at disrupting system operation may manifest themselves to the observer in much the same way as system faults, such as normal component failures. A successful cyberattack will cause the system behavior to deviate from the desired or expected behavior.

- Cyber-attacks may be designed to cause system failure because that is the desired destructive behavior or to simply emulate behaviors, for cloaking or detection avoidance; both strategies, for example, have been employed in case of Stuxnet [37].

- The ability to correctly distinguish between failure and attack post-mortem is just as important as the ability to prevent attacks in the first place; a recognized and analyzed attack pattern enables prevention of attack repetition and elimination of attack vectors.

- The engineering effort available for develop accurate simulation models for an IDS facilitating cyber-attack prevention at distribution level is large. Practical IDS development and operation should therefore be largely independent of accurate physical model of the investigated physical system.

The first two observations motivate the use of behavioral models representing the expected cyberphysical system behavior for reference and on-line detection; the latter observations however also encourage an engineering strategy for IDS development that integrates well with other engineering and business work flows and focuses on a close interaction with available data sources (a data-driven engineering approach).

Howeverm, a purely data-driven approach cannot be sufficient to discriminate between the various types of attack vectors. In addition to behavioral representations, also a qualitative framework is required to hypothesize and evaluate alternative attack scenarios.

Intrusion detection in the SALVAGE project has therefore been seen as one part of a two-layered approach as illustrated in Figure 65. The analysis of measurements and integration of cyber-physical behavioral models aims toward anomaly detection; the interpretation of observed anomalies is then passed to another analysis layer aimed at intrusion detection, which integrates anomaly information with knowledge about expected system behaviors (for example control signals, current system configurations, operating modes and procedures).

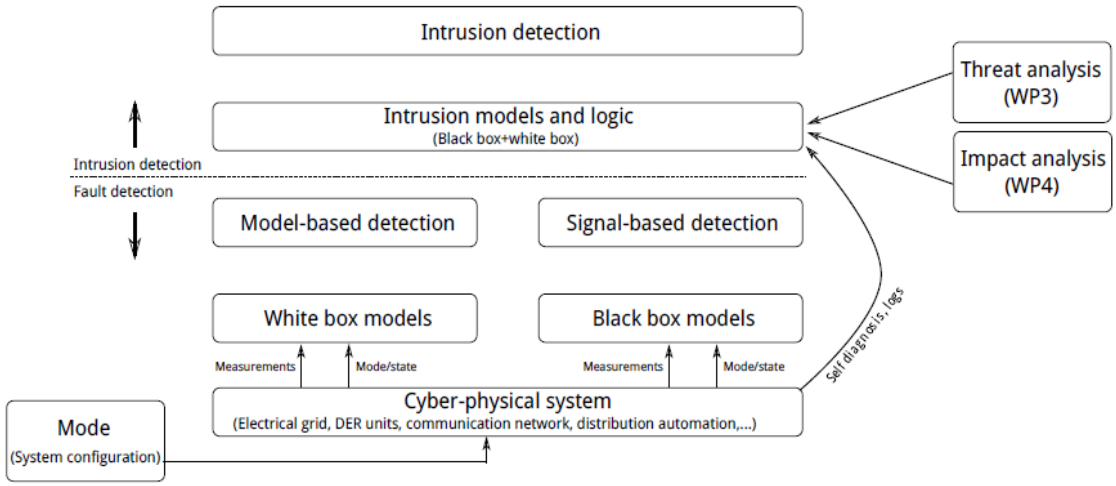

Figure 65: Conceptual diagram of the SALVAGE intrusion detection framework.

Behavioral models represent a previously observed and explained behavior; in case of cyber-physical models, this behavior is typically the dynamic behavior of a physical system which is under some control influence. This behavior can be employed serve as a normative reference ("normal behavior"), and in this way they may be employed as a normative reference for cross-validating measurements to detect deviations from the modeled outputs as behavior anomalies.

Whereas such models represent physical and controlled system behavior well, they contain no information about possible intrusion pathways, thus they may only serve to detect behavior anomalies, but do not identify the "intent" in the behavior.

Detection scenario

We assume the proposed intrusion detection system to be situated within the SALVAGE distribution network scenario outlined in section 1.5.2.1. This scenario considers a low voltage distribution grid, as presented in Figure 66, with buildings and PVs connected to the transformer (Tr1). In this report we are focusing on the independent components and representing them with models, therefore we are considering cybersecurity aspects of buildings and PVs.

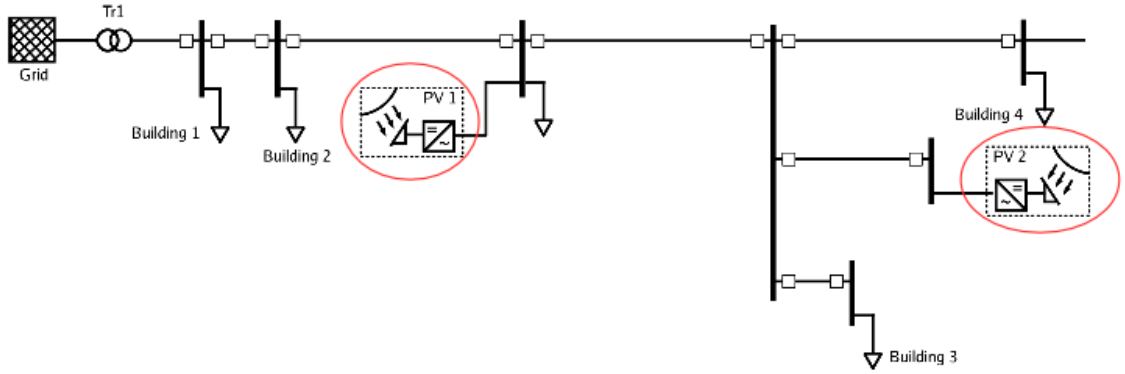

Figure 66: PowerCap scenario, with PV inverters highlighted.

The DER components considered in this report can be externally controlled, for example from the Aggregator. As shown in figure 67, the interaction (marked with a red dot) refers to sending setpoints from the Aggregator to the DER unit. The detection scenario considers an

intruder overwriting the setting originating at the Aggregator or sending an independent setpoint to the unit.

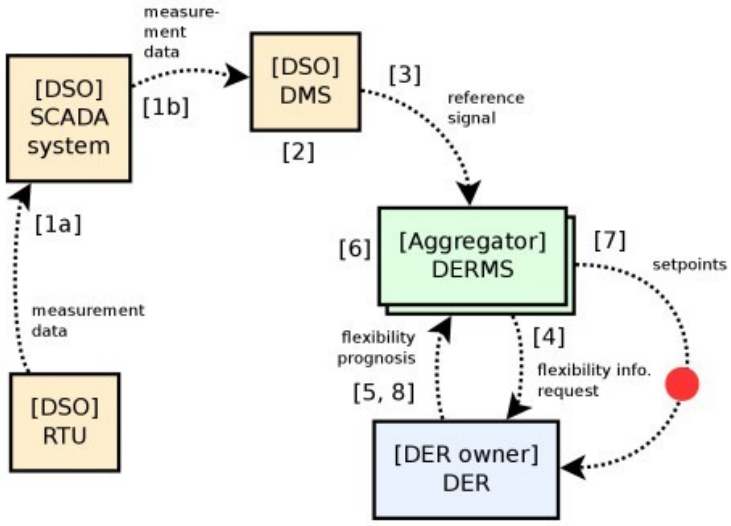

Figure 67: Interaction between DER and Aggregator in the considered scenario

1.5.6.3 Intrusion model for a DER component

The project has investigated anomaly-based intrusion detection with the use of models applied to individual power system components, or DERs. For this purpose, intrusion models for DER components were developed within the project. The existing models are described, categorized and evaluated by type, complexity of modeling tasks, behavior modes, available redundant input, relevance for and impact on the grid, and usefulness for the selected SALVAGE scenario. We present applications of data driven modeling, its advantages and disadvantages.

Existing models

Figure 68 presents a list of selected power system components, demand response units and grid models which had been developed at DTU prior to the SALVAGE project, which had been validated against measurements of physical components in DTU's SYSLAB laboratory, and which were available to the project. They can be grouped into the following categories:

- Photovoltaic system including grid-tie inverter
- Battery storage system
- Distribution grid
- Demand response building - heating system and thermal building model
- Demand response appliances

No	Device type	Description	Input		Output	Analytical/ Computational	Quality			Time resolution	Programming language	Link/publication
			Controllable	Uncontrollable			RMSE	variance	validated			
1	House	Flexhouse1heating single room model, discrete space state	Heat power [W]	Solar rad [W/m2], External temperature [C]	Room temperature [C]	Analytical	na	na	yes		R	
2	House	Flexhouse1heating single room linear model, discrete space state	Heat power [W]	Solar rad [W/m2], External temperature [C]	Room temperature [C]	Analytical	na	na	yes		Matlab, Java	
3	House	Flexhouse1heating single room linear model, discrete space state	Heat power [W]	Solar rad [W/m2], External temperature [C]	Room temperature [C]	Analytical	na	na	yes		R	
4	House	Flexhouse1heating single room non-linear model, discrete space state	Heat power [W]	Solar rad [W/m2], External temperature [C], wind speed [m/s]	Room temperature [C]	Analytical	na	na	yes		R	
5	House	Flexhouse 1 heating multiple (8) room model with 10 heathers	10xHeat power [kW]	Solar rad [W/m2], External temperature [C]	8xRoom temperature [C]	Analytical	na	na	yes	variable	Matlab, Python	
6	House	house heating model, adjustable time step and hose size configuration	Ambient temperature [C], solar rad W/m2] wind speed [m/s]		P, P1-3, Q, Q1-3, Frequency, Voltage, Current, house state, wind speed, wind direction, solar read	Analytical				1 sec	Java	risoe.syslab.control.resemu.model.HouseSimulator in syslab pvsec2012
7	PV	SYSLAB PV 319			power [kW], theoretical Q, 5 min prediction of P and Q	Analytical					Java	risoe.syslab.control.experiment.pvsec12.PVPotential in syslab pvsec2012
8	Grid model										PowerFactory/ Matlab	
9	Battery	SYSLAB Vanadium battery	voltage of the battery [Volt] SOC[%]	external temperature [C]	P,Q	Analytical	available, to be computed	available, to be computed	yes	1 sec	Matlab/ Simulink	http://www.sciencedirect.com/science/article/pii/S0378775313020570
10	Fridge	single state model	electric power [W]	room temperature [C]	refrigerator temperature [C]	A	available, to be computed	available, to be computed	yes	Variable (1-10 min)	Matlab	
11	House	single room thermal model	cooling power[W]	external temperature[C] solar radiation [W/m2]	internal room temperature[C]	Computational	available, to be computed	available, to be computed	yes	Variable (1-5 min) depend of the input data resolution	Python	
12	PV	SYSLAB PV 319		solar radiation[W/m2] outdoor temperature[C], wind speed [m/s], time [h]	power[W]	Computational	available, to be computed	available, to be computed	yes	Variable (1-5 min) depend of the input data resolution	Matlab	
13	Washing machine	Kitchen 776	start/stop, load[full load, half-load, small load]		power [W]	Computational			not (time series of real consumption)	Variable (min 1 sec)	Simulink	
14	Microwave	Kitchen 776	start/stop, load[full load, half-load, small load]		power [W]	Computational			not (time series of real consumption)	Variable (min 1 sec)	Simulink	
15	PV	PV319	Ambient temperature [C], solar rad W/m2] wind speed [m/s]		power [kW], theoretical Q, 5 min prediction of P and Q	Analytical	available, to be computed	available, to be computed	yes	1 sec	Matlab/ Simulink	ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6715023

Figure 68: Overview of available DER models at DTU

The following evaluation criteria were applied to the existing models, in order to determine their relevance to the SALVAGE project:

- Complexity of modeling tasks: How complex is the component model, what is the effort to create an individual model and a set of models representing the same DER type?

- Behavior modes: Ability of the model to recognize different types of behavior, for example normal behavior of an occupied or unoccupied house.

- Available redundant input: Does the model use redundant data in order to eliminate the anomalies created by faulty sensor readings?

- Relevance and impact on the distribution grid: How large is the impact of the malicious control or behavior of the modeled unit?

- Usefulness for scenario: Does the modeled unit appear in one of the SALVAGE scenarios?

Figure 69 presents the evaluation of the models listed in figure 68 and forms the basis for the final model selection.

DER type Evaluation	PV	Battery	Grid	House	Fridge	Washing machine	Microwave
Complexity of modeling tasks	Complex	Complex	Complex	Complex	Simple	Simple	Simple
Behavior modes	No	No	No	No	No	No	No
Available redundant input	No	No	No	No	No	No	No
Relevance and impact on the distribution grid	Relevant	Relevant	Very relevant	Relevant	Low impact	Low impact	Low impact
Use for scenario	Scenario 1	Not considered	Scenario 1	Scenario 1	Not considered	Not considered	Not considered

Figure 69: Evaluation of existing DER models.

Based on the above evaluation, models of PV, battery, grid and house/building are the most relevant models, considering the impact of the malicious operation on the distribution grid. Only PV, grid and house models are relevant for the chosen SALVAGE scenario. Two of these models are DER models: PV and house. The PV model was chosen for the first testing of intrusion detection system, as its operation only depends on the environmental conditions and external control signals. The house is a much more complicated case as it also includes human interaction which is not easily modeled and predicted.

The existing PV system model had been developed based on data from just one of the three PV systems available in the SYSLAB laboratory. This PV system had been equipped with dedicated instrumentation as part of a detailed measurement campaign in another project, in order to create a very accurate model for PV use optimization. Unfortunately, this PV model turned out to be very specific to the one installation it was derived from, and cannot necessarily be assumed to be valid for other installations. In order to ensure broader applicability of the IDS model, the SALVAGE project investigated as an alternative the use of data driven methods and machine learning to construct a more general PV model based on observed historical data.

Data-driven modeling

Data driven modeling refers to modeling efforts based on historical observations of the considered system. This type of modeling can be used for black-box modeling, where only input and output of the system is observable, and the internal workings are unknown. Data-driven modeling is common in data mining and machine learning. Machine learning is a method of programming computers to act in a way that have not been explicitly programmed. Data mining is a computational process aiming at discovering patterns in data. The advantage of using data driven modeling is its flexibility: this computational method can identify the observed unit without a prior knowledge of its inner workings. Models can be created from historical observations of any system. Even unknown patters and complex phenomena can be discovered in the data and expressed in the model. Model can be

calculate automatically and updated if the system behavior changes. Disadvantages include a large computational effort to generate the model, the model quality depends on the data quality and generality of the model depends on the statistical properties of the data and the data size.

All data used for the PV models has been recorded from three laboratory PV systems (7, 10 and 10kWp), their grid connection points and one meteorological station, all part of the SYSLAB laboratory (figure 70) at DTU Risø campus. SYSLAB [38] is a research facility for intelligent, active and distributed power systems targeted at research and testing of control concepts and strategies for power systems with distributed control.

Figure 70: The SYSLAB facility at DTU Risø campus.

SYSLAB's SCADA system provides measurement data at a base resolution of 1 second, covering ca. 30 parameters for each DER point of common coupling and ca. 40 parameters per PV system. The entire facility has been running continuously for several years, providing researchers with long timeseries of DER operation.

1.5.6.4 PV modeling strategy

The raw data from the laboratory must be prepared before it can be used for modeling. The stages of the chosen data driven model development are outlined in figure 71.

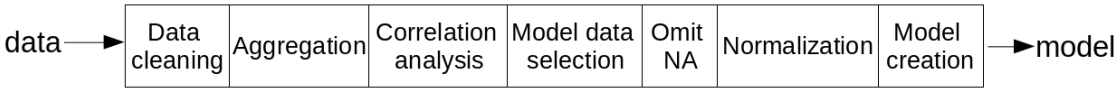

Figure 71: Data driven modeling strategy for ANN models.

The data cleaning, preparation and pre-processing stages are as follows:

1. Data cleaning: There can be many errors appearing in the raw sensor data. The process of cleaning the data starts form filling missing time series with NA values, so all time-series for a day consist of exactly 86400 data points. The next step is discovering if the time-series have unexpected values. The threshold between realistic and unrealistic values can be determined manually. For example, a temperature of 50°C is an unrealistic value for outdoor temperature in Denmark and can be discarded. Unrealistic values are replaced with Not-a-number (NaN). If the number of unrealistic values is large the sensor might have been broken. In this case we use signal processing to remove the noise form the data and replace it with NaN.

2. Aggregation: Once the data is clean and uniform size it can be aggregated. One second values are aggregated to 1 minute values, reducing the size of time-series to

1440 samples. This process is performed in order to save processing time for model creation, while not losing much of information. The aggregation is done omitting NA and with mean value for samples.

3. Correlation analysis: The correlation of model output and input is calculated. If the correlation is very small (correlation<0.2) the entire day is removed from the data. This step eliminates data with standard deviation equal to zero (which in unrealistic for sensor data) which are periods with a long sensor failures. The statistical significance is not considered here.

4. Model data selection: The days secreted by the previous step have been used for the model creation. The timestamps are removed and data samples are grouped in vectors, where sample I is as follows:
$sample^{(i)}=\left(input_1^{(i)},input_2^{(i)},\ldots,input_n^{(i)},output^{(i)}\right)$.

5. Omit NaN: All samples where at least one value in the vector $sample^{(i)}$ is equal to NaN are omitted, as the used modeling method, ANN, does not accept vectors with NaN values. The set of observations $S=sample^1,sample^2,\ldots,sample^k$ is divided into three groups of random samples: The training set S_T, the cross-validation set S_{CV}, and the validation set S_V, so that $S=S_T\cup S_{CV}\cup S_V$.

6. Normalization: Vector normalization is usually performed before ANN model fitting. The samples from S_T have been normalized.

7. Model creation: The model was created using one of two ANN libraries for R (`nnet` and `neuralnet`) with parameters adjusted to fit the training data.

The following sections present simple linear and polynomial regression meteorological models, an ANN meteorological model and an ANN neighborhood model, all predicting active power output of a PV system. All models are trained using the same set of 1 second time-series data: Power production of PVs in the SYSLAB laboratory and meteorological data from the month of October 2014. The data used to train models from the following sections is a 1 minute time-series consisting of 44640 rows, randomly divided into the 3 sets S_T,S_{CV},S_V of size 14841, 14901, and 14898, respectively.

1.5.6.5 Regression meteorological model

Several simple linear regression models were created to predict the active power production of the considered PV. Presented meteorological models take solar irradiation, wind speed, wind direction and ambient temperature and output the expected power production in kW, as presented in figure 72.

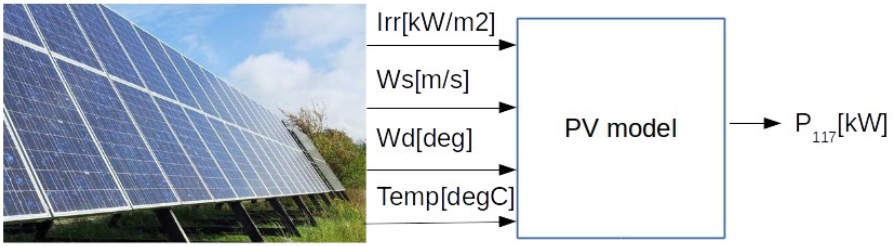

Figure 72: Graphical representation of the simple regression PV model

The R library `lm` was used to create linear and polynomial models from input data. In this section we present five models and compare them.

The Proposed hypotheses $h_{0,}h_{1,}h_{2,}h_{3}$ for the construction of five models $M_{0,}M_{1,}M_{2,}M_{3}$ are as follows:

$$h_0\left(\theta^{(1)}\right)=\theta_0^{(1)}+\theta_1^{(1)}Irr$$

$$h_1(\theta^{(1)})=\theta_0^{(1)}+\theta_1^{(1)}Irr+\theta_2^{(1)}Ws+\theta_3^{(1)}Wd+\theta_4^{(1)}Temp$$

$$h_2(\theta^{(2)})=\theta_0^{(2)}+\theta_1^{(2)}Irr+\theta_2^{(2)}Ws+\theta_3^{(2)}Wd+\theta_4^{(2)}Temp+\theta_5^{(2)}Irr^2+\theta_6^{(2)}Ws^2+\theta_7^{(2)}W^2+\theta_8^{(2)}Temp^2$$

$$h_3(\theta^{(3)})=\theta_0^{(3)}+\theta_1^{(3)}Irr+\theta_2^{(3)}Ws+\theta_3^{(3)}Wd+\theta_4^{(3)}Temp+\theta_5^{(3)}Irr^2+\theta_6^{(3)}Ws^2+\theta_7^{(3)}Wd^2+\theta_8^{(3)}Temp^2+\theta_9^{(3)}Irr^3$$

Parameters for each hypothesis $\theta_{0,}\theta_{1,}\theta_{2,}\theta_{3}$ were calculated using the training set S_T of 14841 samples (out of 44640 samples in the S set), with the help of the `lm` function in R.

Regression model-0

This model uses irradiation data to calculate the power consumption as follows:

$h_0\left(\theta_0^{(1)}\right)=\theta_0^{(1)}+\theta_1^{(1)}Irr$, where $\theta^{(0)}=(0.1094376 \quad 2.2604230)$

Figure 73 shows the linear, single variable model for the PV production, data points are model input data from set S_T and the power calculated form this set. Figure 9 presents the model prediction (red) compared to the real production data (black) from set S_{CV}. It is visible that the model is not very accurate in minimums and maximums of the production, therefore a simple model cannot be used for the intrusion detection purposes.

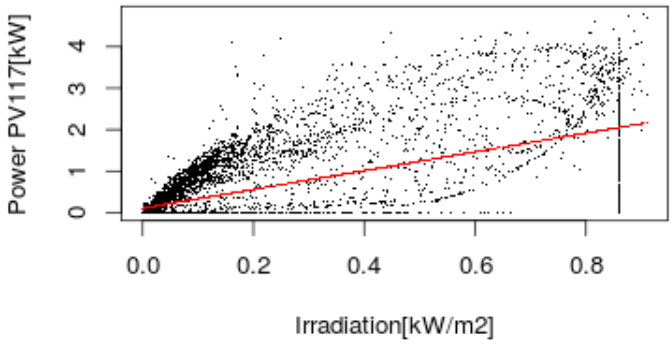

Figure 73: Model-0 training data and model output.

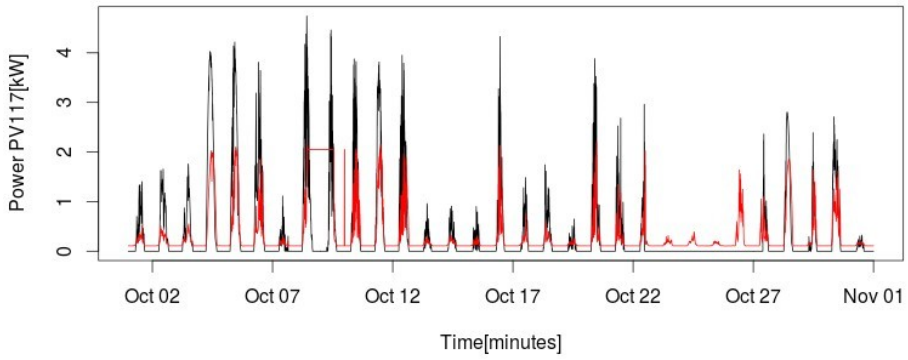

Figure 74: Model-0 output compared to the measured PV117 production.

Regression model-1

This model uses irradiation, wind speed, wind direction and ambient temperature data to calculate the power consumption as follows:

$$h_1\left(\theta^{(1)}\right)=\theta_0^{(1)}+\theta_1^{(1)}Irr+\theta_2^{(1)}Ws+\theta_3^{(1)}Wd+\theta_4^{(1)}Temp$$, where

$$\theta^{(1)}=[0.0152916763304,2.6721188099024,$$
$$(0.0689680031581,-0.0001766575963,0.0014208500330)$$

Figure 75 shows the linear model for the PV production, data points are model input data from set S_T and the power calculated from this set. Figure 76 presents the model prediction (red) compared to the real production data (black) from set . In comparison to the output for Model-0 (Figure 74), model-1 predicts better in the minimums,
and performs slightly better in the maximums.

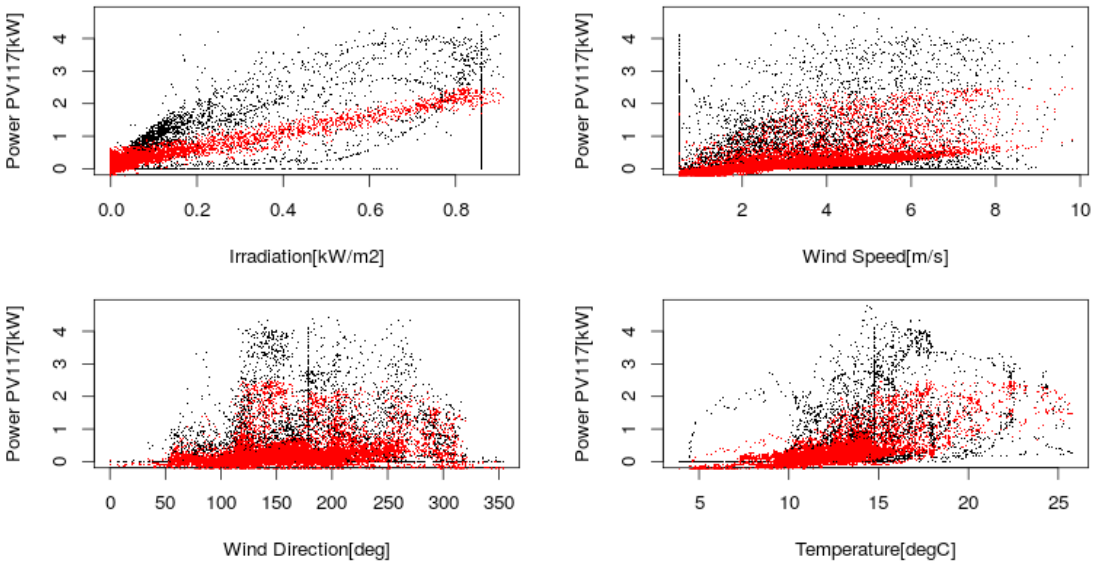

Figure 75: Model-1 training input and output data.

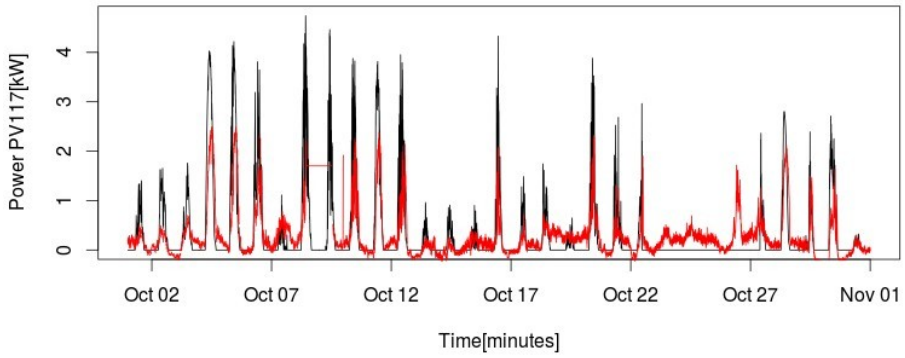

Figure 76: Model-1 PV power prediction results (red) obtained with the cross-validation data set.

Regression model-2

This linear model uses irradiation, wind speed, wind direction and ambient temperature to calculate the power consumption as follows:

$$h_2(\theta^{(2)}) = \theta_0^{(2)} + \theta_1^{(2)} Irr + \theta_2^{(2)} Ws + \theta_3^{(2)} Wd + \theta_4^{(2)} Temp + \theta_5^{(2)} Irr^2 + \theta_6^{(2)} Ws^2 + \theta_7^{(2)} Wd^2 + \theta_0^{(2)} Temp^2 ,$$

where

$$\theta^{(2)} = (-0.990668284934828, 6.070479844300378, 0.110911311188601, -0.003277889451628,$$
$$0.166264220499390, -4.012042544422539, -0.006648793318575, 0.000006192536912,$$
$$-0.006105482360976)$$

Relations between the polynomial model presented in this section and the training input data S_T is presented in figure 77. The parabolic shape of the power prediction mapped to the irradiation does not reflect the strong correlation between the data, and therefore suggests that the power production maximums will be decreased in the model.

Wind direction and temperature are better mapped by the model to the power production, in comparison to model-0 (figure 73) and model-1 (figure 75), improving the power prediction only slightly, as shown in figure 78.

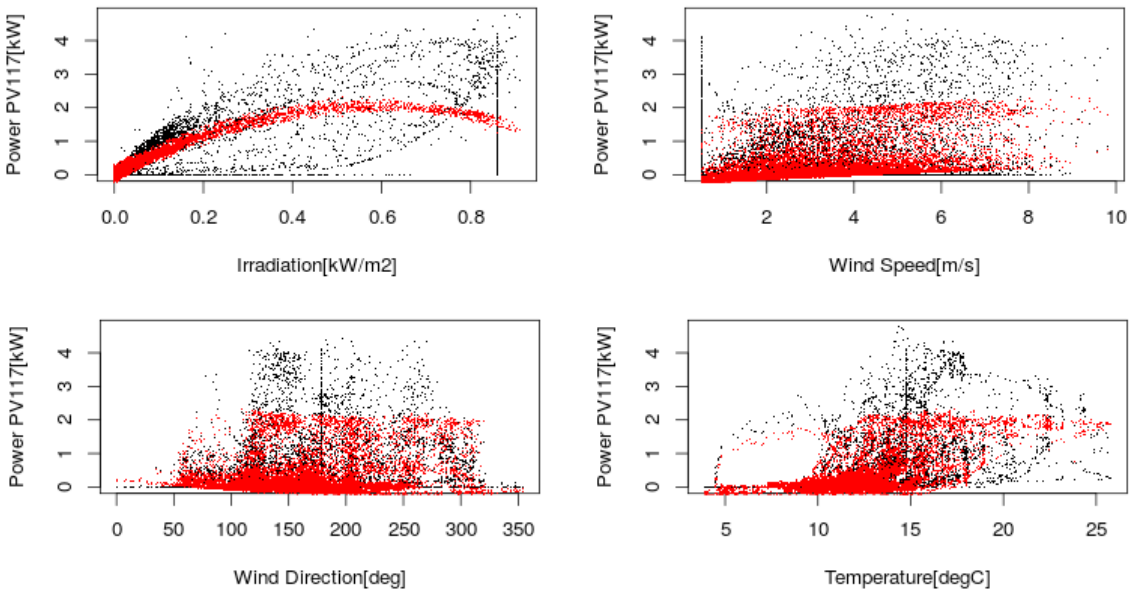

Figure 77: Model-2 graphical representation on the training data.

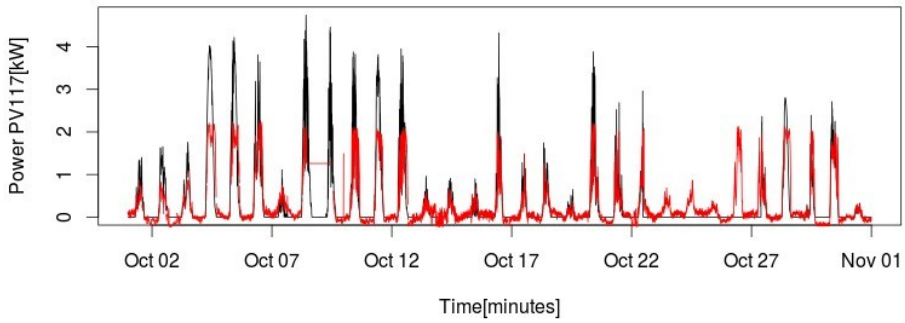

Figure 78: Model-2 PV power prediction results (red) obtained with the cross-validation data set.

Regression model-3

This model uses irradiation, wind speed, wind direction and ambient temperature to calculate the power consumption as follows:

$$h_3\left(\theta^{(3)}\right)=\theta_0^{(3)}+\theta_1^{(3)}\,Irr+\theta_2^{(3)}\,Ws+\theta_3^{(3)}\,Wd+\theta_4^{(3)}\,Temp$$
$$+\theta_5^{(3)}\,Irr^2+\theta_6^{(3)}\,Ws^2+\theta_7^{(3)}\,Wd^2+\theta_8^{(3)}\,Temp^2$$
$$+\theta_9^{(3)}\,Irr^3+\theta_{10}^{(3)}\,Ws^3+\theta_{11}^{(3)}\,Wd^3+\theta_{12}^{(3)}\,Temp^3$$

,where

$$\theta^{(3)}=\{8.815803\text{e-}01,5.244864\text{e+}00,6.418637\text{e-}01,-1.302377\text{e-}03,-3.323558\text{e-}01,$$
$$(6.125233\text{e-}02,-1.589988\text{e-}01,-9.234099\text{e-}06,2.597408\text{e-}02,-4.634485\text{e+}00)$$
$$(1.249784\text{e-}02,3.622099\text{e-}08,-6.469007\text{e-}04)$$

The model input (black) and output for irradiation, wind speed, wind direction and ambient temperature is presented in figure 79. In comparison with model-2 (figure 77), model-3 behaves similarly; also inheriting the problem with predicting maximums of the power production, see figure 80. The additional variables do not improve the model-2 in comparison to model-3. A nonlinear model should be explored to improve the prediction.

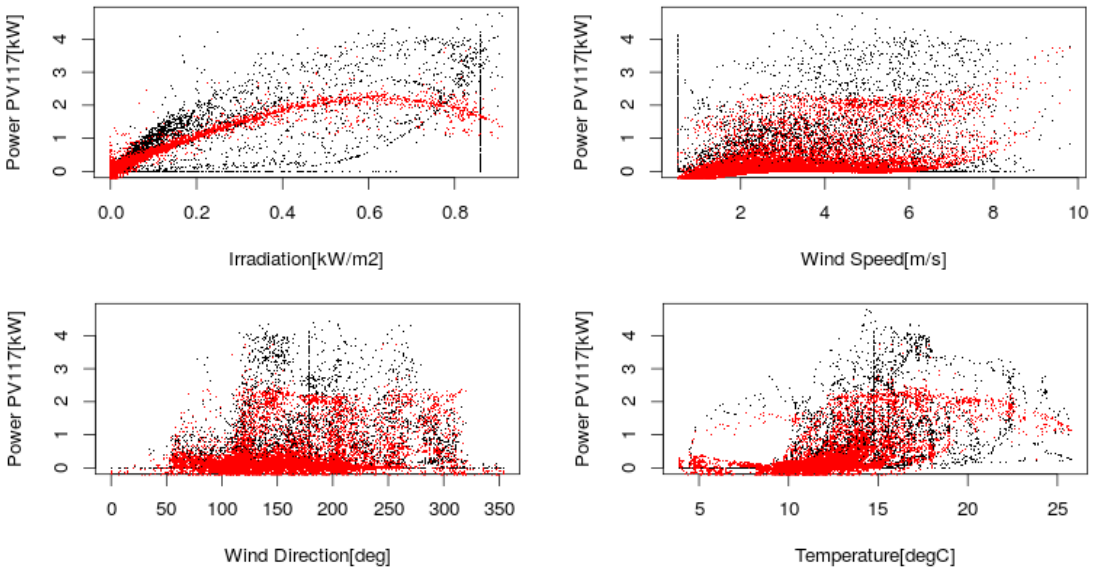

Figure 79: Model-3 graphical representation on the training data.

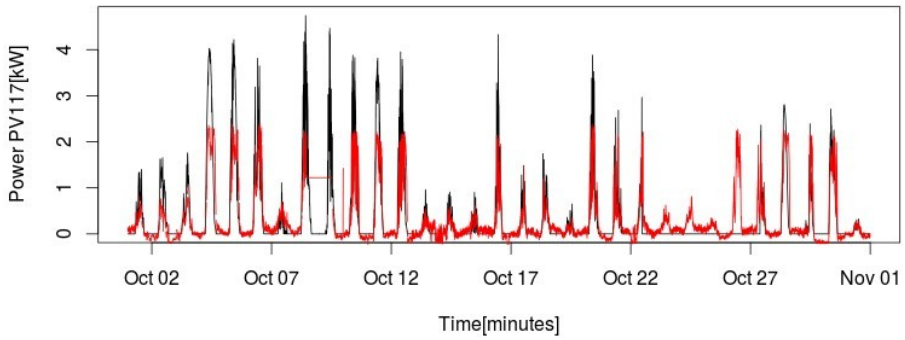

Figure 80: Model-3 PV power prediction results (red) obtained with the cross-validation data set.

1.5.6.6 ANN meteorological model

The nonlinear meteorological model presented in this section uses solar irradiation, wind speed, wind direction and ambient temperature in order to predict the expected power production.

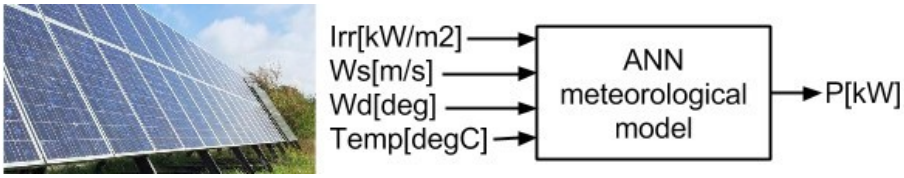

Figure 81: Graphical representation of the ANN meteorological model

The R Package nnet (Feed-Forward Neural Networks and Multinomial Log-Linear Models) is used to create of a supervised learning ANN model. The nnet package was created to calculate single hidden-layer neural networks based on input data, with ANN parameters chosen for each model. The outcome of this process is a single-hidden-layer ANN representing a model of an observed PV.

The data selection process included the correlation analysis of 24-hour periods at a time resolution of one second, as shown in figure 82. The purpose of the correlation analysis was to remove data form the model corresponding to faulty sensors and readings, and possibly control actions, in order to create the model of the normal operation of the PV.

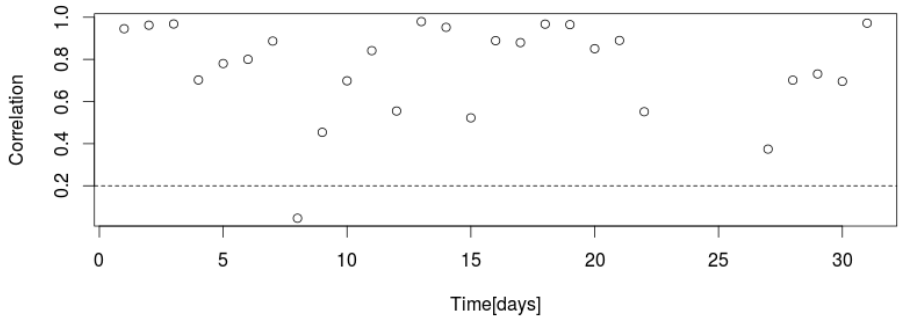

Figure 82: Correlation analysis between the active power and the solar irradiation for days in October 2014

Based on the correlation analysis, the following days in October were chosen as an input to the model: 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 27, 28, 29, 30, 31. The selected days are highlighted with a blue background in figure 83.

The data from each day was aggregated into 1 minute values and divided into three sets S_T, S_{CV}, S_V . A training data set was used to fit the PV model. The initial size of the training set was 12480 rows. After removing all rows where any of the data points has a NaN value, the size of the remaining set was 11739 rows. The neural network training specification is as follows:

```
nnet(power117~irr+windspeed+winddir+temp, data =
october.norm[train] , size = 4, decay = 5e-4)
```

where

- `power117` is the output target variable from the model training set
- `irr` is the solar irradiation in kW/m² from the model training set
- `windspeed` is the wind speed in m/s from the model training set
- `winddir` is the wind direction in deg from the model training set
- `temp` is the outdoor ambient temperature in °C from the model training set
- `october.norm` is the normalized training data set
- `train` is the training set S_T
- `size` is the size of the single hidden-layer, indicating 4 neurons to be trained
- `decay` is the parameter for weight decay
- `range` are the initial random weights on [-rang,rang].

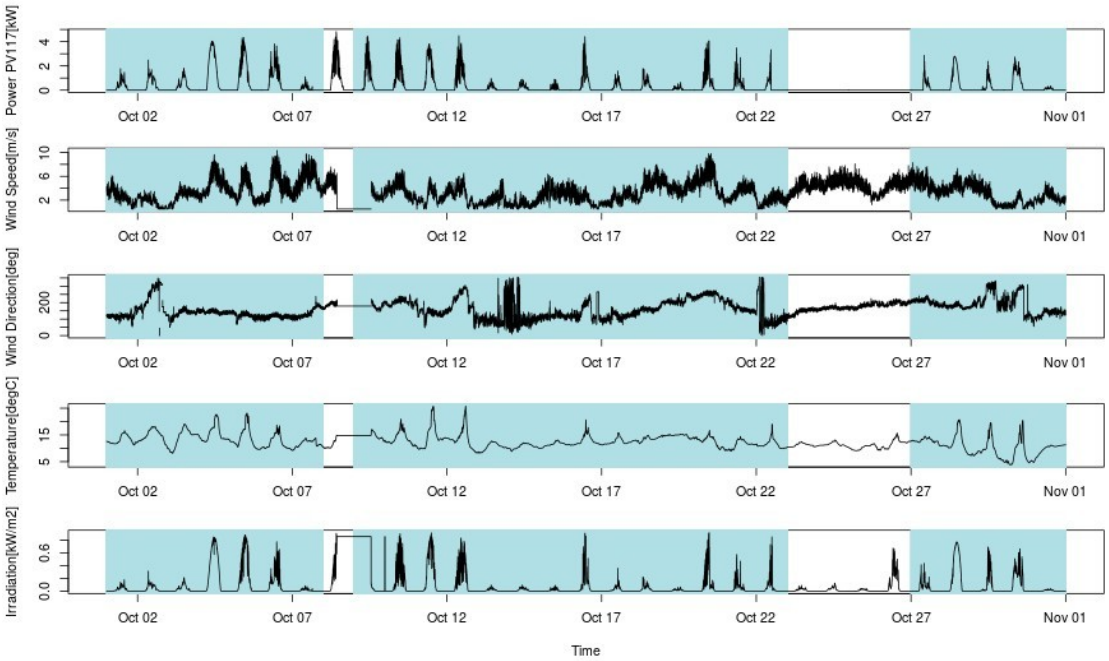

Figure 83: Input data to the meteorological ANN model, training set highlighted with blue background.

The generated model is visualized in the neural network graph in figure 84, where $X1 \ldots X4$ are the features of the model; $B1, I1 \ldots I4$ is the input layer of the neural network, consisting of bias unit and input neurons; $B2, H1 \ldots H4$ is the hidden layer of the network with bias unit and four generated neurons, $O1$ is the output neuron and $Y1$ is the model target variable.

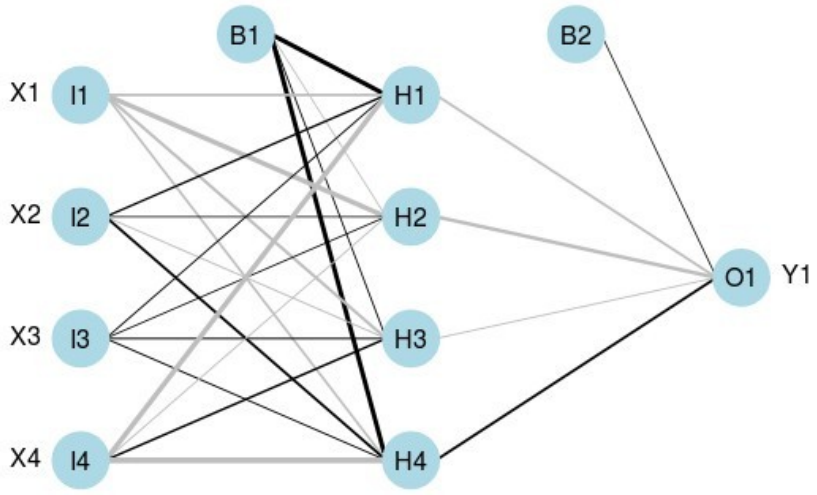

Figure 84: Generated ANN for the meteorological model.

The estimated weights of the model that are the model parameters are presented in figure 85.

Table 2. ANN Meteorological model weights.

From neuron	To neuron	Weight	From neuron	To neuron	Weight	From neuron	To neuron	Weight
B1	H1	20.9850260	B1	H3	1.231233	B2	O1	0.2886376
I1	H1	-9.7884280	I1	H3	-11.795018	H1	O1	-8.7411344
I2	H1	5.2897340	I2	H3	-2.379878	H2	O1	-15.7221988
I3	H1	1.6668830	I3	H3	3.049785	H3	O1	-1.7848833
I4	H1	-24.1790220	I4	H3	6.558410	H4	O1	9.2394091
B1	H2	-0.6929721	B1	H4	21.003534			
I1	H2	-23.2922793	I1	H4	-7.883099			
I2	H2	0.3179632	I2	H4	9.226874			
I3	H2	0.3695722	I3	H4	1.912373			
I4	H2	-1.4526547	I4	H4	-29.720484			

Figure 85: ANN Meteorological model weights.

Figure 86 presents how the ANN Meteorological model maps inputs to outputs of the training set, divided into a graph for each model feature. In comparison with presented regression models (in figure 74, figure 76, figure 78 and figure 80), the non-linear model reflects the power production maximums and minimums well. In figure 87 the model prediction (black) is compared to the actual PV production recorded in October 2014.

The model performs well when predicting the PV production, respecting most of minimums and maximums. Between the 8th and 9th of October, the meteorological station experienced faults for all sensors, while the PV production was undisturbed. This can be verified by data presented in figure 84. Since the model depends on the meteorological input, the output of the model was false, carrying the sensor error forward to its prediction. From October 23rd to 26th, the PV was curtailed to 0kW production by an external setpoint input (see figure 84). The model predicts that production should have been occurring at this time, revealing the control action performed on the PV.

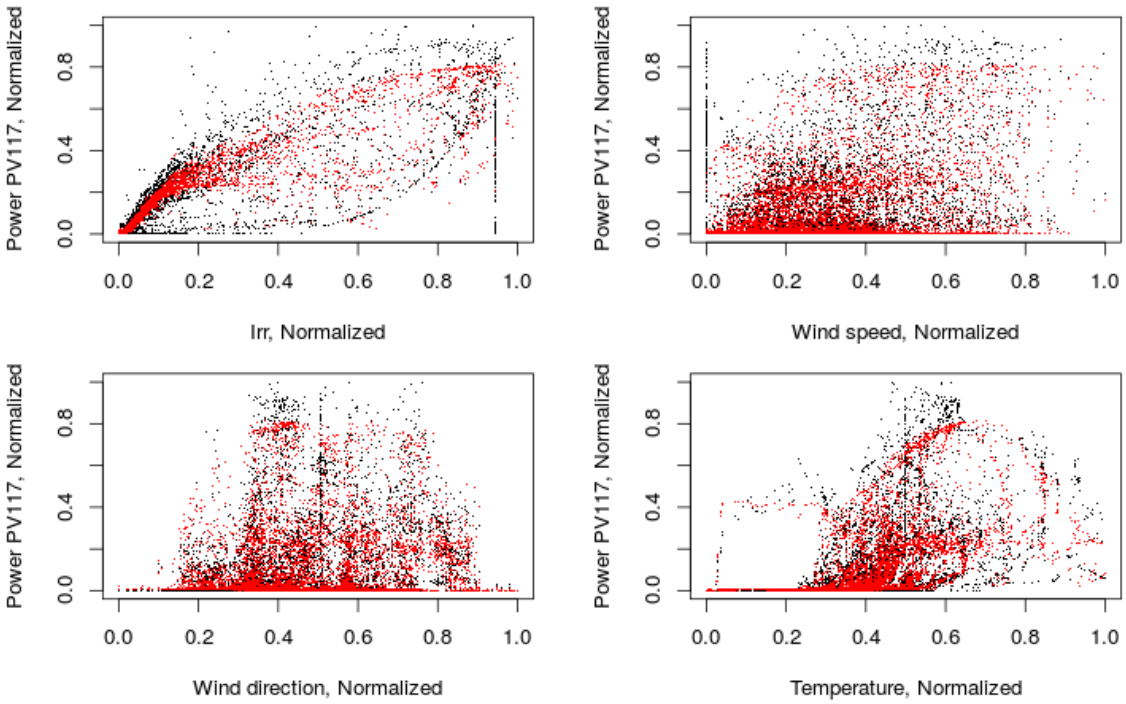

Figure 86: Graphical representation of the input (black) and output (red) data of the ANN meteorological model for different model features: solar irradiation, wind speed, wind direction and outdoor temperature.

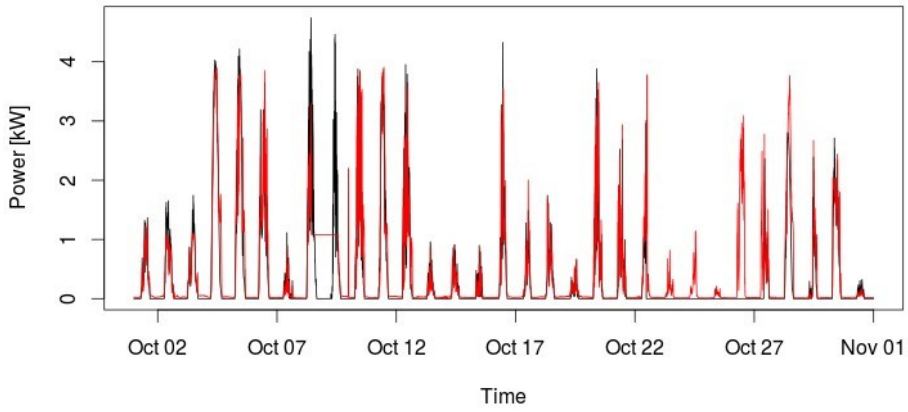

Figure 87: ANN Meteorological model output (red) and corresponding actual production (black).

1.5.6.7 ANN neighborhood model

The proposed model takes the advantage of the geographical location of three PV panels; it assumes that the PV production of a single PV can be modeled with the PV production of two neighboring PVs.

This idea is explored in the following investigation; with use of 1 second data form three geographically dispersed PVs at the SYSLAB laboratory. The general representation of the model and the geographical distance between PV panels is presented in figure 88.

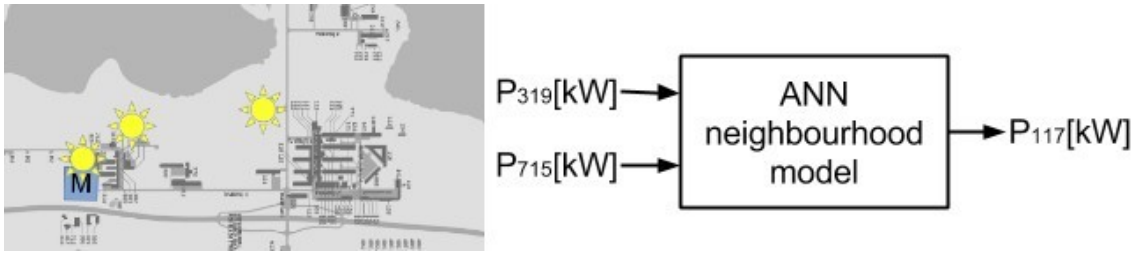

Figure 88: Graphical representation of the neighborhood ANN model.

Similarly to the previous work on the ANN meteorological model, the R package nnet was used for supervised learning of the ANN neighborhood model. Also corresponding to the previous model, the data selection process included a correlation analysis of entire days of data. The purpose of the correlation analysis was to remove data from the model corresponding to faulty sensors and readings, and possibly control actions, in order to create a model of normal PV operation. As a result, several days in October were chosen as an input to the model: 1, 3, 5, 6, 7, 9, 10, 11, 12, 16, 17, 19, 20, 22, 27, 29, 31, as shown in figure 90.

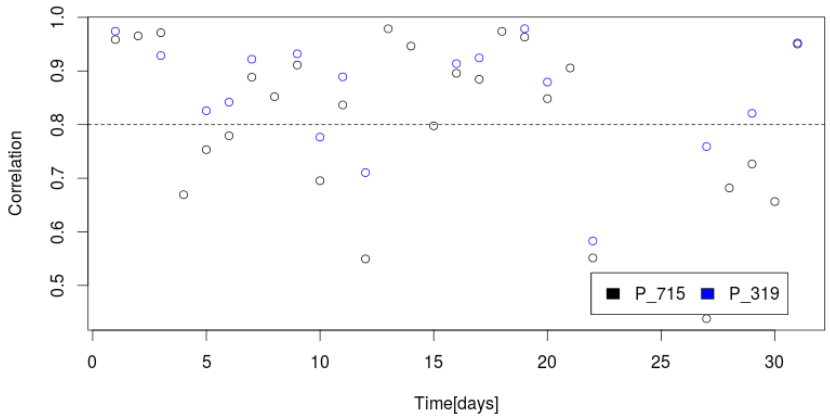

Figure 89: Correlation between PV117 and PV319 and PV117 active power production for each day in October 2014.

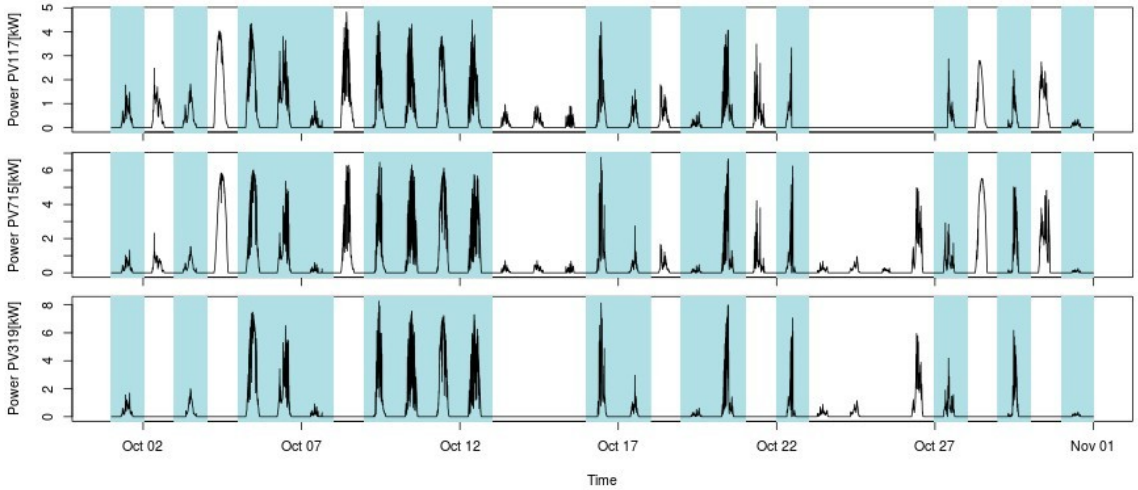

Figure 90: Input data to the neighborhood ANN model, training set highlighted with blue background.

In order to create the ANN neighborhood model, a set of 22341 training samples was created and included in the S_T sample vector for building the model. After the correlation analysis, the remaining sample size was 8202. Samples, where any of the values for each time stamp

was missing (NaN), were removed during the production of the final training set consisting of 8201 rows. The nnet model is defined as follows:

```
nnet.october<-nnet(power117~ power715+power319, data =
october.normalized[train], size = 3, decay = 5e-4, rang = 0.5)
```

where

- power117 is the active power of PV117 production in kW, output target variable from the model training set.
- power715 is the active power of PV715 production in kW, input feature from the model training set.
- power319 is the active power of PV319 production in kW, input feature from the model training set.
- october.normalized is the normalized data set.
- train is the training set S_T .
- size is the size of the single hidden-layer, indicating 3 neurons to be trained in the hidden layer.
- decay is the parameter for weight decay.
- range are the initial random weights on [-rang,rang].

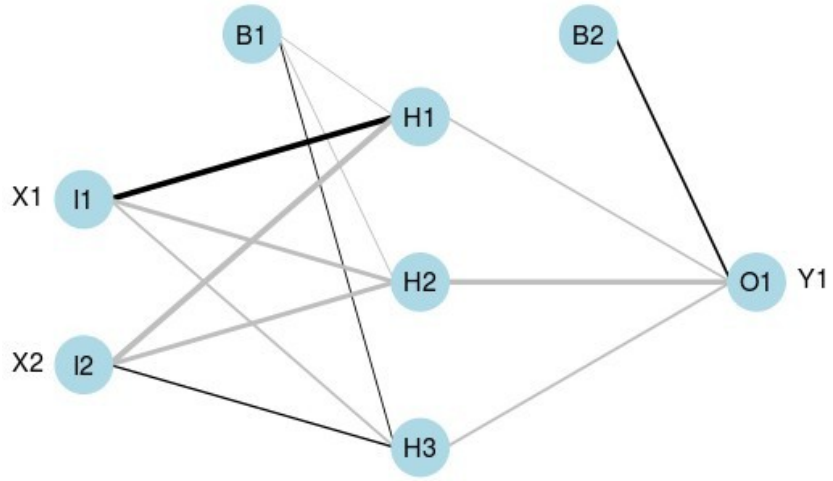

Figure 91: Neural network representing the neighborhood model.

The result of the model training is the neural network presented as a graph in figure 91, where $X1$ and $X2$ are the features of the model; $B1$, $I1$ and $I2$ are bias unit and input neurons; $B2, H1 \ldots H3$ is the hidden layer of the network consisting of bias unit and three generated neurons, $O1$ is the output neuron and $Y1$ is the model target variable. The model parameters, called weights, have be obtained with the presented model fitting technique and presented in figure 92.

From neuron	To neuron	Weight	From neuron	To neuron	Weight
B1	H1	-0.6071153	B2	H3	1.312821
I1	H1	16.0967600	H1	H3	-5.606807
I2	H1	-16.5486141	H2	H3	3.636337
B1	H2	-1.186549	B2	O1	5.203566
I1	H2	-11.274409	H1	O1	-4.580979
I2	H2	-12.308967	H2	O1	-15.039515
			H3	O1	-6.115878

Figure 92: ANN neighborhood model weights.

The relationship between active power from PV715 and target active power from PV117 with observed data (black) and model output data (red) is presented in figure 93. The data used in Figure 93 is from S_{CV} , the data set not used for model fitting. The model output reflects the measurement data quite well.

Figure 94 presents the measurement data of the PV117 production of active power (black) and ANN neighborhood model output (red). Noticeable model errors occur on October 2, 4, 8, 13,14,15,18 and 21. This can be explained by the missing input from one of the PVs (see figure 90), which has a large impact on the model accuracy. In the training period, PV319 was out of operations for several days in October 2014.

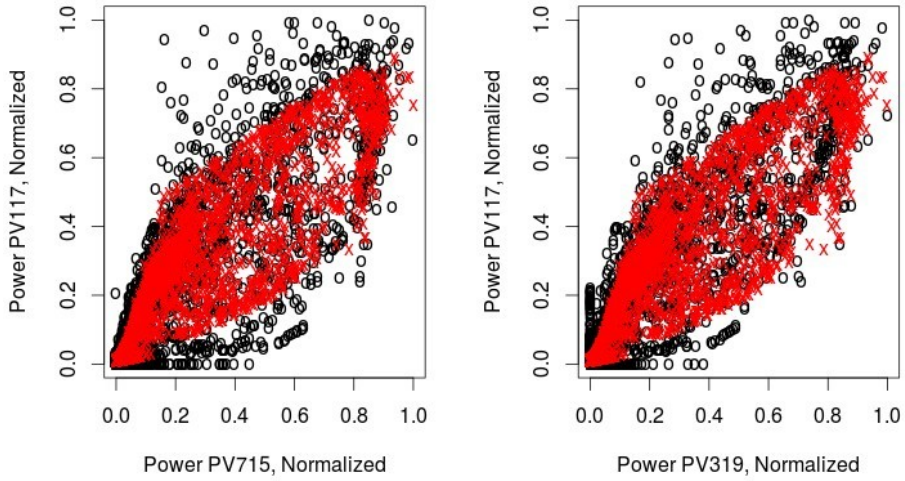

Figure 93: Power production data from PV117 mapped to output of the ANN neighborhood model, both for power production of PV715 and PV319.

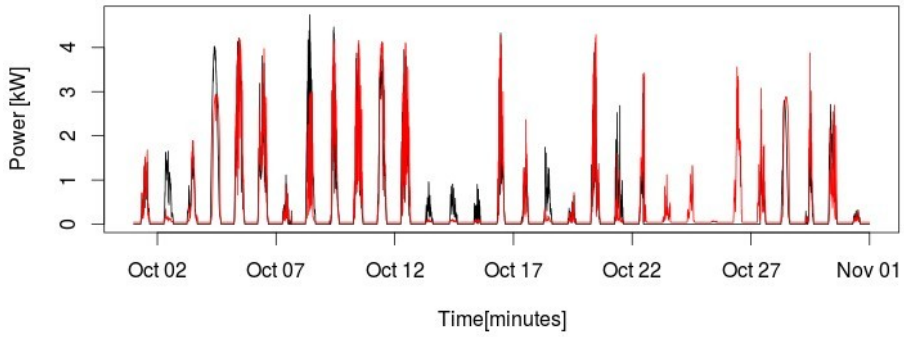

Figure 94: ANN neighborhood model output for all PV data in October 2014.

Between 8th and 9th of October, the meteorological station experienced a failure of all sensors, while the PV production was undisturbed. Since the model does not depend on the meteorological input, the output of the model was more accurate than from the ANN methodological model. Between October 23rd to 26th, the PV117 inverter was curtailed to 0kW production by an external setpoint (see figure 90). The model predicts that PV production should have been occurring at this time, discovering the control action performed on the PV.

Both ANN models perform well in predicting the PV117 production from available inputs. In the following section, the regression and ANN models are compared and evaluated.

1.5.6.8 Model comparison

In order to compare the above models, we calculate the root mean square error (RMSE) between the model output and the measured signal. The RMSE values for all models against the cross-validation data sets S_{CV} and the validation data sets S_V are shown in figure 95.

Model	**RMSE** S_{CV}	**RMSE** S_V	**RMSE** S_{CV} **-RMSE** S_V
Model-0	0.552821866169510	0.559086268457309	-0.006264402287799
Model-1	0.523846479673415	0.532684901739672	-0.008838422066257
Model-2	0.459676871734066	0.470629781117767	-0.010952909383701
Model-3	0.447028350270900	0.455733376498775	-0.008705026227875
ANN meteorological	1.084456005304160	1.109368569589230	-0.024912564285070
ANN neighborhood	0.411795311863365	0.410360936095280	0.001434375768085

Figure 95: Model comparison and validation results.

Cross-validation is a validation technique, testing how a data-driven model generalizes to an independent data set that is different form the model training set. The RMSE S_{CV} column in figure 95 represents the standard deviation of the fitting error (difference between the observed and estimated value) for each model. The RMSE S_v column presents RMSE on another set of data. The small difference between RMSE S_{CV} and RMSE S_V indicates how well a model generalizes to an unknown set of input data. Based on the results presented above, the ANN neighborhood model performs best with an average RMSE of 0.41 kW.

1.5.6.9 Anomaly detection

An anomaly detection method identifies rare data instances or events that do not match an expected pattern [39]. The development of models used for anomaly detection requires cyber-security and power system expertise, and additionally, if data driven models are

required, data analysis knowledge. Once both cyber and physical anomaly detection analysis is performed, cyber-physical metrics need to be developed to combine the information from both domains to address the tight relations between the power system and the ICT domains. Anomaly detection with regression models has been used for discovering cyber-attacks on a SCADA system [40], wind turbine fault detection [41] and PV (photovoltaic power plant) fault diagnostics [42]. A special case of a PV attack against voltage control in distribution power grids has been described in [43].

Two types of anomaly detection can be distinguished: point and contextual. The point anomaly detection takes a global view of the data. Contextual or conditional anomalies were introduced in [44] and are defined as data points that are anomalous in a specific context and acceptable in another context. For example for spatial data, the location of a measurement is its context. For time series, time is the context for each measurement. The advantage of the contextual over point anomaly detection is the detection accuracy. The disadvantage is that this method requires context data, which is not always available.

Two methods for contextual anomaly detection exist: (1) Reduction to a point anomaly detection problem and (2) utilizing existing structures in the data. The reduction to point anomaly detection problem technique divides the data into contextual groups and analyses behaviour attributes for each context separately, reducing the problem to several point anomaly detections. This method produces a model for each context; as a consequence, several models are used to represent a single system. In case of the time contextual data, models for every year, month, day of the month, minute and so on would have to be created. Contextual anomaly models utilising the structure of the data modify the structure of the training data to include time components (day, month, year etc.) in the set of input variables. The modified input data is then used for the training of a single contextual model.

In the energy domain, contextual anomaly detection has been previously used for recognising user behavior in a residential dwelling based on non-parametric belief propagation for energy efficiency [45]. Here, user behaviour is categorised based on unusual equipment usage or bursty occupancy and is used to adjust the energy management schedule. [46] proposes the use of on-line contextual anomaly detection for fault diagnostics of power transformers. In the following, we propose several approaches to contextual anomaly detection: A neural network-based detection algorithm, an ensemble regression-based detection algorithm and the concept of a data integrity monitor, combining both the physical and the ICT state of a DER component with a neural network-based approach.

1.5.6.10 Neural network based detection algorithm

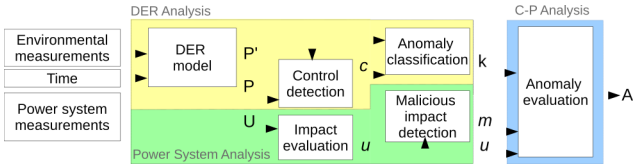

Figure 96: IDS with anomaly detection and power system stability evaluation

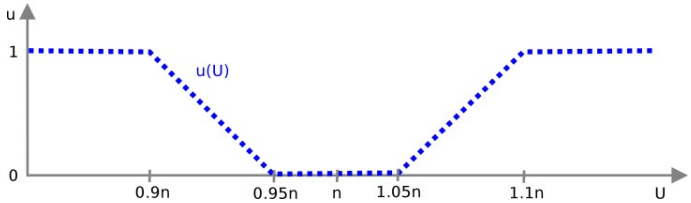

Figure 97: Impact evaluation function

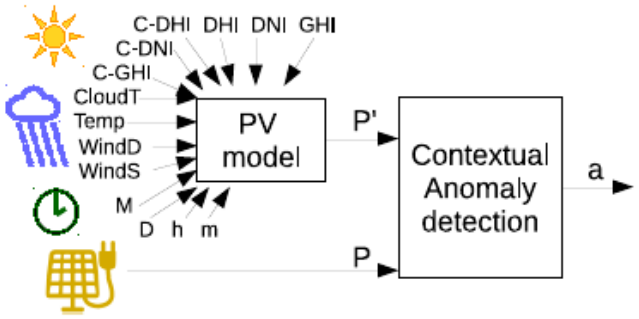

Figure 98: Contextual anomaly detection with a PV model.

Training data

The following sources of input data were used to train the model:

1. PV power production data was taken from the public-domain Dataport database [47] produced by the Pecan Street Smart Grid Demonstration project. The database contains anonymized data of home electricity use, PV power, EV charging and demand response recorded in a suburb of Austin, Texas, from participating households. The data used in SALVAGE is a timeseries of 1-minute active power production data recorded from a single-family home (referred in Dataport as house 774) between January 1st, 2013 to December 31st, 2013.

2. Meteorological data was extracted from the National Solar Radiation Database (NSRDB) [48] maintained by the US National Renewable Energy Laboratory. The data for SALVAGE is a timeseries of 30-minute values from a meteorological station in Austin, Texas during the same time period as the PV data above. The relevant data includes diffuse and clear-sky values for horizontal irradiance (DHI and C-DHI), direct normal irradiance (DNI and C-DNI), global horizontal radiation (GHI and C-GHI), cloud type, temperature, wind speed and wind direction.

The relationship between contextual attributes (hour and month) and power production is presented in figure 99. Linear interpolation was performed on the meteorological data in order to match the 1-minute resolution of the PV data. The resulting total of 525540 data rows were divided into 80% training set (420660 data rows) and 20% validation set (104880 data rows). This training set was used for the PV model training. Additional data covering the period between January 1st, 2014 to January 31st, 2014 was used in the simulation as on-line data. The PV production data used for simulation was modified in order to simulate PV control. In the simulation, an instantaneous and constant curtailment is assumed.

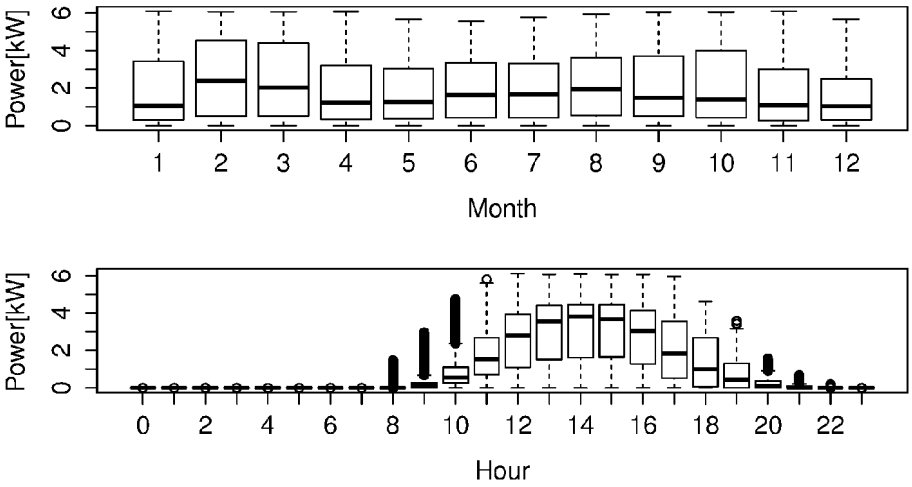

Figure 99: Box plot of time of the day and month, and PV power production.

ANN model

We consider a single layer feed-forward ANN with $n \in N_1$ inputs, one output, and a single set of model features $x = [x_0, x_1, x_2, \ldots, x_n]^T$ and output variable $y \in R$. The hidden layer consists of $h \in N_1$ neurons. In order to train the ANN, the forward propagation algorithm is used. The ANN model hypothesis is as follows:

$$H_w(x) = w_0 + \sum_h w_{1h} \phi\left(\alpha_h + \sum_k w_{kh} x_i\right)$$

where w are the model weights, ϕ_0 is the output function and ϕ_1 is the activation function. Here, the neural network is built to model a non-linear continuous function. According to the Cyberenko theorem, the sigmoid activation function of a single layer feed-forward ANN fulfills the universal approximation theorem, therefore an ANN with a sigmoid activation function can approximate continuous functions:

$$\phi_z = \frac{1}{1 + e^{-z}}$$

The weights are chosen to minimise the cost function with least squares. In forward-feed ANNs, the problem of over-fitting can be minimised by regularization; this is used to minimise the weights of the model. The cost function J with regularization becomes:

$$J(w) = \sum_i \left(\left(H_w\left(x^{(i)}\right) - \left(y^{(i)}\right) \right) \right) + \lambda \sum_h \sum_k w_{kh}^2$$

Ripley [49] suggests to use $\lambda = 10^{-4} \ldots 10^{-2}$ as a regularization parameter for least-squares fitting. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [50] was used for solving the unconstrained nonlinear optimization problem of minimising the cost function $J(w)$.

The point ANN model (ANN-P) consists of 10 input neurons, 15 hidden neurons and one output neuron. The regularization parameter is chosen as λ = 0.0006. The contextual ANN model (ANN-P) consists of 14 input neurons, 20 hidden neurons and one output neuron. The regularization parameter is chosen as λ = 0.0006. Both numbers of the hidden neurons and

the regularization parameter for each model were chosen to minimise the root mean square error (RMSE) of the model prediction. The ANN-C model, with a model RMSE of 0.88, was found to be less accurate than the ANN-P model with a model RMSE of 0.43.

Simulation

A cosimulation set-up combining PV, house and meteorological station model, an electrical load flow simulator (pyPower) and an attack simulation was implemented, using the Open source framework mosaik [51] as an orchestrator. Two scenarios were chosen to test the detection method: Both contain a small distribution grid consisting of two feeders with residential buildings and rooftop PV. For each scenario, two variants were generated: One for normal operation and one in which the system is under cyberattack. The test hypothesis is that an attacker controls operation of the PV system in order to influence voltage on the line, leading to reduction of power quality. An autonomous IDS monitor observes each PV plant and tests the scenario hypothesis. The objective of the monitor in each case is to determine if the PV control leads to over- or undervoltage on the line, defined as a 10% deviation from the nominal voltage.

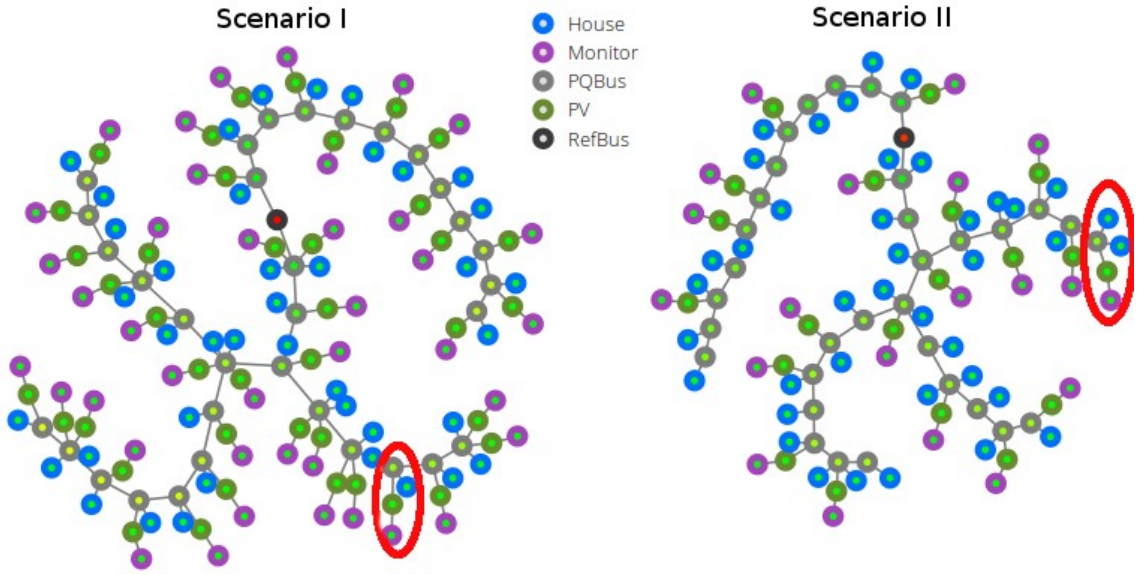

Figure 100: System configuration for Scenario I and II.

Scenario I considers 100% residential PV penetration. The system configuration used for this scenario consists of 40 houses and PVs, divided into two feeders: 12 of houses and PVs on feeder A and 28 of houses and PVs on feeder B (figure 100). Ten houses and corresponding PVs have been created from real house data and replicated to create 40 prosumers.

The actors in the normal operation variant are: houses, PVs, monitors and an aggregator. The aggregator reads the voltage from each PQbus (connection point to the grid from both house and the PV) and curtails the PV in case of overvoltage. The outcome of the aggregator operation is presented in figure 101(a). During 45 minutes of the operation, voltage problems are visible (30 minutes of overvoltage and 15 minutes of undervoltage).

In the variant where the system is under attack, the actors are as follows: houses, PVs, monitors and an attacker. The attacker gathers information about the active power production of each PV and voltage on each PQbus. The attacker sends control signals to each PV in order to cause either under- or overvoltage. It is visible in figure 101(c) that the attacker's decision was not to curtail the PV operation and worsen the overvoltage condition, as presented in 101(d).

Under attack, the voltage problems increase to 240 minutes (225 minutes of overvoltage and 15 minutes of undervoltage). The difference between voltages during normal operation and

during attack is presented in figure 101(e). It is visible that the voltage mostly decreases in this scenario.

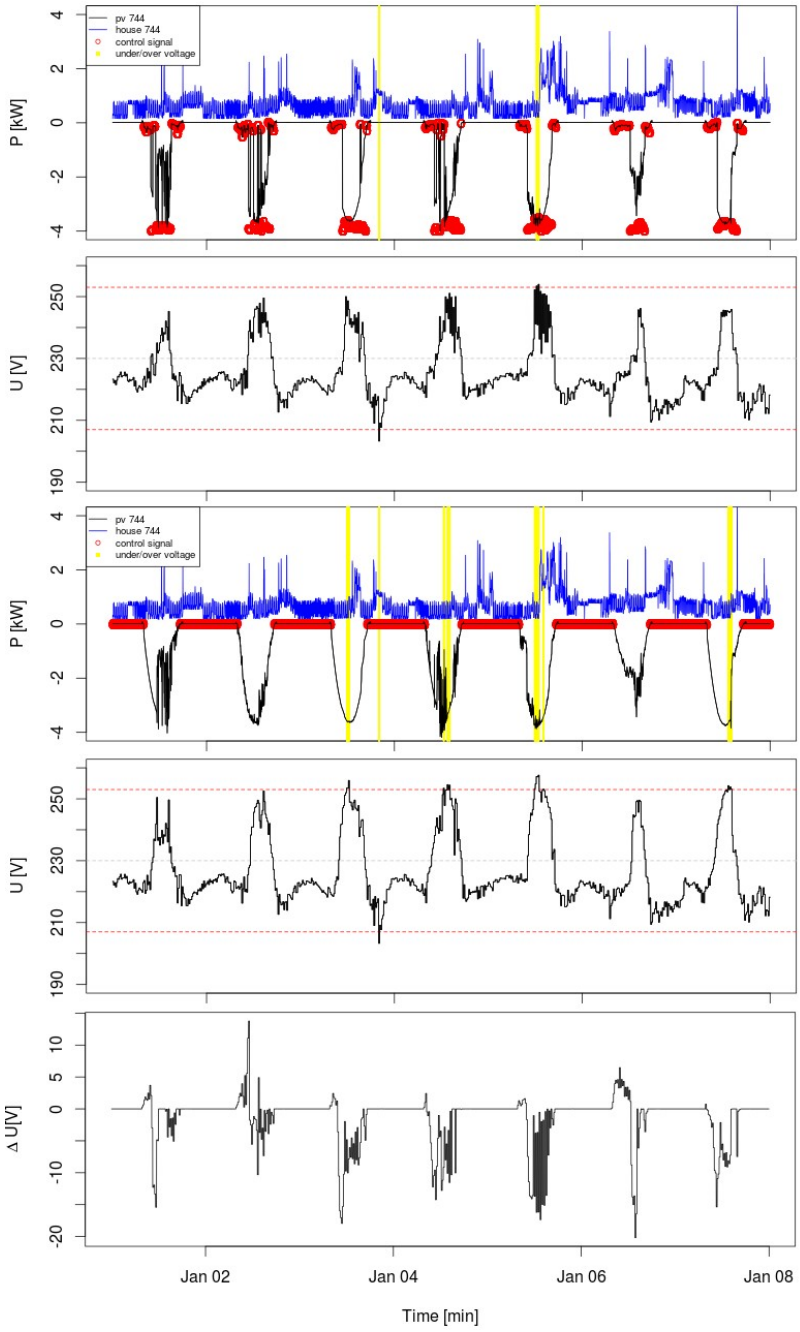

Figure 101: Scenario I: House and PV load pattern (a) and voltage (b) normal behaviour; house and PV load pattern (c) and voltage (d) behaviour under attack; (e) voltage difference between the normal behaviour and the attack.

In Scenario II, 50% of the houses are equipped with rooftop PVs. The system configuration for this scenario consists of 40 houses and 20 PVs, divided into two feeders: 12 houses and 5 PVs on feeder A and 28 houses and 15 PVs on feeder B (figure 100). Similarly to normal operation in Scenario I, the aggregator is controlling the PV systems in order to meet the voltage limits, as presented in figure 102(a,b).

There are several voltage problems: 15 minutes of overvoltage and 135 minutes of undervoltage. In the attack use case, the attacker is aiming at increasing the over- and undervoltage duration by controlling the PV. It is visible in figure 102(c) that the attacker decides to curtail PV plant "PV744" to 0kW , which leads to a decrease in voltage. The total

duration of voltage problems is increased to 420 minutes of undervoltage. The voltage difference is shown figure 102(e). It is visible that voltage has been significantly decreased in this scenario.

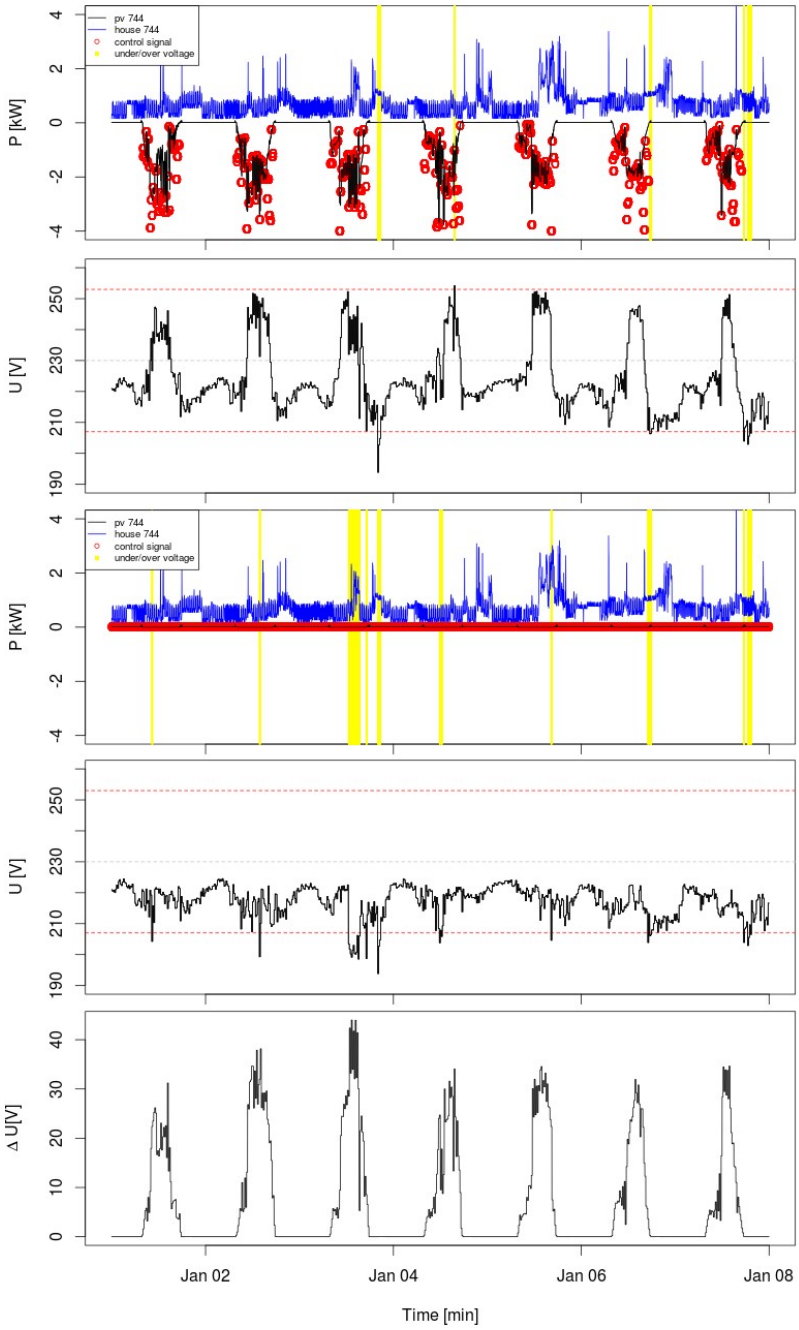

Figure 102: Scenario II: House and PV load pattern (a) and voltage (b) normal behaviour; house and PV load pattern (c) and voltage (d) behaviour under attack; (e) voltage difference between the normal behaviour and the attack.

Results

The detection method has been tested in two scenarios. The results are divided into accuracy of control detection and overall results of the malicious control detection.

A confusion matrix and accuracy calculations are used to evaluate the control and attack results. The confusion matrix is a collection of occurrences of true positives (TP), true negatives (TN), false positives (FP), false negatives (FN) evaluated from a population of results. The accuracy is calculated as follows:

$$Acc = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

As seen in figure 103, the accuracy of the control action detection for point detection ranges between 0.39 and 0.58, where contextual anomaly accuracy is between 0.79 and as much as 0.94 for attack use case in Scenario II. Both methods recognised fewer control actions during attack in Scenario I than in Scenario II. On average the accuracy of detection for the contextual method increases by 0.37 over the point method that accounts to 55% in the presented scenarios.

Use case	TP	TN	FP	FN	Acc
contextual anomaly					
Scenario I: normal	2033	5956	218	1873	0.79
Scenario I: attack	6	8863	1208	3	0.88
Scenario II: normal	2034	5956	218	1872	0.79
Scenario II: attack	3498	5953	213	416	0.94
point anomaly					
Scenario I: normal	1770	2194	3980	2136	0.39
Scenario I: attack	9	5498	4573	0	0.55
Scenario II: normal	1466	2194	3980	2440	0.36
Scenario II: attack	3693	2194	3972	221	0.58

Figure 103: Confusion matrix, control detection

As presented in figure 104, the discovery of malicious control is performed well by both point and contextual detection, scoring 0.99 or 1 accuracy. For the attack in Scenario II both methods have 0.93 accuracy. However the attack case of the Scenario I is more problematic or both methods however, contextual anomaly recognised 4 times more true positives than point anomaly detection, increasing the accuracy by 56%.

Use case	TP	TN	FP	FN	Acc
contextual anomaly					
Scenario I: normal	0	45	0	0	1
Scenario I: attack	44	15	0	181	0.25
Scenario II: normal	0	150	0	0	1
Scenario II: attack	249	141	0	30	0.93
point anomaly					
Scenario I: normal	0	45	0	0	1
Scenario I: attack	11	15	0	214	0.11
Scenario II: normal	0	149	1	0	0.99
Scenario II: attack	249	140	1	30	0.93

Figure 104: Confusion matrix, malicious control detection

Discussion

The simulation results obtained from the chosen scenarios confirm that a contextual anomaly detection is more accurate than point anomaly detection. In the present implementation the IDS analysis is limited to a simple voltage use case. A broader analysis module is required for other types of malicious control. The presented DER model is calculated based on historical data from the near past; in order to improve the detection quality, the model would need to be recalculated periodically or be based on a larger set of data.

The algorithm has so far only been analyzed for a single DER. If the underlying model is recalculated periodically the ANN training execution complexity should be considered. Additionally, future work should include different attack profiles.

1.5.6.11 Ensemble regression based detection algorithm

In the proposed model-based anomaly detection method, normal DER behaviour is modelled in the DER model component (figure 105). The output of the model is compared to sensor measurements (or target data) in the anomaly detection component. Differences between normal and observed DER behaviour can originate form several sources: sensor error, model error, DER fault, or malicious or verified DER control. The output of the model-based anomaly detection is either a label (class) or an anomaly score for every data input.

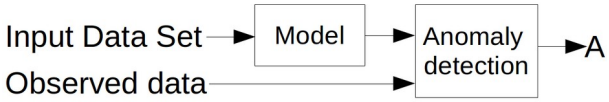

Figure 105: Flow diagram of the model-based anomaly detection

Ensemble learning combines several models to produce a prediction to solve classification and regression problems. The increased robustness and accuracy of ensemble methods over single model methods was reported in [52]. Ensemble learning consists of three steps: generation, pruning and integration. First several redundant models are generated, then the set of models is pruned by removing some of the generated models, finally the base model results are combined to create the ensemble prediction. An overview of ensemble regression approaches for generation, pruning and integration are presented in [53]. The ensemble is evaluated by the degree of agreement between predictions represented by their overall spread.

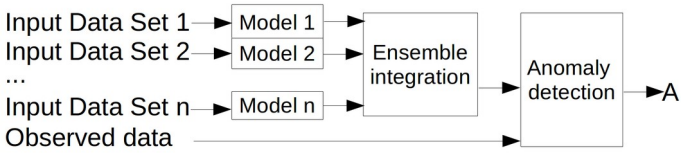

Figure 106: Ensemble model-based anomaly detection architecture.

The proposed ensemble model-based anomaly detection (EM-AD) uses two or more DER normal behaviour models which produce the same output variables based on disjoint sets of inputs. The additional Model merging component calculates the final model output that is next compared to the observed output in the Anomaly detection component (figure 106). Within the SALVAGE project, the EM-AD method was applied to a PV component and implemented as a proof of concept, using historical time series of power and meteorological measurements obtained from a PV plant.

Model building

The semi-supervised anomaly detection uses partially labelled data to train the normal model. Since the historical data has not been labelled, we use correlation analysis as a method for selecting a training set to improve the normal model and consequently enhance the anomaly detection performance. The chosen model building stages are as follows: data cleaning, aggregation, data scoring with correlation analysis, model data labelling and selection, removal of missing values, normalization and finally ANN model creation with supervised model training (figure 107).

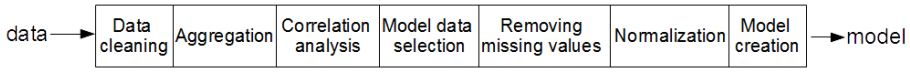

Figure 107: The proposed model building method.

The correlation analysis serves two purposes: filtering data for the normal behaviour model and discovery of sensor faults. This data selection step is based on the assumption that the output of the model is correlated to one or more of its features. Here, we use the standard Pearson product-moment correlation coefficient of two variables. The proposed correlation analysis takes a defined subset of features and the model output and calculates its total correlation. Let $x_i^{(j,k)}=\left\{x_i^{(j)},x_i^{(j+1)},\ldots,x_i^{(j+k)}\right\}$ be a subset starting from sample $j\in N_0$ of size $k\in N_1$ where $k\leq n$, of the i-th feature, and $y^{(j,k)}=\left\{y(j),y^{(j+1)},\ldots,y^{(j+k)}\right\}$ is a variable that is the matching subset starting from sample j of size k of the output. The correlation Pearson product-moment correlation coefficient is calculated as $corr\left(x_i^{(j,k)},y^{(j,k)}\right)=\frac{cov\left(x_i^{(j,k)},y^{(j,k)}\right)}{\delta x_i^{(j,k)}\delta y^{(j,k)}}$, where $cov(x)$ is the covariance of x and $\delta(x)$ is the standard deviation of x. The calculated correlation serves as a normality score for model data selection.

Samples of all features from the training set are evaluated based on the calculated correlation score. The proposed method allocates a sample into one of two groups: Normal behaviour and suspicious behaviour. For a chosen $\alpha\in[0,1]$, samples with $corr\left(x_i^{(j,k)},y^{(j,k)}\right)>\alpha$ are allocated to normal behaviour. If $corr\left(x_i^{(j,k)},y^{(j,k)}\right)\leq\alpha$ or if $corr\left(x_i^{(j,k)},y^{(j,k)}\right)$ does not exist, the samples are allocated to the suspicious behaviour group and are removed from the training set. Note that the correlation cannot be calculated if the standard deviation of $x_i^{(j,k)}$ or $y^{(j,k)}$ is zero.

ANN model creation

We consider an ANN with $n\in N_1$ input variables $x=\left[x_0,x_1,x_{2,\ldots},x_n\right]^T$ where $x_n\in R$ and $x_0=1$ is a bias unit. The output variable of the considered ANN is $y\in R$. Let $a_i^{(j)}$ be the activation of neuron i in layer j, where $j\in[1,2,\ldots,l]$, with l being the number of layers. $\Theta(j)$ is a matrix of weights controlling the function mapping from layer j to layer j + 1. The considered hypothesis function approximated by the ANN is $h_\Theta(x)\in R$. Any layer L_j of the ANN consists of s_j neurons $a^{(j)}=\left[a_0^{(j)},a_1^{(j)},a_2^{(j)},\ldots,a_{sj}^{(j)}\right]^T$. The size of the layer j can be different for every hidden layer. The input layer L_1 is of size n, corresponding to the features vector. The output layer L_3 is of size 1 since the considered hypothesis function is $h_\Theta:R_n\rightarrow R$. The neural network architecture, including the number of inputs, outputs, layers and neurons in each layer, as well as the selection of the transfer function, describes an artificial neural network. Supervised learning methods for training ANN use the training examples $x_0,x_1,x_2,\ldots,x_n,y$ to calculate weight matrices $\Theta^{(0)},\Theta^{(1)},\ldots,\Theta^{(l-1)}$. The neural network architecture and the calculated weight matrices are jointly used for the approximation of an unknown function representing the relationship between input features and output variables. This way an artificial neural network can be trained to approximate transfer functions, especially unknown non-linear relationships.

EM-AD for a PV plant

The architecture of the EM-AD is presented in figure 108. Sensor data of solar irradiation, wind speed, wind direction, ambient temperature, hour of day and power consumption of two neighbouring PVs (PV319 and PV715) are used as input. The proposed ensemble regression is composed of two regression models. The models were generated from disjoint parameter sets and a contextual parameter (hour of day), creating redundant heterogeneous ANN regression models of active power production. The ensemble model set was not pruned because the set contains only two models. The ensemble integration is usually calculated as a linear combination of the predictions. In this case, the ensemble power prediction P' is calculated from predictions for each model P^N and P^M as $P'=\alpha P^N+\alpha P^M$, where $\alpha=0.5$ corresponds to equal weight averaging.

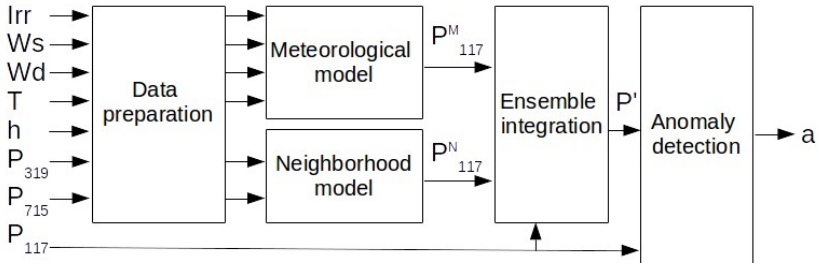

Figure 108: Architecture of the proposed PV ensemble regression model anomaly detection (EM-AD)

The ensemble prediction P' is weighted with the anomaly score in the anomaly evaluation component. The anomaly score is based on the correlation analysis for both ANN models as presented in figure 132. Partial anomaly scores a_M and a_N are calculated for both models as:

$$a=\begin{Bmatrix}1 & corr\geq 0.2\\ 10 & corr<0.2\end{Bmatrix}$$

The anomaly score a_s combines the partial scores for the models a_M and a_N and is calculated as $a_s=\frac{1}{(a_M \cdot a_N)}$. The anomaly score a_s is multiplied by the difference between the ensemble prediction P' and measured power P to calculate the anomaly $a=a_s\cdot(P'-P)$. The chosen anomaly threshold is $\epsilon=01$, therefore only observations with $a>\epsilon$ are considered.

Results

In the considered scenario, one month of the historical active power production of a single PV plant is analysed. In the analysed period of time, the PV should have not been controlled by external input. The EM-AD method is designed to detect anomalous curtailment of the PV active power production to zero.

The degree of agreement between the ensemble predictions is given by their overall spread $s=P^N-P^M$ with the first and third quadrant at -0.011 and 0.006, respectively, and a standard deviation of 0.511. This indicates that the models generally agree in their predictions. Figure 109 presents nine approaches for model-based anomaly detection which were performed using the October 2014 PV data set. The evaluated models are M (meteorological), N (neighbourhood), MN (joint model with inputs from M and N), EMN

(ensemble of M and N). The used training sets are: cor (correlated days for the data set) and full (entire data set). Two anomaly detection methods are used: M-AD (model based anomaly detection) and EM-AD (ensemble regression model anomaly detection).

The confusion matrix is a compilation of instances of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), evaluated from a population of results. To measure the correctness of the anomaly detection we calculate eight significant measures: accuracy (ACC), precision (PPV), negative predictive value (NPV), false negative rate (FNR), sensitivity (TPR), specificity (TNR), false positive rate (FPR) and false discovery rate (FDR). The proposed EM-AD with the correlation training selection approach achieves an accuracy of 0.976, which improves the accuracy by 0.4-11.1% for single model AD, 2.3-9.2% for joint model AD, and 7.3-7.6% over the method without correlation ensemble integration. The precision of the proposed method is 0.947, which improves the precision by 23.8-73.8% for single model AD, 10.1-58,6% for joint model AD, and 61-63.8% over the method without correlation ensemble integration. EM-AD with correlation training data selection additionally keeps low values for FNR of 0.358, FPR of 0.002 and FDR of 0.053. While the specificity has improved only by 10.6% at best, totalling to 0.998, the sensitivity is 0.642 which presents an improvement of up to 35.5% over other presented methods.

Model	Train	AD	TP	TN	FP	FN	ACC	PPV	NPV	FNR	TPR	TNR	FPR	FDR
M	full	M-AD	1198	37445	4531	1466	0.866	0.209	0.962	0.550	0.450	0.892	0.108	0.791
M	cor	M-AD	1633	41305	671	1031	0.962	0.709	0.976	0.387	0.613	0.984	0.016	0.291
N	full	M-AD	1543	39242	2734	1121	0.914	0.361	0.972	0.421	0.579	0.935	0.065	0.639
N	cor	M-AD	1736	41660	316	928	0.972	0.846	0.978	0.348	0.652	0.992	0.008	0.154
MN	full	M-AD	764	38723	3253	1900	0.885	0.190	0.953	0.713	0.287	0.923	0.077	0.810
MN	cor	M-AD	764	41756	220	1900	0.953	0.776	0.956	0.713	0.287	0.995	0.005	0.224
EMN	full	M-AD	1447	38730	3246	1217	0.900	0.308	0.970	0.457	0.543	0.923	0.077	0.692
EMN	cor	M-AD	1709	38614	3362	955	0.903	0.337	0.976	0.358	0.642	0.920	0.080	0.663
EMN	cor	EM-AD	1709	41880	96	955	0.976	0.947	0.978	0.358	0.642	0.998	0.002	0.053

Figure 109: Results, confusion matrix and statistical metrics

1.5.6.12 Data integrity monitor

In this work a first version of the on-line data integrity monitor was created for a PV plant with an OPC-UA interface. The cyber-attack target is to compromise the integrity of the data produced by the PV plant. The attack investigated in this work was a modification of a PV active power limiting setpoint. We considered data in transit integrity. The developed monitor uses the data produced by the OPC UA server and additionally meteorological data and measurement from an independent power instrument. The architecture of the monitor is presented in figure 110.

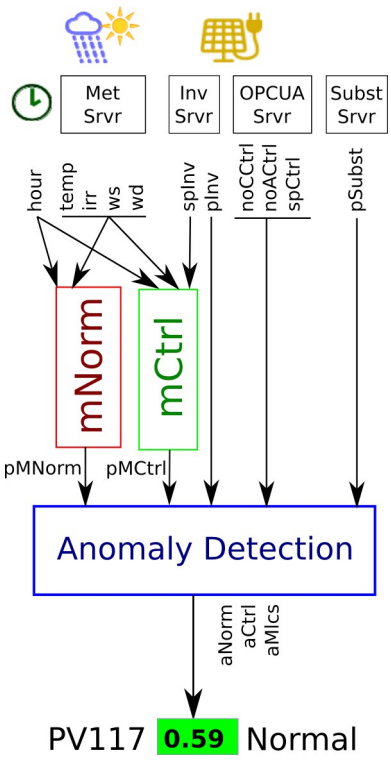

Figure 110: OPC-UA PV monitor architecture.

The components of the architecture are:

- MetSrvr: a server acting as a source of meteorological data from a location close to the investigated PV plant. The data includes temperature (temp), solar irradiance (irr), wind speed (ws) and wind direction (wd).

- InvSrvr: an OPC-UA server providing a monitoring and control interface to the PV inverter. In this work, only one data element in each direction is considered: a measurement of instantaneous active power at the inverter PCC (pInv) and an active power limiting setpoint which can be sent to the inverter (spInv).

- OPCUASrvr: an OPC-UA server providing OPC-UA specific security events, for example new client connection, authentication status, requested data and modified data. These alerts and events are being aggregated and the OPC-UA PV monitor registers the total number of connected clients (noCCtrl), the number of connected authenticated controllers (noACtrl) and the last setpoint set by any controller (spCtrl).

- SubstSrvr: A server associated with an electrical distribution substation, acting as a source of an additional power measurement (pSubst) at the point of common coupling, independent of the PV inverter's built-in measurement.

- nNorm: a PV normal model component which uses hour-of-day (hour), temperature (temp), solar irradiance (irr), wind speed (ws) and wind direction (wd) as input in order to predict active power production of the PV plant (pMNorm).

- nCtrl: a PV normal behaviour with control model component which uses hour-of-day (hour), temperature (temp), solar irradiance (irr), wind speed (ws), wind direction (wd) and inverter setpoint (spInv) as input in order to predict the active power production of the PV plant in the context of the control signal (pMCtrl).

- AnomalyDetection: a classification model using normal power prediction (pMNorm), normal power prediction in the context of control (pMCtrl), PV output measured by

the inverter (pInv), the total number of connected clients (noCCtrl), the number of connected authenticated controllers (noACtrl), the last setpoint set by any controller (spCtrl) and active power measurement at the common point of coupling (pSubst). The model calculates the probability for each class of anomaly: Normal, Controlled and Malicious (aNorm, aCtrl, aMlcs).

PV normal model

The PV plant normal model inputs are solar irradiance [kW/m2] (irr), wind speed [m/s] (ws), wind direction [°] (wd) and hour of day (hour). The model output is active power production [kW] (power). The input significance analysis of the linear model is based on the same inputs and outputs as the presented model, using test statistics under the null hypothesis, shows that all inputs are significant. The training data set consisted of measurements at 1 second resolution from a physical PV system at the SYSLAB laboratory, recorded in June 2016.

An overview of the training data is presented in figure 6. The set consists of several days of data, with three missing days on the 3rd, 7th and 8th of June. These gaps do not diminish the model quality.

An artificial neural network (ANN) was used to model the normal behaviour of the PV plant. The R Package `nnet` for R has been used to parametrize feed-forward neural networks with a single hidden layer. In order to determine the optimal number the hidden neurons in the ANN, ten models of between 1 and 20 hidden neurons were trained. A comparison of the normalised root mean squared error (RMSE) of the ten models is presented in figure 112.

The final ANN consists of 4 inputs, 10 hidden neurons, a bias unit and one output. The formula to train the neural network, as described below, uses a regularization parameter decay of 0.0006.

The data set used to train the ANN was normalised in order to improve the model accuracy.

```
nnet.formula(formula=pInv~irr+hour+ws+wd, data = dataN,
size=10, decay= 6e-04, maxit=Machine$integer.max)
```

The residuals obtained form the model are presented in figure 113. The obtained RMSE is 0.9589178. The graphical representation of the mapping between model inputs and outputs is presented in figure 114.

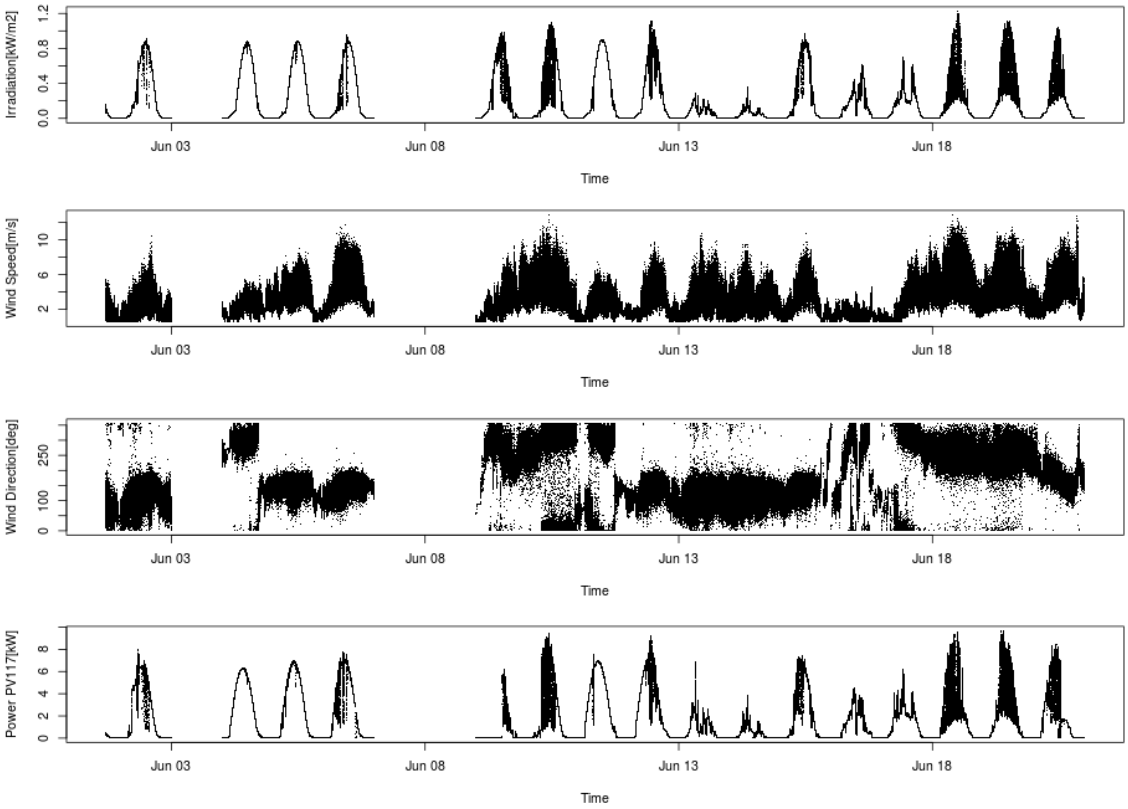

Figure 111: Normal model training data.

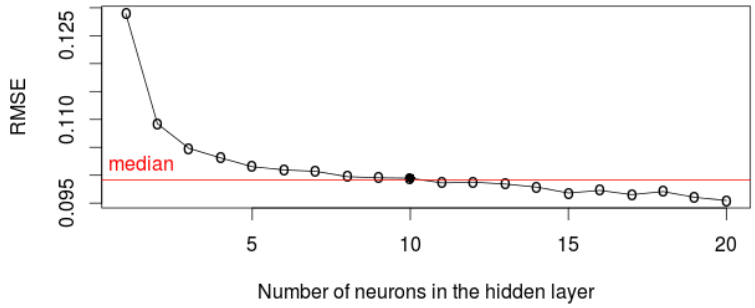

Figure 112: RMSE of the hidden layer neurons ANN training sessions (from 1 to 20 neurons).

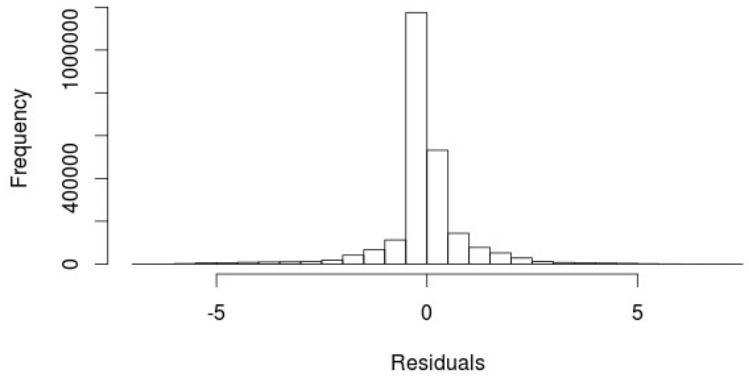

Figure 113: Histogram of the normal model's residuals.

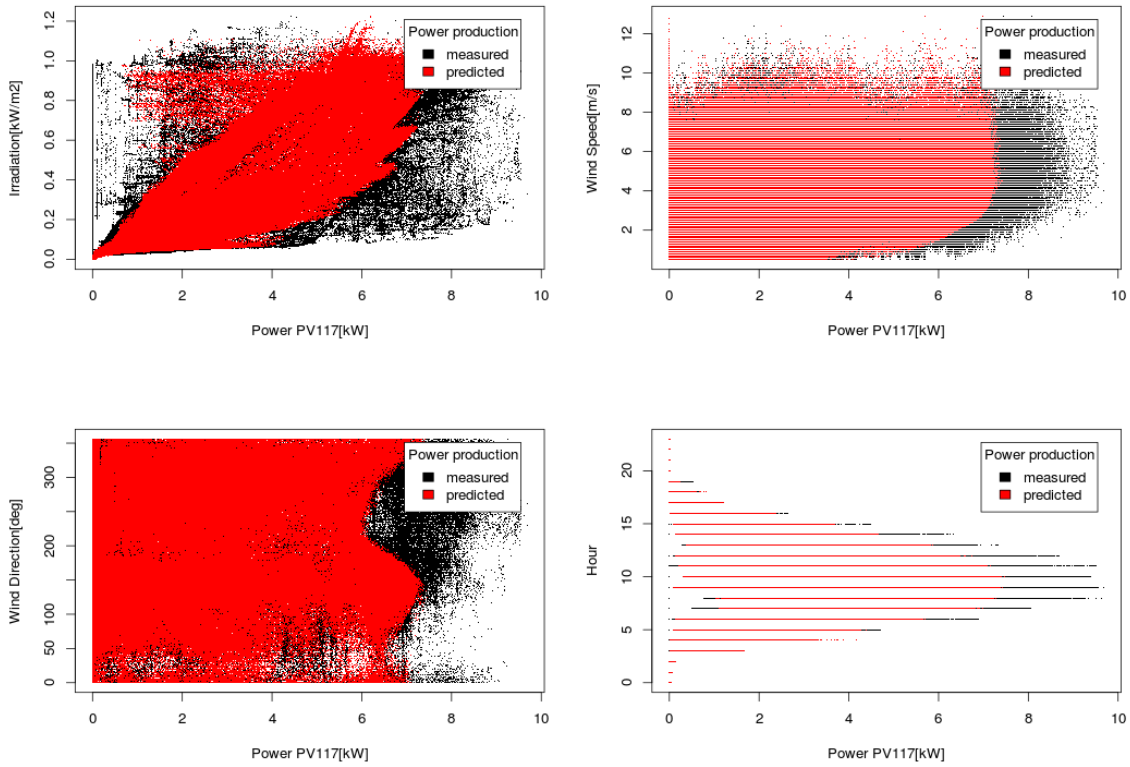

Figure 114: Model's prediction compared to the expected output for each input.

Controllable PV normal model

A complex controllable PV model was replaced with a simple model that inputs the output of the normal model active power production (power) and inverter set-point (spInv) and outputs predicted active power production of the PV in the context of the control signal (pMCtrl). In the controllable PV normal model, if the estimated power is higher than the setpoint, then the model outputs the setpoint, otherwise it outputs the estimated power. The difference between measured power, power estimated from normal and controllable PV normal models are presented in figure 115.

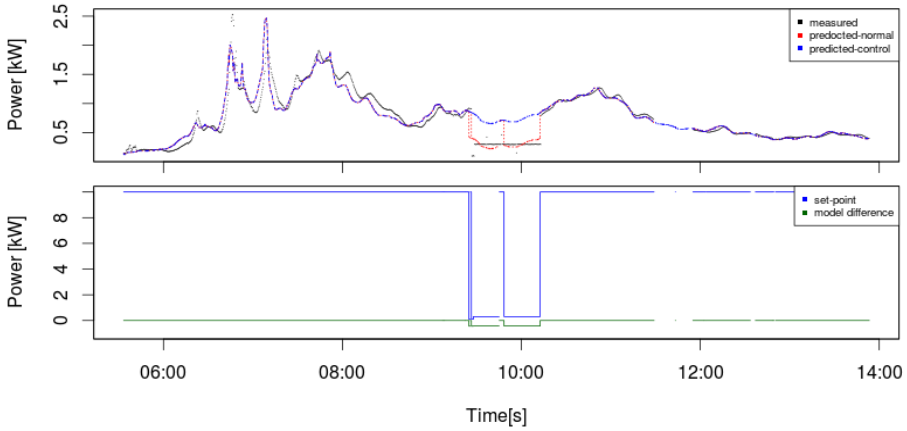

Figure 115: Comparison of the PV normal and Controllable PV normal models. The model difference is the the diference between normal and controller model.

Investigated attacks

The objective of the considered attacks is directed at the integrity of data while performing control on a PV plant. The general idea is that the monitor would discover if the PV is being

controlled and at the same time the data reported from the PV inverter is being actively modified in order to hide the effect or presence of the malicious control actions. Seven attacks were considered in this scenario:

- The set-point data is being modified in order to hide the control action
  - Attack A1: set-points (spInv and spCtrl) are modified to 110%
  - Attack A2: set-points (spInv and spCtrl) are modified to 90%
  - Attack A3: set-points (spInv and spCtrl) cleared
- OPC-UA client authentication and registration in OPC-UA server can be removed in order to hide that the external client is controlling the PV plant.
  - Attack A4: authentication removed
  - Attack A5: authentication and client registration removed
- modification of the power production data
  - Attack A6: active power production pInv modified 110%
  - Attack A7: active power production pInv modified 90%

In practice attacks A4 and A5 are difficult to execute on the OPC-UA server due to its security features, however the delivery of the alert could be delayed or the message could be lost in transit. Additionally the integrity of the event message could be compromised in transit.

Anomaly detection model trained with synthetic attack data

The training data set for the Anomaly detection model consisted of measured data at a time resolution of 1 second from a PV system at the SYSLAB laboratory, recorded during the period of 22nd to 24th of June 2016. The PV data was then edited to emulate the different attacks.

The classification model can output either a score: the probabilty for each class of anomaly (normal, controlled and malicious) - or a label: the anomaly class (normal, controlled and malicious). In order to determine the best type of classification model for the investigated problem among the models implemented in the nnet R package, several models were trained. Their confusion matrix, accuracy, precision and sensitivity were then used to select a suitable type of model.

The confusion matrix is a compilation of instances of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), evaluated from a population of results. To measure the correctness of the anomaly detection we calculate three significance measures: accuracy (ACC), precision (PREC) and sensitivity (SEN). Accuracy is a description of random errors, precision is the fraction of predicted instances that are relevant and sensitivity (also called recall) is the fraction of predicted instances that are retrieved. Accuracy, precision and sensitivity are calculated as follows:

$$ACC = \frac{TP+TN}{TP+FP+FN+TN}$$

$$\mathrm{PREC} = \frac{TP}{TP+FP}$$

$$SEN = \frac{TP}{TP+FN}$$

The following supervised classification methods were considered in order to train the anomaly detection model:

- log-linear: fitting multinomial log-linear models using artificial neural networks
- logit: logistic regression. Logistic regression can be calculated only for two classes, therefore the classes Normal and Controlled were combined.
- softmax: feed-forward neural network using the `softmax` function as the activation function in the output layer.
- softmaxSkip: recurrent neural network using the `softmax` function as the activation function in the output layer.
- entropy: feed-forward neural network using maximum conditional likelihood for training.
- entropySkip: recurrent neural network using maximum conditional likelihood for training.
- lout: feed-forward neural network using a linear function as the activation function in the output layer.

The confusion matrix, sensitivity, precision and accuracy scores for each model are presented in figure 116.

Algorithm	Class	TP	TN	FN	FP	SEN	ACC	PREC
log-linear	Normal	23497	211859	0	0	1.000	1.000	1.000
	Controlled	6	208683	26592	75	0.000	0.887	0.074
	Malicious	185186	23503	75	26592	1.000	0.887	0.874
softmax	Normal	23497	211859	0	0	1	1	1
	Controlled	15282	208654	11316	104	0.575	0.951	0.993
	Malicious	185157	38779	104	11316	0.999	0.951	0.942
softmaxSkip	Normal	23497	211859	0	0	1	1	1
	Controlled	15140	208314	11458	444	0.569	0.949	0.972
	Malicious	184817	38637	444	11458	0.998	0.949	0.942
entropy	Normal	0	211859	23497	211859	0	0.474	0
	Controlled	26598	208758	26598	185261	0.5	0.526	0.126
	Malicious	185261	50095	185261	26598	0.5	0.526	0.874
entropySkip	Normal	0	197761	23497	211850	0	0.457	0
	Controlled	12782	208758	26598	184970	0.325	0.512	0.065
	Malicious	184970	50095	185261	12782	0.500	0.543	0.935
lout	Normal	0	18181	127071	1667	0	0.124	0
	Controlled	0	144567	2352	0	0	0.984	-
	Malicious	0	129423	17496	0	0	0.881	-
logit	Controlled	0	48189	784	0	0	0.984	-
	Malicious	4184	42357	1648	784	0.717	0.950	0.842

Figure 116: Confusion matrix of the anomaly detection model trained with real data.

The neural network model has a single hidden layer with 10 neurons. The regularization parameter is set to decay = 0.0004. In the case when precision cannot be calculated, the sum of TP and FP is equal to zero.

As seen in figure 115, the simple logit and lout models perform badly for identifying true positives in the data. Both the entropy and entropySkip models recognize the Controlled and Malicious classes with an accuracy of 0.512 to 0.543 but score low on sensitivity and precision. The log-linear model can identify the Normal behaviour well, but the number of false negatives for the Controlled behaviour detection and false positives for the detection of Malicious behaviour is very high. Overall, the softmax model performs best: Only the sensitivity value in the Controlled class has a low score of 0.575. Changing to a recurrent ANN does not improve sensitivity, precision or accuracy. Therefore, the softmax model has been subsequently used for anomaly detection in the SALVAGE project.

Anomaly detection model with real attack data

For a test against real attack data, the same ANN model as in the previous section has been used: a single hidden layer with 10 neurons, decay = 0.0004. As in the previous case, the model was trained with PV data recorded from the SYSLAB laboratory.

Set	Normal	Controlled	Malicious	Total
Training	2856	350	662	3868
Validation	1318	459	456	2233

Figure 117: Test and Validation set class samples

The properties of the training and the validation data sets are presented in figure 117. The training set consists data recorded on July 8th and August 19th, 24th and 25th, 2016. The validation data was recorded on August 26th, 2016. The number of attack cycles of a length of around 1 minute for each attack type are presented in figure 118.

Set	No attack	A1	A2	A3	A4	A5	A6	A7	Total
Training	3207	64	129	160	90	69	73	76	3868
Validation	1777	94	43	82	49	1	43	144	2233

Figure 118: Test and Validation set attack samples

The ANN model was developed on a random sample of 80% of the training set and tested with a random sample size of 20% of the training set. The results of testing different ANN types are presented in figure 119. It is clearly visible that the softmax model achieves the highest sensitivity, precision and accuracy.

Algorithm	Class	TP	TN	FN	FP	SEN	ACC	PREC
softmax	Normal	1318	901	0	14	1.000	0.994	0.989
	Controlled	439	1618	20	156	0.956	0.921	0.738
	Malicious	286	1757	170	20	0.627	0.915	0.935
softmaxSkip	Normal	982	276	633	342	0.608	0.563	0.742
	Controlled	46	1589	119	479	0.279	0.732	0.088
	Malicious	79	1475	374	305	0.174	0.696	0.206
entropy	Normal	633	468	1270	275	0.333	0.416	0.697
	Controlled	39	2068	165	374	0.191	0.796	0.094
	Malicious	86	1780	453	327	0.160	0.705	0.208
entropySkip	Normal	623	508	1356	270	0.315	0.410	0.698
	Controlled	37	2068	165	487	0.183	0.764	0.071
	Malicious	123	1780	453	401	0.214	0.690	0.235
lout	Controlled and Normal	0	6131	568	0	0	0.915	-
	Malicious	0	5431	1268	0	0	0.811	-
logit	Normal	1194	174	421	444	0.739	0.613	0.729
	Controlled	0	2068	165	0	0	0.93	-
	Malicious	132	1317	321	463	0.291	0.649	0.222
log-linear	Normal	984	274	631	344	0.609	0.563	0.741
	Controlled	36	1702	129	366	0.218	0.778	0.090
	Malicious	115	1392	338	388	0.254	0.675	0.229

Figure 119: Confusion matrix of the AnomalyDetection model with real attack data, per classification method.

The selected softmax ANN model was used in the intrusion detection system. Overall, the softmax ANN model achieves a sensitivity of 0.627, an accuracy of 0.915 and a precision of 0.935. When the prediction for every attack is considered (as shown in figure 120), the lowest sensitivity is observed for attack A1 and A7, while the accuracy and precision is between 0.912 and 1.

Attack	TP	TN	FN	FP	SEN	ACC	PREC
None	1757	286	20	170	0.989	0.915	0.912
A1	36	2139	58	0	0.383	0.974	1
A2	33	2190	10	0	0.767	0.996	1
A3	69	2151	13	0	0.841	0.994	1
A4	49	2184	0	0	1	1	1
A5	0	2232	1	0	0	1	-
A6	43	2190	0	0	1	1	1
A7	56	2089	88	0	0.389	0.961	1

Figure 120: Confusion matrix of the soft-max anomaly detection model, per attack type.

The sensitivity metric measures how many relevant samples are selected, in other words how complete the results of the prediction are. The sensitivity can also be treated as the probability that a randomly selected relevant sample is retrieved in a search. The different sensitivity measures for each attack recognition points to an issue that the model represents some attacks better than other and therefore recognises relevant samples better.

Conclusion

The proposed intrusion detection system uses information about PV power production, meteorological conditions and cyber-security events to discover cyber attacks on PV remote control. The proposed method was verified with experimental data. Further improvements of the method mainly focus on the model improvements to decrease the method sensitivity in the attack recognition. In order to achieve this improvement, more varied attack data is

required for the model training. Additional long term monitoring tests are needed to assess the repeatability of the results presented in this report and applicability of the developed model to fall, winter and spring seasons.

1.5.6.13 Behaviour model of residential demand response

Residential demand response is maturing from a concept to real-world applications, and it is considered a significant resource of localized flexibility. In particular in cases where the heat and cooling needs of buildings are satisfied by electric heating or heat pumps. As demand response is maturing from a vision to real-world applications, it is also becoming a potential target for cyber attacks. A real-time demand response system is a cyber-physical system; therefore, there should be physical (non-ICT based) indicators of anomalous behaviour. In the following, we investigate the observable characteristics of individual household consumption with respect to real-time demand response.

The demand response behaviour is partly governed by physical properties of the process, partly by the autonomous behaviour of residents, and in part by the local control systems, which may be parametrized by local users. users. This combination of uncertain and in-transparent system properties leads to new challenges for reliability and security of operation. Further, the involved control systems are more diverse and open, which offers more entry points for cyberattacks. We investigate the feasibility of an online monitoring system characterizing the dynamic response behaviour of price-controlled demand. The goal is to formulate indicators of anomalous behaviour based on the observable characteristics of individual households. The investigations are based on a data set obtained by the EcoGrid.eu project [54].

As demand response requires a large number of typically quite diverse individual units, one cannot cannot expect direct and manual monitoring. Further, occupant privacy should be considered in such systems, thus give preference to the applications of aggregate, purpose-build detection models, and to avoid the use of individually traceable information. Based on experimentally observed data, we therefore aim to develop a modeling approach to detect specific kinds of “anomalies” in observable response dynamics.

Whether a cyber-intervention is the actual cause of anomalous behaviour cannot be inferred from physical models alone: They may simply be a reflection of changes in the inhabitants’ behaviour. The project has therefore followed a hypothesis-driven approach, which a) will account for more apparently goal-directed changes, and b) neglect commonly observed patterns of behavioural change.

In the previous sections, we have investigated contextual model-based anomaly detection for DER analysis applied to photovoltaic DER. There is however a fundamental difference in the dynamical characteristics of (residential) demand response and the PV models: instead of an algebraic input-output relation, the residential demand response is characterized by a) unobservable (random) behaviour of residents influencing both demand volume as well as parameters of the response characteristic, and b) the unknown thermal dynamics of the household, influenced by further exogenous parameters.

In previous work [55], [56] on characterizing the price-change responses from houses with smart metering equipment, the suggested approach has been to extract finite impulse response (FIR) coefficients from the price input to the demand.

The following steps are proposed:

1. General power consumption modelling. Models the part of the system that is not sensitive to the real time prices.

2. Online price sensitivity modelling. Models the price sensitivity, to reduce features of the price sensitivity signal for anomaly detection.

3. Anomaly detection and feature extraction based on the system online estimated parameters from step 2.

Based on these principles, the data processing concept shown in figure 121 has been developed. The model comprises an offline phase where a general system behaviour and dynamical response characteristics are modelled, and an online phase where system parameters are continuously identified and anomaly detection parameters are computed as input to an IDS.

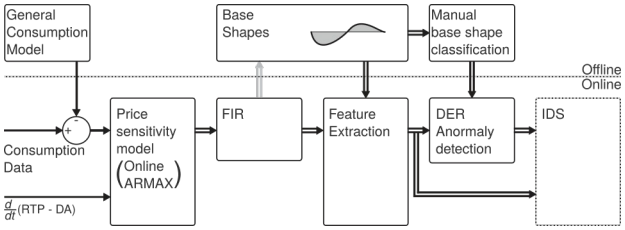

Figure 121: Conceptual outline of detection approach.

Methods

The methods employed belong to a very basic toolset of statistical methods: linear regression, ARMAX time series modeling and two clustering methods. The data set has been recorded in a large scale field demonstration of demand response.

1. Linear Regression: Linear regression is a well known approach for modelling a linear relationship between a set of input variables and an output. It has the nice property that it will always converge to the global minimum, if there are as many or more samples than unknowns. The general form is $y_t + \epsilon_t = x_t \theta_x$, where x_t , y_t and ϵ_t is the input, output and error, and t is the estimated proportionality constant [57].

Using this method, Larsen et al [58], [56], show that it is possible to extract FIRs to the real time market from another period of the Ecogrid EU data. With prices, weather and a set of Fourier terms as input to model the general behaviour and a difference model modelling the response from changes in the pricing. This provides a very condensed set of information at DER level, describing the system response to market price changes.

2. ARMAX system identification: ARMAX is a model for system identification. It estimates a linear model which on a transfer function form can be written as

$$y_t = \frac{B(q)}{A(q)} u_t + \frac{C(q)}{A(q)} e_t$$

A(q) is the system polynomial, B(q) and C(q) are the input polynomials of input and noise, where u_t , ϵ_t and y_t are the input. noise and output, respectively. As an extension of the linear regression, it represents the dynamics of the system by using delayed system outputs (auto-regression) and estimation of noise. It is possible to estimate the parameters of the polynomials in a recursive manner, creating an online system identification which can follow a changing system.

3. k-Means algorithm: The k-Means algorithm is a commonly used clustering algorithm for unclassified data. The algorithm associates data with a predefined number of clusters, iteratively minimizing the total distance from all the samples to the cluster centroids. This is achieved by alternating between associating the data samples with a cluster of the closest centroid and updating the cluster centroid as the mean point of the cluster. This algorithm is fast and does not require prior knowledge of the data.

Key to successful clustering in higher-dimensional sample spaces is the choice of a distance measure to calculate the distance between each centroid and sample.

4. Cosine distance: The Cosine distance is a measure applicable for the k-Means algorithm. It is not a true metric, but has proven useful as a measure of data with a high dimensionality where the direction of a sample vector is as important as the sample itself. The cosine distance builds upon the dot product of vectors, thus it captures the angular distance between the sample and centroid in the k-Means algorithm. It is defined as

$$D_c(A,B) = 1 - \cos(\theta) = 1 - \frac{A \cdot B}{(A)(B)}$$

where A and B are the feature vectors.

5. Gaussian mixture models: Gaussian mixture models take a more statistical approach to the clustering problem. The idea is to approximate n statistic sub-populations in a dataset. This is done by maximum a posteriori estimates. Like k-Means clustering, this is often done in iterative steps of ascending the likelihood function and updating the population estimates until convergence.

Dataset

The Ecogrid EU project was a research and demonstration project for a future smart grid across the European Union. It took place on the Danish island Bornholm during the years 2011 to 2015. Characteristic for Bornholm is a high penetration of renewable energy sources and the grid is almost separated from the mainland, with a single power line to Sweden. A corner stone of the Ecogrid EU, was therefore a real-time market for activating demand response from small scale DER [54]. The market generated 5-min realtime imbalance prices to which household level controllers would respond, adjusting electricity consumption of electric space heating. The houses involved in Ecogrid EU had a smart-meter monitor their power consumption in five minute intervals. For this work about 4 months of consumption data have been assessed: in total 1736 houses and 40033 datapoints, along with local weather data. There are, to our knowledge, no actual attacks on the power grid, local controllers or real-time market in the data used for this project.

Stationary responsiveness

In the offline characterization we follow the methods and approach presented in [55], [56], applying linear regression. The results for the baseline consumption models are very similar to the original work and will not be further discussed here. The goal of the price sensitivity model is to extract the demand response FIR from the power consumption data on DER level. Following [56], the price-responsiveness of household power consumption was modeled in form of a finite impulse response (FIR) to the real-time price variations. The input u_t for these models is the derivative of the difference between the real time price (RTP_t) and the day ahead price (DA_t). The output y_t is the consumption data of the household subtracted the prediction from a baseline model x_t . The baseline model for the data set has been created following [56] and is not further discussed here.

$$u_t = \frac{d}{dt}(RTP_t - DA_t)$$

$$y_t = c_t - x_t$$

The price-changes u_t are then time-lagged as input $u_t = \left[u_{t\ldots}u_{t-T_L}\right]^T$ where T_L is the time lag here chosen to be 200 minutes, or 40 samples. This yields the linear model

$$y_t + \epsilon_t = u_t^T \theta_{FIR}$$

The response characterization θ_{FIR} is a stationary model for an individual household.

Dynamic responsiveness characterization

An ARMAX model was implemented to emulate an online estimation of the dynanmic responsiveness. The order of the ARMAX model is chosen by trial and error on test data to be 16 for the system (A(q)), spanning 80 minutes; 24 for the input polynomial (B(q)), spanning 2 hours; 2 for the error dynamics, spanning 10 minutes. The characteristic time of the recursive parameter estimation algorithm is set to 14 days. This choice balances noise with the ability to follow changing system properties.

The output of the ARMAX model is a set of parameters for the three polynomials at each point in time. From each parameter set, a FIR can be approximated as the superposition of a FIR response with the input through the parameters in B(q) and an infinite impulse response (IIR) of the system through A(q).

Figure 123 illustrates the dynamic FIR as computed from the online ARMAX estimation for a single household.

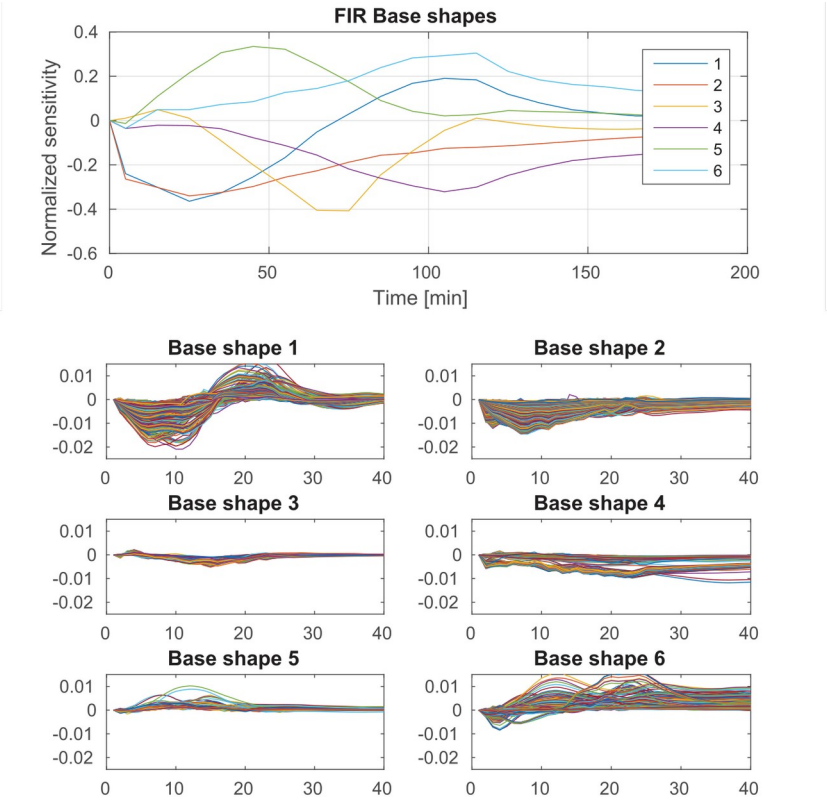

Figure 122: Top: Normalized FIR coefficients for each of the 6 cluster centroids(k=6). Bottom: Sample FIRs from each cluster (sample selection based on $conf_{controlled} > 0.5$ *and shape similarity* $\delta_{shape,i} > 0.5$ *). Note the x-axis corresponds to the coefficient index (40*5min = 200min) and y-axis is the absolute sensitivity.*

The upper graph plots the FIR coefficients by lag time, as presented in figure 122, and allows visual comparison with the stationary response shape of the linear regression model. The lower timeseries illustration provides a colourmap of the evolution of the ARMAX-estimated FIR parameters (vertical) along the time period of the modeling (horizontal).

This "online" ARMAX was only applied to a subset of 150 houses of the original data set. These were chosen by selecting a) the 50 least responsive houses and b) the 50 most responsive houses, along with c) 50 random draws among the remaining houses. The responsiveness was measured by three amplitude parameters: the integral, the maximum and the minimum value of the FIR.

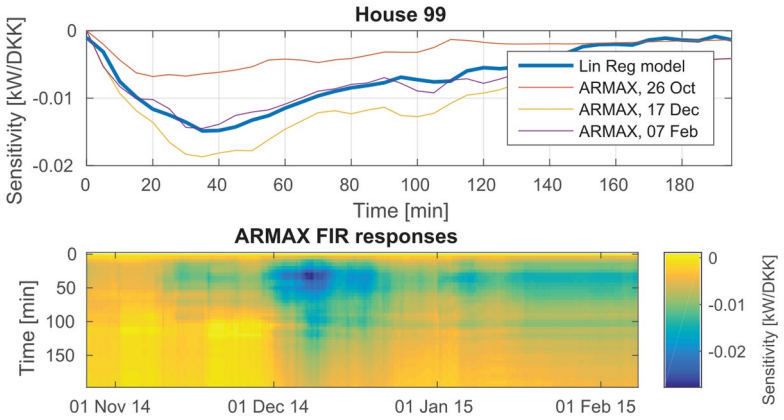

Figure 123: Illustration of ARMAX-based online FIR estimation. Top: comparison of stationary response with 3 ARMAX sample responses; bottom: variation ARMAX FIR over observation time (vertical axis: delay-time corresponding x-axis in top-plot). The response amplitude is intensified, but the characteristic shape remains.

Clustering of FIR

The set of stationary responses has been clustered by application of k-Means using a cosine-distance measure applied to normalized and down-sampled FIR parameters. The respective cluster centroids (normalised) are plotted in presented in figure 122 (top). The bottom plots in the same figure illustrate the (unscaled) FIR responses associated with the respective cluster. Each line represents one household FIR.

Characterization of anomaly and intent

The models outlined above are aimed to serve the characterization and identification of anomalous or even malicious behaviour in the demand response systems. To mitigate the high uncertainties and variance in the response behaviour, two types of metrics are applied:

1. a measure of the response amplitude or volume

2. a similarity assessment, comparing the observed behaviour shape to benevolent or malicious behaviours.

Using Gaussian Mixture (GM) models, the statistics of these metrics are then modelled for both "normal" and "undesired" behaviour.

1. Responsiveness measure: A histogram of the log magnitude of the three amplitude features for these 150 houses reveals three proto-distributions: the least controlled (uncontrolled) are separated from the most controlled group, with the random selection in the middle as expected (figure 124). As afirst measure, a filter was estimated fitting a Gaussian mixture model to the data set containing only the 100 most/least responsive houses. This filter yields the quantity $conf_{controlled}$, plotted as the black line in figure 126.

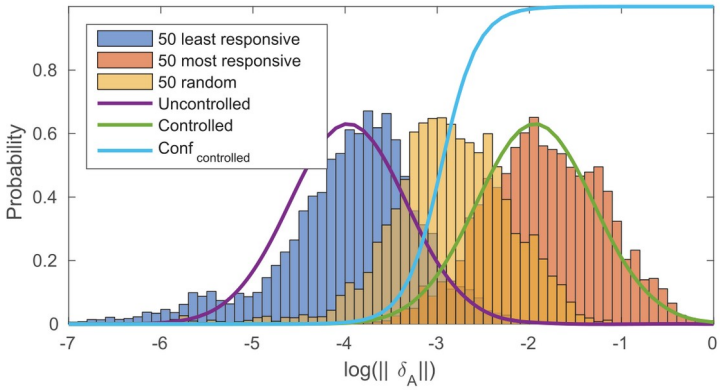

Figure 124: Histogram of $\log\left(\left(\left(\delta_A\right)\right)\right)$ *with characterization of the price-responsiveness amplitude. Overlayed is the Gaussian mixture*

2. Similarity to base shapes: Using the cosine-distance of the normalized response shapes, a measure similar to $conf_{controlled}$ can be derived based on the summation of distance features associated with the respective shapes. Two approaches have been formulated to assess similarity to base shapes.

   The first approach is based on the measure feature $\delta_{shape,i}$, defined as the inverse cosine distance to the base shape, normalized with the sum of the distance to all the shapes. The inverse relationship on the distance is penalizing long distances, while the normalization ensures that the features can be compared. This approach has been applied in the cluster allocation in figure 121. By manually grouping the base shapes into desirable and undesirable, this measure has been applied to compute the affinity with that response type: desirable (green: BS-1 to BS-4) and undesirable (red: BS-5 and BS6), as illustrated in the upper plots of figure 126.
   For the stable case ("House 99"), only little change in the response type is observed, even though the response amplitude is changing over time (figure 123). For the house undergoing an intervention ("Augmented house 23"), the measure becomes stable for higher amplitudes but is sensitive at low FIR amplitudes.
   The second approach uses a Gaussian Mixture Model to characterize a confidence for cluster allocation, similar to the responsiveness measure introduced above. To identify the 2D GMM with two centers, the cluster centroids were again associated either with desirable or undesirable base shapes; two features were then computed based on the sum of log-distances to either undesirable, (x_1) or desirable (x_2) base shapes. The identified GMM produces a more informed characterization of the classification of response samples by offering a confidence-level of the classification. Note that in preparation of the GMM base shapes, the k-Means clustering has been applied to a subset of the total data set, including only the houses with 20% highest responsiveness range to extract only significant response contributors, yielding a different set of cluster centroids (base shapes) than utilized in application of the first approach. The base shape numbers here are therefore also different.
   In the chosen approach to clustering, the engineering choices in sample selection and the manual base shape classification directly influence the detection outcomes. To evaluate the final detection effectiveness, reference cases would be required. With such reference cases, or analytic metrics, an automatic base shape classification could be developed.

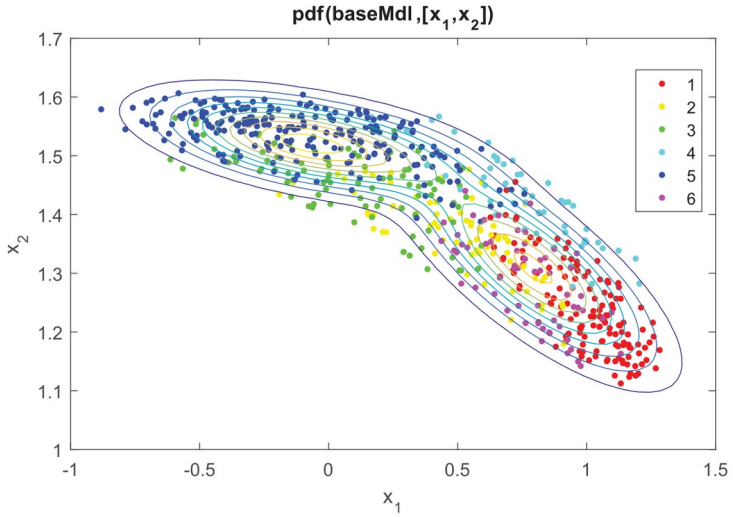

Figure 125: Scatterplot of sum of log-cos-distance to undesirable shapes x_1 *vs. sum of log-cos-distance to desirable shapes* x_2 *; the data points are coloured by associated base shape clusters (desirable BS: {1, 2, 4, 6} ; undesirableBS: {3, 5}). The data is overlayed with the two-parameter GM to be used for response type identification: upper-left: undesirable response shape, right: desirable response shape.*

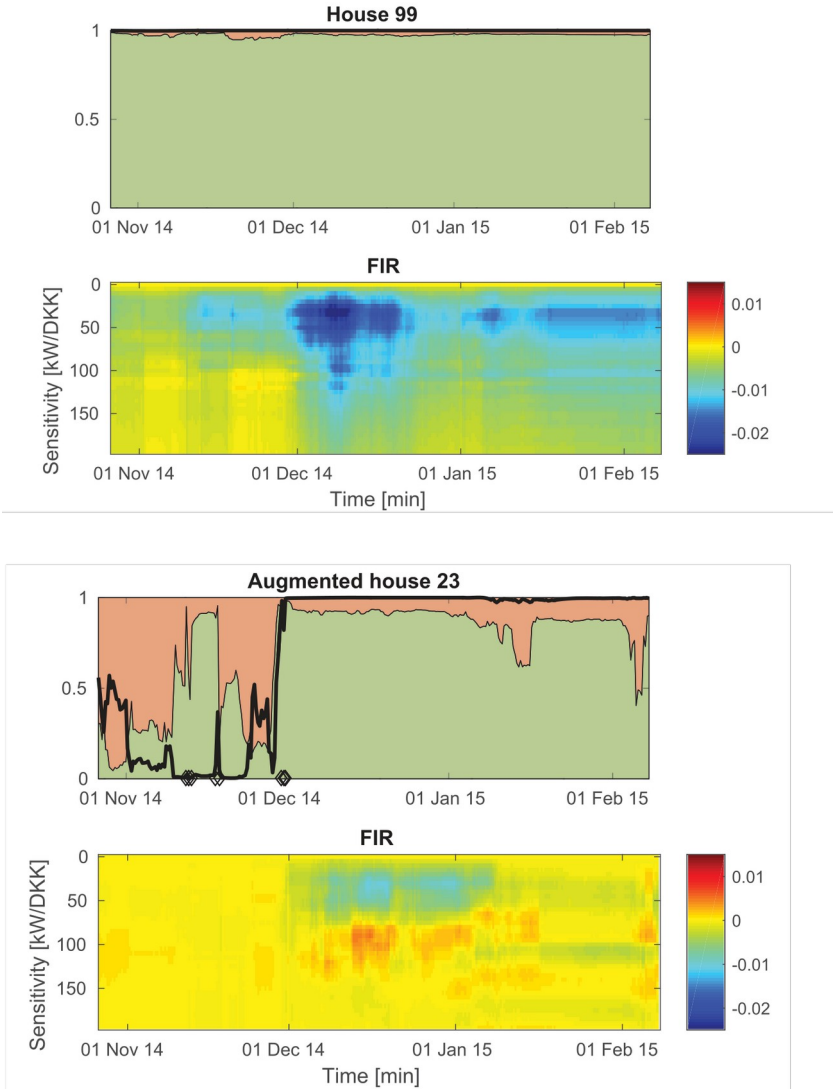

Figure 126: The ARMAX and anomaly detection applied to two houses: "House99" from fig. 123 and "Augmented house 23" with artificially mixed responsedata: shifting from low price-responsivenes to high price-responsiveness on Dec.1st.

3. Anomaly Detection: The implemented anomaly detection evaluates changes in the feature vector d δ_{FIR} . Every point in time is assigned a probability of it being an anomaly. Assuming the difference of the features is normally distributed, their standard deviations are estimated based on all 150 houses. These parameters are then used to calculate an anomaly probability at each timestamp per house.
   In figure 126, this feature is marked by a diamond, as can be seen in the lower plot set. Here, it is apparent that this anomaly detection is overly sensitive to low response amplitudes (observed false positives before Dec. 1st). Along the same lines, the similarity measure (separation line between red/green areas) reacts very sensitive in the low responsiveness period before Dec. 1st ($conf_{controlled} < .5$. On the contrary, it is rather stable in combination with high responsiveness, as can be observed in both examples in figure 126. This suggests a combination of the measures, e.g. by reducing the confidence in the similarity measure in dependence of another measure such as $conf_{controlled}$.
   Here it is worth noting, that in the case of integration into a cyber-physical intrusion detection system (CPS-IDS), the anomaly detection component will not be applied independently. Here the statistical measures outlined previously are better applied, as they deliver a continuous probability value, to be employed by the CPS-IDS hypothesis quantification component.

Discussion

Based on smart metering data, the behaviour of price-responsive control of loads can be monitored, and these observations may be integrated in a cyber-physical intrusion detection system (CPS-IDS). The load response behaviour was characterized by its Finite Impulse Response (FIR) behaviour. The wide variety of response shape indicates that relevant "anomalies" are not easily identified in the time domain behaviour of a group of loads, but engineering intuition was applied to classify the shapes of time domain behaviours observable in the data. This expert-based approach was employed to intuitively classify response types into “desirable” and “undesirable” features.

Statistical methods were then applied to detect and classify behaviour anomalies. Behaviour change has been formulated as a criterion for anomalies using two independent features: Response amplitude ($conf_{controlled}$) and a similarity measure. Both measures are formulated as a probability metric using statistically identified distributions, so that the observed probabilities can be employed in the further probabilistic reasoning step in the CPS-IDS for risk analysis.

The results demonstrate a feasibility of a statistical approach to integrating cyber-physical observations in demand response oriented intrusion detection system. Parameter identification using the chosen ARMAX technique requires about 24h of observation until convergence, which puts limitations on the integration in online CPS-IDS systems, but is in line with the time-scale of typical smart-metering data acquisition. The validity and accuracy of the developed statistical models has to be evaluated in future studies. Approaches to avoid the manual classification of response types should also be replaced by more principled metrics based from attack goals.

The reported monitoring for normal and anomalous demand response behaviour offers a number of possible applications beyond the CPS-IDS application outlined here, such as: system supervision and decision support, monitoring of an aggregator’s portfolio to estimate flexibility or monitor user behaviour, or validation of a contracted response.

1.5.7 Integrated analysis framework

1.5.7.1 Proposed approach

Traditionally, and from the point of view of a control room operator in an electrical power system, the ICT aspect of the power system (which includes the SCADA and DMS systems) has not been part of the system to be operated; it was an infrastructure which was assumed to "just work". Today, the ICT domain (both OT and IT) is explicitly taken into account during operations. However, the pure ICT risk assessment still tends to separate security relevant events from ICT and physical domains.

Our approach assumes an operational context in which the risk of several cyber-security breaches are evaluated at the same time, and where there is highly uncertain information about possible security breaches. In such a context, only an integrated assessment is meaningful, where a risk-oriented prioritization of potential threats and impacts is required to accommodate probabilistic information.

Background

Cross-dependencies between domains make it hard to isolate the impact analysis of the physical and cyber domains. Consequences of IT-domain breaches which manifest themselves in the physical domain, are not quantifiable using the same metrics as a pure analysis of the IT domain. Furthermore, the model types and propagation mechanisms are different in each of these domains. Different propagation mechanisms make a direct model integration impossible; a coupling of models could however be achieved, similar to the strategy for co-simulation approaches. A pure co-simulation based assessment would

however require a full model parametrization, simulation of combined models and result assessment. Probabilistic input hypotheses and dependencies would still not be feasible in such an approach. Nevertheless, the good success of probabilistic modeling in – for example – intrusion detection or ICT architecture assessment (CySeMol), indicates that uncertain knowledge from several domains can successfully be combined. Attack modeling methods based on probabilistic methods and tree structures are quite successful [59].

However, attack probability alone does not constitute sufficient information if the goal is to be able to prioritize the response to several attack scenarios. The impact quantity, i.e. the consequence of an attack in terms of the physical system, is of equal importance. Calculating the overall risk of a particular attack scenario (asfunction of probability and potential impact) cannot be done without integrating an impact analysis into the vulnerability analysis process, which is necessarily domain-specific.

Problem statement: Key requirements

We see a need for a framework that performs a real time (online) integrated assessment of the state of the power grid, using aspects from multiple domains (power system impact, intrusion detection, cyber vulnerability) and multiple sources of input (power system measurements, distributed energy resources (DERs), IT and OT systems) – a framework that aims to provide a joint prioritization of possible threat/impact-scenarios, taking into account uncertainty of input information.

We see that an automated, integrated assessment has the potential to yield more accurate results than performing several separate assessments whose results need to be interpreted and finally merged and synthesized by a human. The strength of the integrated assessment resides in

1. the possibility to handle dependencies that cross boundaries of different domains in an automated fashion and already during the assessment process; and hence

2. the ability of the framework to operate in real time (online assessment), and therefore the ability to provide frequent re-assessments based on an actual state of the power grid and its IT/OT infrastructure.

Core concepts

The system configuration is the model of the underlying system that is being evaluated – including the power grid, distributed energy resources attached to it, and all relevant pieces of IT/OT infrastructure that govern and/or otherwise interact with any and all of the former.

The term Domain, in the context of this paper, is referring to one of the following: cyber security/vulnerability, DER intrusion detection, or power system [impact] analysis. Figure 127 shows the relationships between the domains, expressed in terms of the possible flows of information between them.

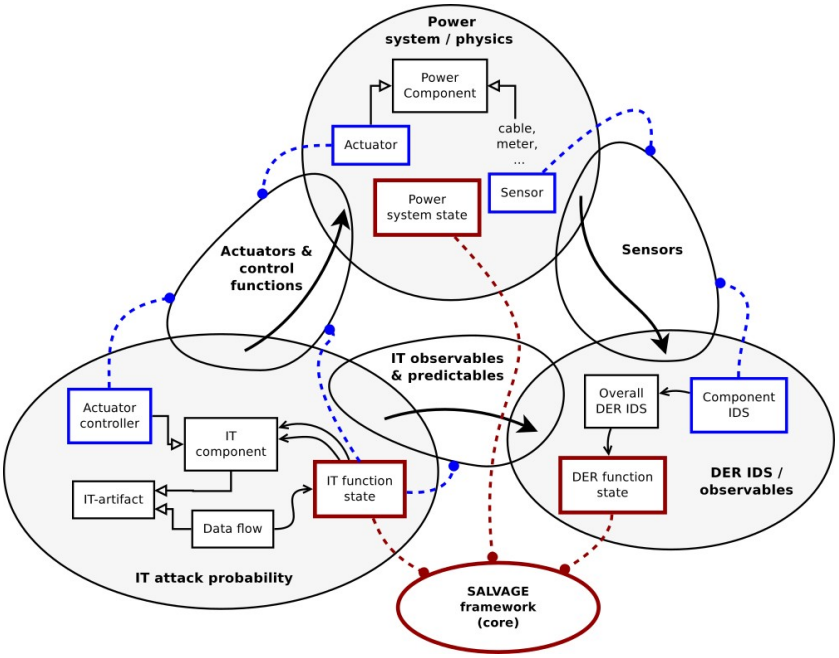

Figure 127: Domain interactions

A domain-specific assessment (DSA) is a quantification (function) obtained from a computational component. For a given input parametrization, the DSA quantifies one or several (branching!) pairs of a probability (p) and a different quantity (a Domain specific variable, e.g. electric current, operating state, switch state). The input parametrization may include both static parameters (e.g. system configurations), dynamic online data (e.g. meter readings), and other [dynamic] invocation parameters (e.g. state of a particular switch or function). Example: e.g. given a certain state to occur (DER on or off), what is the quantitative impact (power distribution), and what is the chance of it occcuring.

A domain specific variable (DSV) is a quantity (an intermediate or final impact value, e.g. electric current) that is specific to a DSA, and can be obtained by the DSA, and may be a (partial) input parametrization of another DSA.

A Hypothesis template is a tree-like structural representation (strictly taken, an acyclic directed graph) of the following:

1. An “attack tree” in form of a probability tree using logical gates, which aggregates probabilistic data from DSA inputs at its leaf level all the way up to the top-level probability of the assessed realism of a hypothesized scenario;

2. A set of domain-specific variables (DSVs, e.g. power supply loss due to opening of a specific breaker) related to a system configuration – the “output quantities” of DSAs, which, according to the structure of the hypothesis template, ultimately contribute to the risk value (the aggregated product of partial probabilities and impacts) of the hypothesized scenario. The DSVs can be defined together with an “input parameterization” of a DSA, since the DSA might need to be dynamically invoked with the output parameters of another DSA (instead of the baseline system configuration);

3. The coupling of output probabilities of each DSA to the probability tree (through logical gates such as AND, OR, XOR and NOT);

4. The coupling of DSA outputs with invocations of other DSAs (at a higher level in the tree), leading to a final scenario quantification.

The fisk node is a node used as the root node of a hypothesis template, joining the final impact quantity (associated with an attack goal) with the final hypothesis probability into a single, overall risk value of the (grounded) hypothesis.

A Risk belief is the quantified risk value of a hypothesis (hypothesized scenario).

The Hypothesis probability (belief) is a fully quantified probability value of a hypothesis.

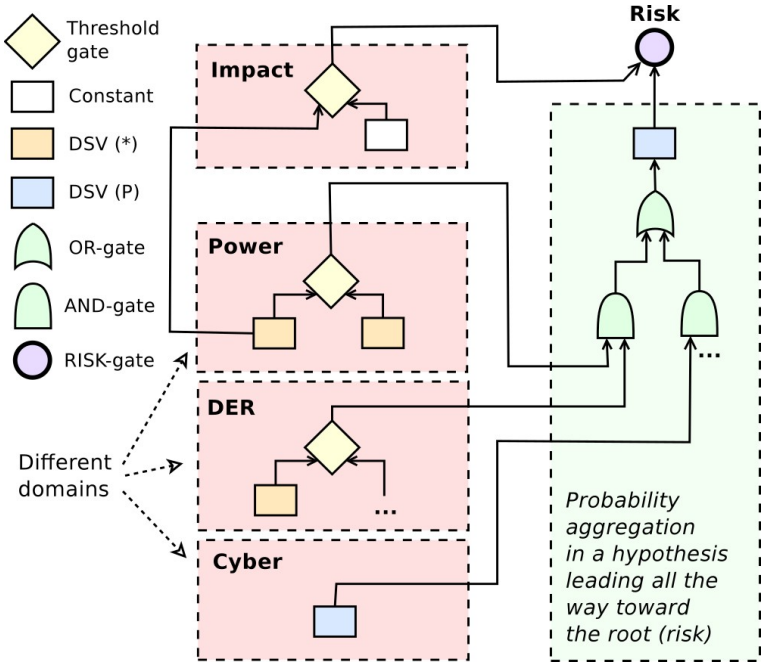

Figure 128: Conceptual structure of a hypothesis template and resolution

Figure 128 illustrates the concepts defined above and their relations. In a hypothesis, probabilities and other quantities are aggregated through functional gates all the way up to the root value of a hypothesis which represents its risk value, i.e. the product of the impact of an adverse event and its probability.

1.5.7.2 Proof of concept: The fuse blowing scenario

In the following, we will demonstrate the feasibility of the above approach. As discussed in section 1.5.7.1, the initial key issue to be solved is related to the joint assessment across multiple domains rather than the size of the system to be analyzed. We start with the test system shown in figure 129 which represents a multi-domain system while keeping the complexity within the individual domains at an absolute minimum, i.e. a trivial grid configuration, a trivial IT network and very simple DER behaviour.

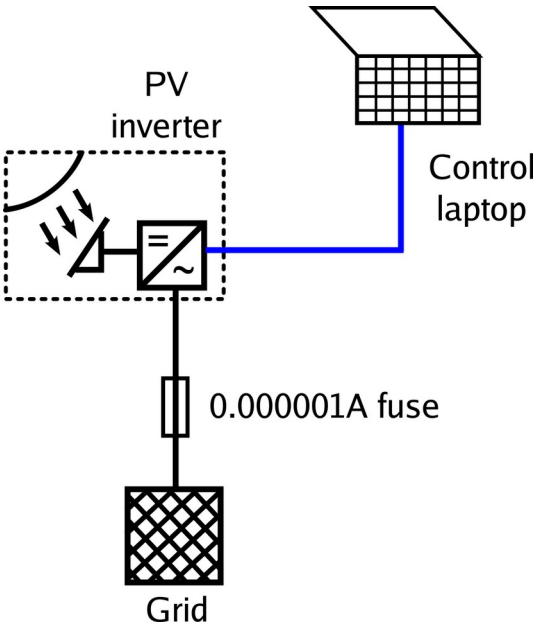

Figure 129: Fuse blow scenario

The test grid configuration consists of a PV inverter connected to an electrical power grid of infinite capacity. The inverter can be remote controlled from a laptop (equipped with remote control software) through a minimalistic communication network consisting of a direct, unswitched ethernet connection between a laptop and the controllable inverter. A single inverter function is available through the remote-control interface: Switching the inverter on — which would cause the inverter to feed power to the grid, depending on current solar irradiation — or off, which would reduce the current flowing between inverter and grid to exactly zero.

The cable between inverter and grid is protected by an infinitely small fuse (here: 1µA) , such that the fuse would inevitably blow and disconnect the inverter from the grid as soon as the inverter is switched on. This reduces the range of possible grid impacts of a cyberattack to a binary choice: Turning the inverter on will result in permanent damage to the system; the associated need for repair is easily quantifiable in terms of financial damage. The only alternative action is to leave the inverter in the off state, which has no consequences.

Processing the fuse blow scenario

Applying the method described above results in the hypothesis template shown in figure 130.

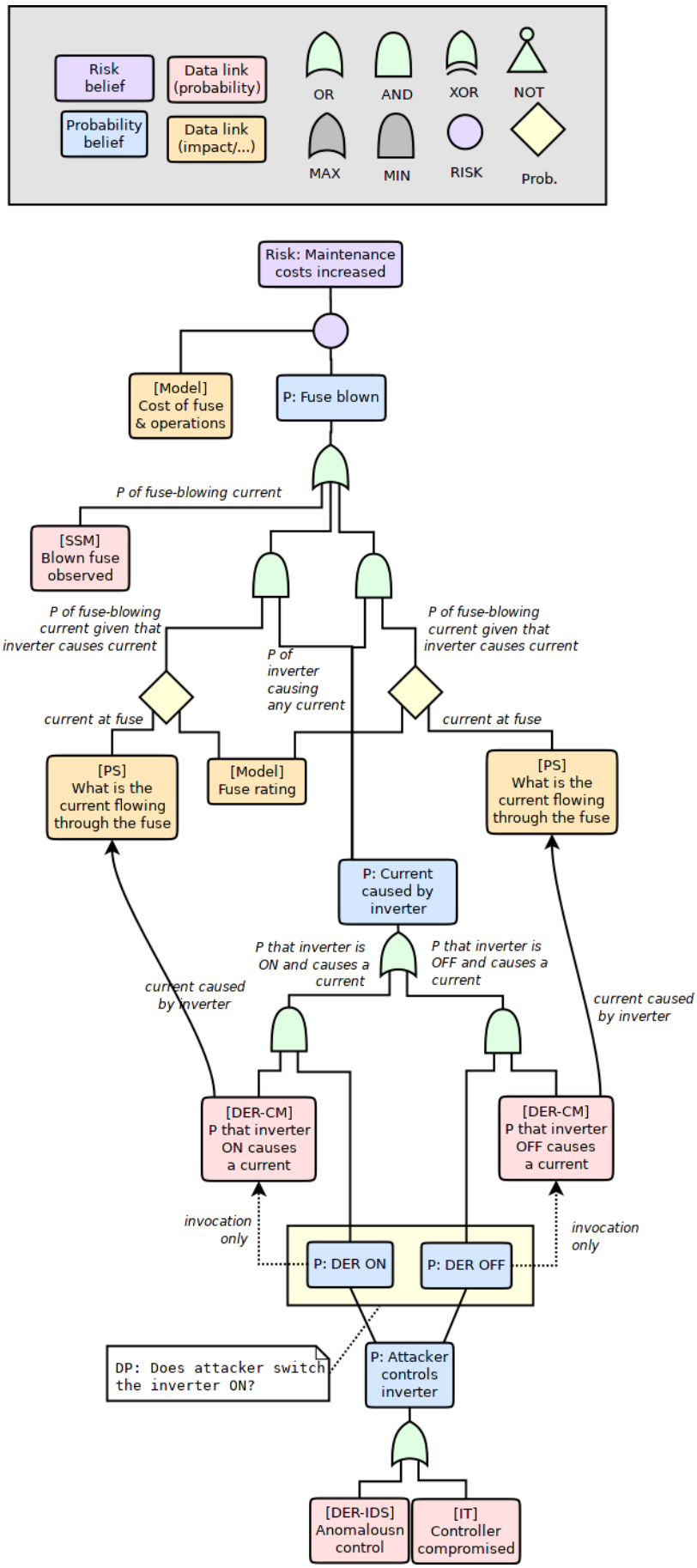

Figure 130: Hypothesis template for the fuse blow hypothesis

Starting at the bottom of the tree, the first two inputs to the resolver are the outputs of the DER intrusion detection tool [DER-IDS] and the vulnerability analysis tool [IT]. Combined, they

express the probability of the inverter having been compromised. At the next level, the actions of the attacker must be considered; specifically what the attacker intends to do with the compromised system.

Due to the design of the test case, the attacker has only two options available: switch the inverter on, or leave it off. These two options, while the probabilities for their respective occurrences may be known, lead to a widely different outcome for the physical system. Therefore, two different hypotheses must be generated from the hypothesis template, and each outcome must be evaluated separately. For each hypothesis, the response of the inverter to attacker behaviour must be evaluated using a component model of the inverter [DER-CM].

The output of the component model is twofold and consists of (a) the magnitude of the electrical current at the inverter terminals in case of the inverter switching on, and (b) the probability of the inverter switching on after receiving a remote switching command. The current magnitude is then inserted intosystem model [PS] which calculates the line flow between inverter and grid connection, thereby determining the current through the fuse. By comparing this result with the fuse characteristics, it can be transformed into the probability of an inverter operation causing a fuse blow, which in turn can be combined with the probability of an inverter operation to yield the probability of a fuse blow.

Independent of the latter, deduction-based result, a blown fuse may also be observed by a system monitoring the state of the physical system in real time [SSM]. The disjunction of both yields the overall probability of a blown fuse. Multiplication with the calculated impact — here expressed as the financial damage caused by the required replacement of the fuse — yields the risk value.

The scenario presented above is very simplistic. Nevertheless, the application of more complex scenarios is not expected to lead to notably greater tree depths in the hypothesis template. Rather, they will lead to increased width of the trees, as the multiplicity of components in a physical system is taken into account. A practical assessment system would be built around a library of hundreds or thousands of different hypothesis templates and perform re evaluation on a continuous basis. This would provide a human operator or another information system with a frequently updated set of the most threatening scenarios according to the present state of the power grid and its cyber infrastructure.

1.5.7.3 A larger scenario

While the above scenario has been useful to illustrate the general approach and the concept of a hypothesis template, it is too simplistic and too far removed from an actual application to serve as a full proof of concept. In the following, we will discuss the application of the proposed approach to one of the base scenarios introduced in section 1.5.2.

Figure 131 contains the grid configuration of the ”protection and control scenario” (scenario 2). We analyze the attack case in with an intruder is able to manipulate the configuration of the protection system in order to change the selectivity of the protection arrangement. In a correctly configured protection system, each of the overcurrent trip relays belonging to circuit breakers B-1a, B-1b and B-1c should trigger before the overcurrent relay belonging to circuit breaker B-1. Likewise, circuit breakers B-1ba and B-1bb should trip before circuit breaker B-1b. In our attack case, the attacker is able to change the trip curves of one or more of these relays such that a fault in a lower-level branch of the grid (here: at the location marked in red) is not contained by a trip action of the nearest circuit breaker (here: B-1ba) but leads to the tripping of circuit breakers in higher-level branches of the grid (here: B-1b and/or B-1). This would lead to an unnecessarily large loss of load, as non-faulty branches of the grid are disconnected.

Like other attack scenarios involving protection malfunction, the proposed analysis method is difficult to apply to attack prevention, as the observation of system behaviour is only possible

within the very short timeframe in which protection action is executed (i.e. milliseconds). In these cases, the proposed method is expected to be most useful for a fast post-mortem analysis of a successful attack, and for the prevention of repeat attacks.

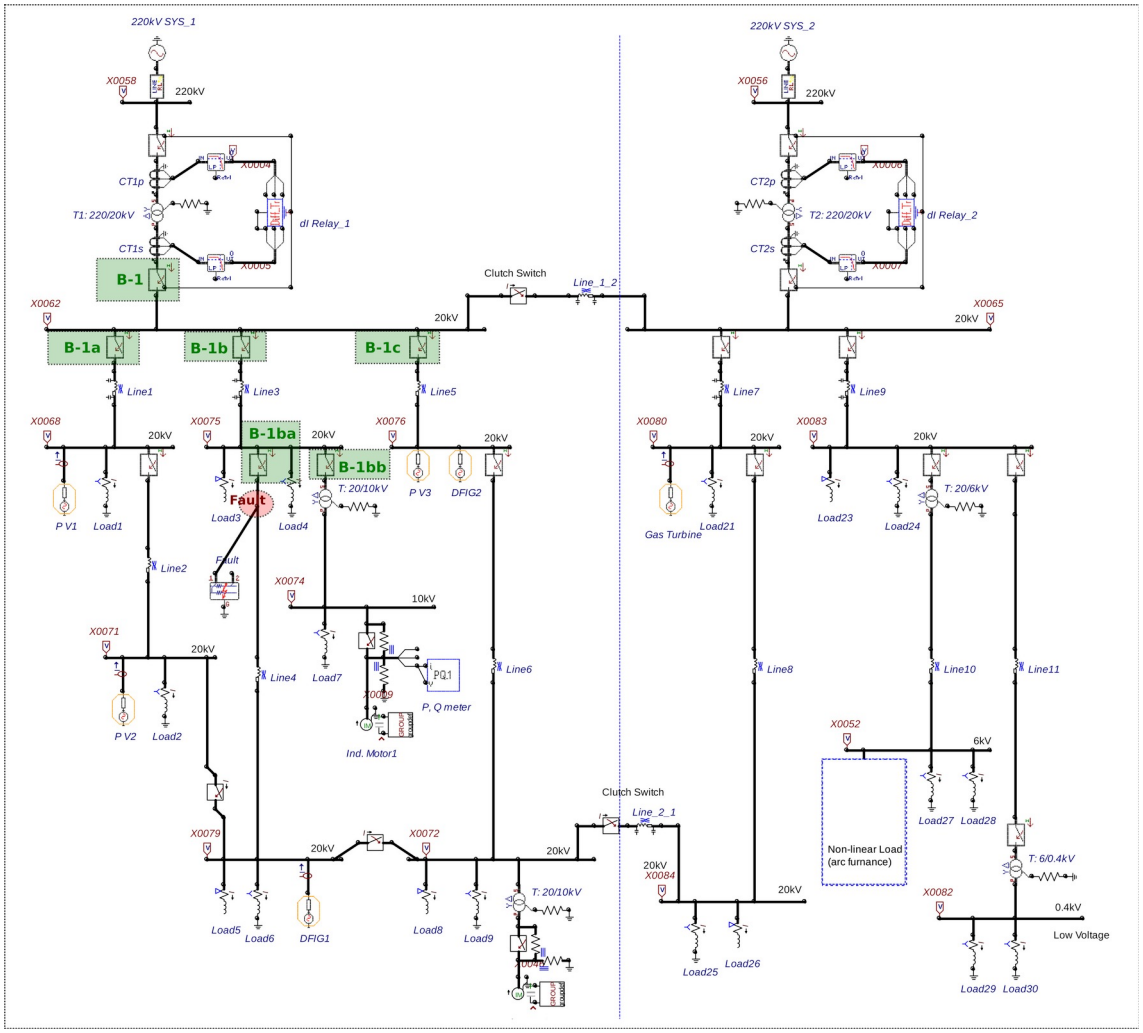

Figure 131: System configuration for the "selectivity hypothesis". The fault location is highlighted in red; all relevant circuit breakers are highlighted in green.

Figure 132 shows the resulting hypothesis template for the selectivity attack. It can be seen that the complexity of the template is not significantly higher than that of the trivial case.

The template can be broken down into four distinct processing stages. Identically to the trivial scenario, the first stage calculates the probability of an intrusion taking place, while the second stage considers the probabilities of different possible actions by a successful attacker. One of the main differences between the scenarios is the much larger degree of freedom enjoyed by the attacker: While the trivial scenario only allowed a binary choice (inverter on or off), the selectivity scenario offers a multitude of combinations of individual overcurrent relays and their settings. This is reflected in to levels of branch points (relay selection and setpoint selection) and, consequently, a much larger number of potential branch points.

The third stage of the template determines the potential impact of each branch point on the power system, while the fourth stage – just as in the trivial scenario - combines all branching options into an overall calculation of consequences and risk.

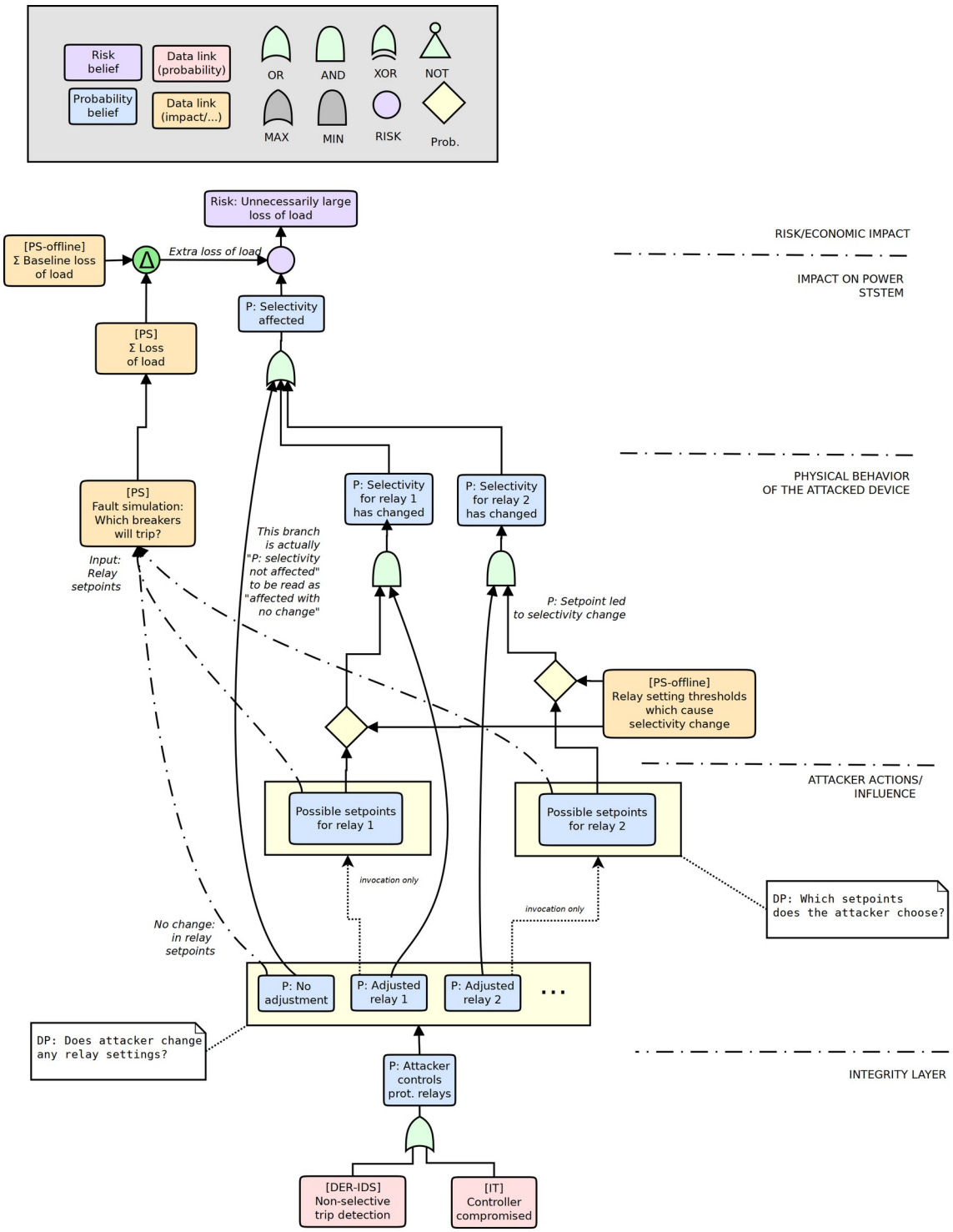

Figure 132: Template for the "selectivity hypothesis"

While the template as above exhibits a similar level of complexity compared to the trivial scenario template, the processing effort is greatly increased due to the much greater number of branch points, each of which requires a separate calculation of the power system impact. This example shows the current limitations of the presented approach with respect to the resolution of branch points.

1.5.7.4 Practical implementation

As part of the SALVAGE project, the process of hypothesis generation (expansion) from a hypothesis template, as well as the evaluation and ranking of multiple hypotheses, has been automated by developing a generic framework. Figure 133 shows the high-level architecture of this framework.

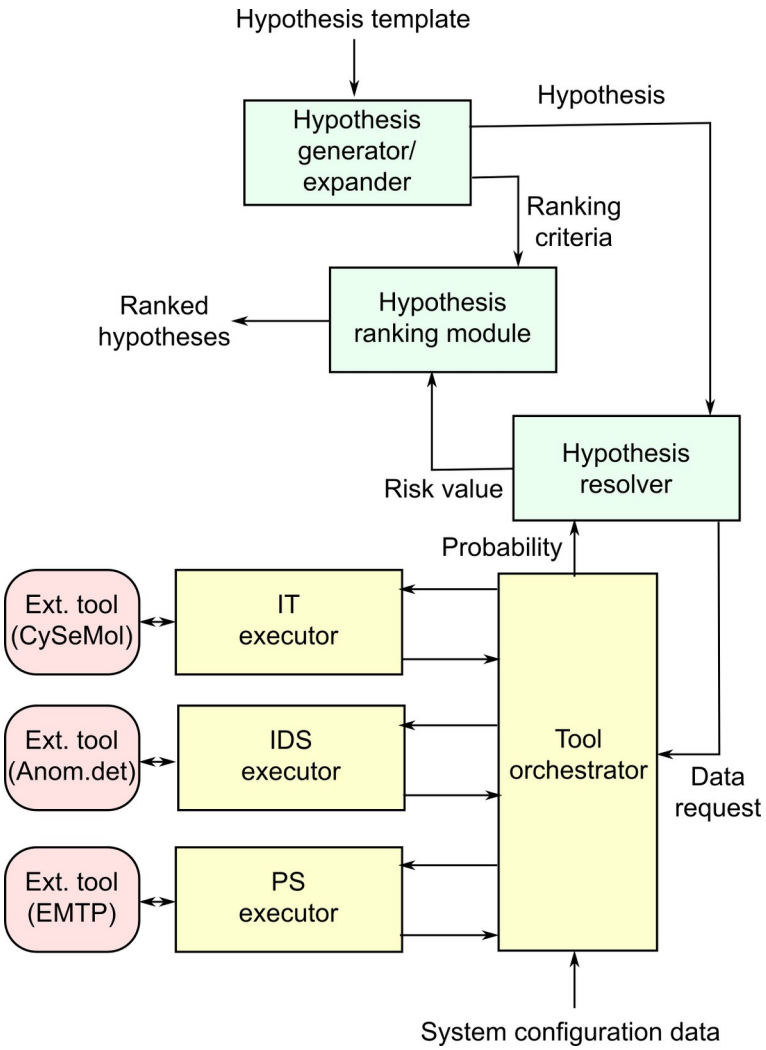

Figure 133: Framework architecture and data flow

A hypothesis template (represented by a hierarchically structured file) is given as input to a hypothesis generator. The generator analyzes the template and extracts a list of branch points which require expansion into multiple hypotheses. Furthermore, the generator identifies the type of information represented by the root node and instantiates an appropriate ranking algorithm for this type of node.

As a next step, the generator expands the template at all n_b branch points by filling in one of the possible states at each branch point, thus sequentially generating 2^{n_b} hypotheses. Currently, this simple brute-force approach to hy pothesis generation/expansion is the only available strategy. Each hypothesis is being forwarded to a resolver module which must match the hypothesis template. The resolver module traverses the tree and identifies the dependencies between ordinary tree nodes and data obtained by executing one of the domain-specific tools (e.g. a power system impact analysis). At each point of the tree traversal where such external input is needed, a data request is forwarded to the tool orchestrator module which coordinates the execution of one or more domain-specific tools, prepares the input data for these tools and collects their output.

The execution of each individual tool is encapsulated into an executor module to isolate the rest of the framework from tool-specific execution modes such as shell invocations, platform dependencies etc. Once the tree has been traversed, the value of the root node is passed on to the ranking module, which applies the ranking criteria provided by the hypoth esis generator. The ranking module will only terminate when the last hypothesis has been generated and resolved. Its output is a list of (the highest ranking) hypothesis ordered by their risk value.

1.5.7.5 Discussion

In the above, we have discussed and presented an approach for the integration of cybersecurity tools from multiple domains into an overall risk assessment tool which takes the complex interactions between domains in smart grid systems into account. We have also presented a very simple example case to serve as a proof of concept of the chosen approach, as well as a generic software framework for the processing of hypothesis templates. Furthermore, a more realistic case has been investigated which investigates possible attacks on the configuration of protection devices in a medium voltage grid, based on one of the SALVAGE base scenarios. The hypothesis template for this larger scenario points to one of the major challenges of the chosen approach: In most real-world scenarios, successful attacks on an IT infrastructure will typically offer a multitude of manipulation options to the attacker. The resulting hypothesis templates will therefore have multiple – often many - branching points, leading to a potential ”explosion of hypotheses” which would require a lot of computing resources to analyze. They would become impractical very quickly if brute-force solving of all branches remains the only available strategy. The scenario is a useful starting point for investigating practical mitigation methods in order to increase the efficiency of the proposed method. The applicability of the presented approach will to a significant degree depend on solutions for this issue.

Both the solution approach and the software implementing it are currently at an early stage of development. A number of questions remain open. The current framework can only handle single actions by an attacker (i.e. the attacker model is somewhat simplistic). It is nor clear yet how the possibility for multiple attacks could be represented in the overall risk figure. Further down the road, the partial automation of hypothesis template design presents an interesting challenge. Currently, hypothesis templates have to be hand-crafted by a group of domain experts covering all domains. Identifying some low-hanging fruits to reduce the effort would strengthen the viability of the approach.

We see that an automated, integrated assessment has the potential to yield more accurate results than performing several separate assessments whose results need to be interpreted and finally merged and synthesized by a human. The strength of the integrated assessment resides in (1) the possibility to handle dependencies that cross boundaries of different domains in an automated fashion and already during the assessment process; and hence (2) the ability of the framework to operate in real time (online assessment), and therefore the ability to provide frequent re-assessments based on an actual state of the power grid and its IT/OT infrastructure.

1.5.8 Cybersecurity workshops

In a joint effort with the EU FP7-funded projects SEGRID and SPARKS, the SALVAGE project organized two full-day workshops on cyber-physical security and resilience in smart grids (CPSR-SG). Each workshop was held in connection with the Cyber-Physical Week, in Vienna on April 12th, 2016 (CPSR-SG 2016) and in Pittsburgh on April 21st, 2017 (CPSR-SG 2017). In both cases, the organization of the workshop was facilitated by the CPS week organizers and the coordinators of the three projects acted as program chairs. Program committees were formed from researchers within the three project consortia.

15 papers were submitted to CPSR-SG 2016 of which 11 were accepted. For CPSR-SG 2017, 12 papers were submitted and accepted. Both workshops were successful, with good attendance during the day and fruitful discussions.

The accepted papers for the 2016 workshop were published by IEEE, in the IEEE Conference Publication Program (CPP), ISBN 978-1-5090-1164-3 (http://ieeexplore.ieee.org/document/7684093/).

The accepted papes for the 2017 workshop were published by ACM, ISBN: 978-1-4503-4978.

1.5.9 List of publications attributed to the project

- Czechowski, R.: "*Security Policy for Low-voltage Smart Grids*". Present Problems of Power System Control, vol. 5, p. 55-72, Oficyna Wydawnicza Politechniki Wroclawskiej, 2014

- Czechowski, R.: "*Cyber-physical security for Low-Voltage Smart Grids HAN security within Smart Grids*". *16th International Scientific Conference on Electric Power Engineering (EPE)*, Kouty nad Desnou, 2015, pp. 77-82. doi: 10.1109/EPE.2015.7161077

- Czechowski, R.; Wicher, P. and Wiecha, B.: "*Cyber Security in communication of SCADA systems using IEC 61850*". Modern Electric Power Systems (MEPS2015), Wroclaw, 2015, pp. 1-7. doi: 10.1109/MEPS.2015.7477223

- Rosołowski, E.; Iżykowski, J. and Czechowski, R.: "*Smart Metering as a means to increase power reliability and meet the demand for electric energy*". Nowoczesna Automatyka Zabespioeczeniowa i Systemy Sterowania i Nadzoru w Sieciach Elektroenergetycznych, MiCOM 2015

- Blom, R.; Korman M.; Lagerström, R. and Ekstedt M.: "*Analyzing Attack Resilience of an Advanced Meter Infrastructure Reference Model*". Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG), Vienna, 2016. pp. 1-6. doi: 10.1109/CPSRSG.2016.7684095

- Czechowski, R. and Kosek, A.M.: "*The Most Frequent Energy Theft Techniques and Hazards in Present Power Energy Consumption*". Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG), Vienna, 2016. pp. 1-7. doi: 10.1109/CPSRSG.2016.7684098

- Kosek, A.M.: "*Contextual anomaly detection for cyber-physical security in Smart Grids based on an artificial neural network model*". 2016 Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG), Vienna, 2016. pp. 1-6. doi: 10.1109/CPSRSG.2016.7684103

- Kosek, A. and Gehrke, O.: "*Ensemble regression Model-Based anomaly detection for Cyber-Physical intrusion detection in smart grids*". IEEE Electrical Power and Energy Conference (EPEC), Ottawa, ON, 2016. pp. 1-7. doi: 10.1109/EPEC.2016.7771704

- Korman, M.; Lagerström, R.; Ekstedt, M. and Blom, R.: "Technology Management through Architecture Reference Models : A Smart Metering Case". Portland International Conference on Management of Engineering and Technology (PICMET), Honolulu, HI, 2016. pp. 2338-2350. doi: 10.1109/PICMET.2016.7806518

- Korman, M.; Välja, M.; Björkman, G.; Ekstedt, M.; Vernotte, A. and Lagerström, R.: "Analyzing the Effectiveness of Attack Countermeasures in a SCADA System". Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids, Pittsburgh, PA, 2017. pp. 73-78. doi: 10.1145/3055386.3055393

- Gehrke, O.; Heussen, K. and Korman, M.: "Integrated Multi-Domain Risk Assessment Using Automated Hypothesis Testing". Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids, Pittsburgh, PA, 2017. pp.55-60. doi: 10.1145/3055386.3055398

1.5.10 Fulfillment of objectives

The project fulfilled three out of its four partial objectives:

- The novel concept of DER intrusion detection was investigated from the idea stage all the way towards a laboratory demonstration of several detection algorithms.

- The capabilities of the attack graph-based cyber vulnerability assessment framework CySeMoL were extended to include ICT infrastructure specific to low-voltage distribution networks.

- A novel method for the integrated assessment of cybersecurity in a smart grid context was developed and evolved towards a proof of concept, combining power system impact, intrusion detection and cyber vulnerability information.

The fourth partial objective, the development of a tool set for quantifying the potential power system impact of various cyber attacks, was not fulfilled. This was mainly due to the non-performance of the responsible project partner PWR. (See section 1.4.5 for further discussion).

1.6 Utilization of project results

The current state of the project results can be best characterized as belonging to Technology Readiness Levels (TRL) 2 through 4, i.e. some parts of the outlined technologies exist as a solid concept while for others, a proof-of-concept (PoC) has been conducted. In some cases, the PoC has been validated in a laboratory environment. This is not unexpected given the project's exploratory nature.

Looking at the three main results of the project,

- the novel concept of cyber-physical intrusion detection, one of the original ideas behind the project, was brought from a mere idea to an experimental demonstration in DTU’s SYSLAB environment (TRL 3-4). The results show that the technique can detect instances of cyber intrusion. However, in many cases it will not be useful as a standalone indicator, but rather as one factor in combination with e.g. vulnerability analysis, impact analysis and traditional (ICT network) intrusion detection.
- the existing vulnerability analysis framework CySeMoL was extended, not significantly increasing its TRL (4) but broadening its application area to include smart distribution grids with a high DER penetration. The framework in its present state can be directly used in the context of system design or system auditing.
- the novel concept of integrated cybersecurity assessment was advanced from an idea towards a limited proof of concept (TRL 3). However, significant development effort would be required to lift the concept towards the level of a semi-automated demonstrator.

1.7 Project conclusion and perspective

The results from WP2 demonstrate the feasibility of a statistical approach to integrating cyber-physical observations in demand response oriented intrusion detection systems. The validity and accuracy of the developed statistical models has to be evaluated in future studies. Approaches to avoid the manual classification of response types should also be replaced by more principled metrics based from attack goals.

The reported monitoring for normal and anomalous demand response behaviour offers a number of possible applications beyond cyber-physical intrusion detection, such as: system supervision and decision support, monitoring of an aggregator’s portfolio to estimate flexibility or monitor user behaviour, or validation of a contracted response.

The results from WP3 provide an overview of the vulnerabilities of various asset combinations and how these combinations enable different forms of attack paths. A number of reference models for particular system environments were identified and prioritized in order to capture the most central and important pieces of infrastructure that a typical power utility owns and operates, from a cyber security centric perspective. The project described reference models for SCADA infrastructure, substation automation infrastructure, smart metering infrastructure, immediate control of distributed energy resources and enterprise IT infrastructure. Additionally, a number of common operating systems and substation automation components have been described.

Further work in this area would involve studying the effects of synergy between and among different protection strategies used in a complementary fashion. Currently, the additional complexity and time demands of the evaluation process would be vast, since all possible combinations of the protection strategy scenarios would need to be modeled and evaluated for each reference model.

WP5 has shown the overall feasibility of integrating cybersecurity tools from multiple domains into an overall risk assessment tool which takes the complex interactions between domains in smart grid systems into account. The central approach is based on a ”hypothesis template” defining the domain interactions and the sequence in which these must be processed. The process of developing a hypothesis template for a real-world scenario pointed to one of the major challenges of the chosen approach: Successful attacks on an IT infrastructure will typically offer a multitude of manipulation options to the attacker. The resulting hypothesis templates will therefore have multiple – often many - branching points, leading to a potential ”explosion of hypotheses” which would require a lot of computing resources to analyze. Future work in this area would concentrate on identifying more efficient methods for the resolution of branches, as well as on the partial automation of hypothesis template generation which at the moment has to be performed manually by a group of domain experts.

The project indicates that an automated, integrated assessment has the potential to yield more accurate results than performing several separate assessments whose results need to be interpreted and finally merged and synthesized by a human. The strength of the integrated assessment resides in (1) the possibility to handle dependencies that cross boundaries of different domains in an automated fashion and already during the assessment process; and hence (2) the ability of the framework to operate in real time (online assessment), and therefore the ability to provide frequent re-assessments based on an actual state of the power grid and its IT/OT infrastructure.

In a longer perspective, this project is part of the ongoing effort to develop the next generation of smart energy systems with a higher degree of embedded automation which are crucially required to enable the green transformation of the energy system.

Annex: Bibliography

[1] L.H. Hansen, O. Sundström, S. Harbo, R. Villefrance, iPower WP 4.8: FLECH - Technical Requirement Specification, 2013.

[2] N. Nordentoft, iPower WP 3.8: Development of a DSO-Market on Flexibility Services, 2013.

[3] The Advanced Security Acceleration Project (ASAP-SG), Security Profile for Advanced Metering Infrastructure, 2012. http://osgug.ucaiug.org/utilisec/amisec/Shared %20Documents/AMI%20Security% 20Profile%20%28ASAP-SG%29/AMI%20Security %20Profile%20-%20v2 1.pdf.

[4] Smart Grid Interoperability Panel - Smart Grid Cybersecurity Comittee, Guidelines for Smart Grid Cybersecurity, vol. 1-3, 2014. http://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf.

[5] T. Sommestad, G. Bjorkman, VIKING Report D2.3: SCADA system architectures, 2010.

[6] Object Management Group, Meta Object Facility (MOF), 2014. http://www.omg.org/spec/ MOF/2.4.2/.

[7] P. Johnson, J. Ullberg, M. Buschle, U. Franke, K. Shahzad, P2AMF: Predictive, Probabilistic Architecture Modeling Framework, in: M. van Sinderen, P. Oude Luttighuis, E. Folmer, S.

Bosems (Eds.), Enterp. Interoperability, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013: pp. 104–117. doi:10.1007/978-3-642-36796-0_10.
[8] P. Johnson, J. Ullberg, M. Buschle, U. Franke, K. Shahzad, An architecture modeling framework for probabilistic prediction, Inf. Syst. E-Bus. Manag. 12 (2014) 595–622. doi:10.1007/s10257-014-0241-8.
[9] T. Sommestad, M. Ekstedt, H. Holm, The Cyber Security Modeling Language: A Tool for Assessing the Vulnerability of Enterprise System Architectures, IEEE Syst. J. 7 (2013) 363–373. doi:10.1109/JSYST.2012.2221853.
[10] H. Holm, K. Shahzad, M. Buschle, M. Ekstedt, P2CySeMoL: Predictive, Probabilistic Cyber Security Modeling Language, IEEE Trans. Dependable Secure Comput. 12 (2015) 626–639. doi:10.1109/TDSC.2014.2382574.
[11] Object Management Group, Unified Modeling Language (UML), 2014. http://www.omg.org/spec/UML/2.4.1/.
[12] Object Management Group, Object Constraint Language (OCL), 2014. http://www.omg.org/spec/OCL/2.4/.
[13] A.J.A. Wang, Information security models and metrics, in: ACM Press, 2005: p. 178. doi:10.1145/1167253.1167295.
[14] CCRA, Common Criteria for Information Technology Security Evaluation, 2012. http://www.commoncriteriaportal.org.
[15] C. Alberts, A. Dorofee, J. Stevens, C. Woody, Introduction to the OCTAVE approach, Carnegie Mellon - Software Engineering Institute, Pittsburgh, PA, 2003.
[16] F. den Braber, I. Hogganvik, M.S. Lund, K. Stølen, F. Vraalsen, Model-based security analysis in seven steps — a guided tour to the CORAS method, BT Technol. J. 25 (2007) 101–117. doi:10.1007/s10550-007-0013-9.
[17] R. Breu, F. Innerhofer-Oberperfler, A. Yautsiukhin, Quantitative Assessment of Enterprise Security System, in: IEEE, 2008: pp. 921–928. doi:10.1109/ARES.2008.164.
[18] H. Huang, S. Zhang, X. Ou, A. Prakash, K. Sakallah, Distilling critical attack graph surface iteratively through minimum-cost SAT solving, in: ACM Press, 2011: p. 31. doi:10.1145/2076732.2076738.
[19] F. Chen, J. Su, Y. Zhang, A Scalable Approach to Full Attack Graphs Generation, in: F. Massacci, S.T. Redwine, N. Zannone (Eds.), Eng. Secure Softw. Syst., Springer Berlin Heidelberg, Berlin, Heidelberg, 2009: pp. 150–163. doi:10.1007/978-3-642-00199-4_13.
[20] K. Ingols, M. Chu, R. Lippmann, S. Webster, S. Boyer, Modeling Modern Network Attacks and Countermeasures Using Attack Graphs, in: IEEE, 2009: pp. 117–126. doi:10.1109/ACSAC.2009.21.
[21] S. Jajodia, S. Noel, B. O'Berry, Topological Analysis of Network Attack Vulnerability, in: V. Kumar, J. Srivastava, A. Lazarevic (Eds.), Manag. Cyber Threats, Springer-Verlag, New York, 2005: pp. 247–266. doi:10.1007/0-387-24230-9_9.
[22] M. Buschle, P. Johnson, K. Shahzad, The enterprise architecture analysis tool - Support for the predictive, probabilistic architecture modeling framework, in: 2013: pp. 3350–3364.
[23] KTH, ICS, The Enterprise Architecture Analysis Tool (EAAT), 2014. https://www.ics.kth.se/eaat/.
[24] S. Borlase, ed., Smart grids: infrastructure, technology, and solutions, Taylor & Francis, Boca Raton, FL, 2012.
[25] J. Ekanayake, K. Liyanage, J. Wu, A. Yokoyama, N. Jenkins, Smart Grid: Technology and Applications, John Wiley & Sons, Ltd, Chichester, UK, 2012. doi:10.1002/9781119968696.
[26] M.S. Thomas, J.D. McDonald, Power system SCADA and smart grids, CRC Press, Taylor & Francis Group, Boca Raton, Florida, 2015.
[27] International Electrotechnical Commission, IEC 61850-8-1:2011 - Communication networks and systems for power utility automation - Part 8-1: Specific communication service mapping (SCSM). Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3., 2011. https://webstore.iec.ch/publication/6021.
[28] International Electrotechnical Commission, IEC 61850-9-2:2011 - Communication networks and systems for power utility automation - Part 8-1: Specific communication service mapping (SCSM). Sampled values over ISO/IEC 8802-3., 2011. https://webstore.iec.ch/publication/6023.
[29] International Electrotechnical Commission, IEC 60870-5-101:2003 - Telecontrol equipment and systems - Part 5-101: Transmission protocols - Companion standard for basic telecontrol tasks, 2003. https://webstore.iec.ch/publication/3743.

[30] International Electrotechnical Commission, IEC 60870-5-104:2003 - Telecontrol equipment and systems - Part 5-104: Transmission protocols - Network access for IEC 60870-5-101 using standard transport profiles, 2006. https://webstore.iec.ch/publication/3746.
[31] National Energy Technology Laboratory, NETL Modern Grid Strategy -- Advanced metering infrastructure, 2008. https://www.smartgrid.gov/files/advanced_metering_infrastructure_02-2008.pdf.
[32] J. Searle, G. Rasche, A. Wright, S. Dinnage, NESCOR Guide to Penetration Testing for Electric Utilities, EPRI, 2013. http://smartgrid.epri.com/doc/NESCORGuidetoPenetrationTestingforElectricUtilities-v3-Final.pdf.
[33] R. Blom, M. Korman, R. Lagerstrom, M. Ekstedt, Analyzing attack resilience of an advanced meter infrastructure reference model, in: IEEE, 2016: pp. 1–6. doi:10.1109/CPSRSG.2016.7684095.
[34] DNV GL, A Report on Distributed Energy Resources, 2014. http://www.nyiso.com/public/webdocs/media_room/publications_presentations/Other_Reports/Other_Reports/A_Review_of_Distributed_Energy_Resources_September_2014.pdf.
[35] H. Holm, A Manual for the Cyber Security Modeling Language, 2014. http://www.kth.se/polopoly_fs/1.588086!/cysemol_manual_v2.2_changelog.pdf.
[36] M. Korman, M. Välja, G. Björkman, M. Ekstedt, A. Vernotte, R. Lagerström, Analyzing the Effectiveness of Attack Countermeasures in a SCADA System, in: ACM Press, 2017: pp. 73–78. doi:10.1145/3055386.3055393.
[37] R. Langner, Stuxnet: Dissecting a Cyberwarfare Weapon, IEEE Secur. Priv. Mag. 9 (2011) 49–51. doi:10.1109/MSP.2011.67.
[38] PowerLab DK, SYSLAB, (n.d.). www.powerlab.dk/facilities/syslab.aspx.
[39] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (2009) 1–58. doi:10.1145/1541880.1541882.
[40] D. Yang, A. Usynin, J. Hines, Anomaly-based intrusion detection for SCADA systems, in: American Nuclear Society, 2006.
[41] A. Zaher, S.D.J. McArthur, D.G. Infield, Y. Patel, Online wind turbine fault detection through automated SCADA data analysis, Wind Energy. 12 (2009) 574–593. doi:10.1002/we.319.
[42] M.A. Sanz-Bobi, A.M.S. Roque, A. de Marcos, M. Bada, Intelligent system for a remote diagnosis of a photovoltaic solar power plant, J. Phys. Conf. Ser. 364 (2012) 012119. doi:10.1088/1742-6596/364/1/012119.
[43] Y. Isozaki, S. Yoshizawa, Y. Fujimoto, H. Ishii, I. Ono, T. Onoda, Y. Hayashi, On detection of cyber attacks against voltage control in distribution power grids, in: IEEE, 2014: pp. 842–847. doi:10.1109/SmartGridComm.2014.7007753.
[44] X. Song, M. Wu, C. Jermaine, S. Ranka, Conditional Anomaly Detection, IEEE Trans. Knowl. Data Eng. 19 (2007) 631–645. doi:10.1109/TKDE.2007.1009.
[45] Z.B. Zhao, W.S. Xu, D.Z. Cheng, User behavior detection framework based on NBP for energy efficiency, Autom. Constr. 26 (2012) 69–76. doi:10.1016/j.autcon.2012.04.001.
[46] V. Catterson, S. McArthur, G. Moss, Online conditional anomaly detection in multivariate data for transformer monitoring, in: IEEE, 2011: pp. 1–1. doi:10.1109/PES.2011.6038911.
[47] Pecan Street Dataport, (2016). https://dataport.pecanstreet.org.
[48] M. Sengupta, Y. Xie, A. Lopez, A. Habte, G. Maclaurin, J. Shelby, The National Solar Radiation Data Base (NSRDB), Renew. Sustain. Energy Rev. 89 (2018) 51–60. doi:10.1016/j.rser.2018.03.003.
[49] W.N. Venables, B.D. Ripley, Modern applied statistics with S, 4th ed, Springer, New York, 2002.
[50] J.D. Head, M.C. Zerner, A Broyden—Fletcher—Goldfarb—Shanno optimization procedure for molecular geometries, Chem. Phys. Lett. 122 (1985) 264–270. doi:10.1016/0009-2614(85)80574-1.
[51] S. Lehnhoff, O. Nannen, S. Rohjans, F. Schlogl, S. Dalhues, L. Robitzky, U. Hager, C. Rehtanz, Exchangeability of power flow simulators in smart grid co-simulations with mosaik, in: IEEE, 2015: pp. 1–6. doi:10.1109/MSCPES.2015.7115410.
[52] N. Garcia-Pedrajas, C. Hervas-Martinez, D. Ortiz-Boyer, Cooperative Coevolution of Artificial Neural Network Ensembles for Pattern Classification, IEEE Trans. Evol. Comput. 9 (2005) 271–302. doi:10.1109/TEVC.2005.844158.

[53] J. Mendes-Moreira, C. Soares, A.M. Jorge, J.F.D. Sousa, Ensemble approaches for regression: A survey, ACM Comput. Surv. 45 (2012) 1–40. doi:10.1145/2379776.2379786.
[54] N.P. Lund, R.D. Grandal, S.H. Sørensen, M.F. Bendtsen, G.L. Ray, E.M. Larsen, J. Mastop, F. Judex, F. Leingruber, K.J. Kok, P.A. MacDougall, EcoGrid EU - A Prototype for European Smart Grids. Overall evaluation and conclusion, 2015.
[55] E.M. Larsen, P. Pinson, G. Le Ray, G. Giannopoulos, Demonstration of market-based real-time electricity pricing on a congested feeder, in: IEEE, 2015: pp. 1–5. doi:10.1109/EEM.2015.7216777.
[56] E.M. Larsen, Demand response in a market environment, PhD dissertation, Technical University of Denmark, 2016.
[57] S. Marsland, Machine learning: an algorithmic perspective, CRC Press, Boca Raton, 2009.
[58] E.M. Larsen, P. Pinson, F. Leimgruber, F. Judex, Demand response evaluation and forecasting — Methods and results from the EcoGrid EU experiment, Sustain. Energy Grids Netw. 10 (2017) 75–83. doi:10.1016/j.segan.2017.03.001.
[59] B. Kordy, L. Piètre-Cambacédès, P. Schweitzer, DAG-based attack and defense modeling: Don’t miss the forest for the attack trees, Comput. Sci. Rev. 13–14 (2014) 1–38. doi:10.1016/j.cosrev.2014.07.001.

Glossary

AMI Advanced metering infrastructure (alternatively SMI - smart metering infrastructure).
AMI NMS AMI network management system.
AMI MMS AMI meter management system.
AMI FS AMI forecasting system.
CIS Customer information system.
DAN Distribution automation node.
DE Data engineering.
DER Distributed energy resource.
DERMS DER management system.
DMS Distribution management system.
DRAACS Demand-response analysis and control system.
DSM Demand side management (related to demand-response management).
DSO Distribution system operator (distribution grid operator).
EMS Energy management system.
GIS Geographical information system.
GMS Generation management system.
HAN Home area network.
HMI Human-machine interface.
IED Intelligent electronic device (further connected to sensors, actuators or other IEDs...).
LAN Local area network.
MDMS Meter data management system.
NAN Neighborhood area network.
OMS Outage management system.
PCT Programmable communicating thermostat.
PDC Phasor data concentrator (part of WAMPAC).
RTU Remote terminal unit.
SCS Substation control system.
TSO Transmission system operator (transmission grid operator).
WAMPAC Wide area monitoring, protection and control.
WAN Wide area network.
WMS Workforce management system.