Cyber-attack detection in smart grids

Jelle Maas
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
j.maas@student.utwente.nl

1. ABSTRACT

Nowadays smart (energy) grids are used more than ever. This makes the smart grids more tempting for hackers and cyber-attacks, because the grids are a critical infrastructure and hackers could profit from this. The reason cyber-attacks are also aimed at smart grids is because of the vulnerability in communication technologies. In this paper, new detection methods against cyber-attacks are investigated. The new detection method is implemented in a simulation that is based on reality and uses data about the potential energy in an electric field and the electric current. One of the detection methods determined with a 94% accuracy if there was a cyber-attack and the other detection method prevented 91% cyber-attacks from happening.

Keywords

smart grid, energy management, cyber-attack, detection, power grid, electric current, voltage, power, machine learning, simulation

2. INTRODUCTION

The smart grid nowadays is still the conventional grid that is used for electricity transmission, but information communications technology is added to the grid, which makes it a smart grid.

The detection of cyber-attacks in smart grids happens more often as the market for smart grids is growing according to [1], but before cyber-attack detection is possible the definition of a smart grid needs to be clear. A smart grid has multiple definitions as can be seen in the overview of international power system conference [11]. The definitions of IEEE and IET in this overview are the most clear and goal-based. According to the IEEE [13] "The smart grid has come to describe a next-generation electrical power system that is typified by the increased use of Communication and Information Technology in the generation, delivery and consumption of electrical energy." Where according to the IET [7] "The smart grid is fully functional around 2030 that will efficiently integrate the actions of all users to it generators, consumers and those that do both - in order to ensure an economically efficient, sustainable power system with low losses and high levels of quality and security of supply and safety." IET expects the smart grid is fully functional around 2030, but in the meantime it will already be used in some places in the world.

32nd Twente Student Conference on IT Jan. 31st, 2020, Enschede, The Netherlands.

Since the amount of smart grids in the world is increasing. We must not only look at the positive elements that it entails for all households, the environment, and people, but also at the negative elements. Because the smart grids are a critical infrastructure and the economy depends on the stability of the smart grid. And one of the negative elements that could threaten the smart grid is a cyber-attack. Cyber-attacks is a vulnerability of the system that you want to take away by being able to detect it and prevent it. Because if we would use smart grids that are not secure yet it could be vulnerable to cyber-attacks according to X. Cehn et al [3].

The added ICT to the conventional grid could be targeted by cyber-attacks and the possible attacks on smart grids include:

- The communication network could be targeted by cyber-attacks and by attacking the communication network within the smart grid it could overload or destabilize the system.
- Bypass authentication, when a cyber-attacker can bypass authentication it can use and control the smart grid [12].
- A smart grid can be attacked by DDoS-attacks.

To ensure there are no problems in the smart grid the cyber-attacks need to be detected and prevented. In this paper new detection methods will be discussed. The methods will use the amount of power and the price for the power as input to determine if a cyber-attack is occurring in the current state. When a cyber-attack is found the system prevents the system from overvoltage and undervoltage.

3. RELATED WORK

For some of these cyber-attacks as discussed in section 2 solutions are presented in literature:

- CUSUM algorithm and Kalman filter for state estimation. The system is a discrete-time linear dynamic system. Where the Kalman filter is used for the smallest mean squared error and the CUSUM algorithm for the quickest detection for cyber-attacks. the detection is for both random and structured cyber-attacks [8].
- Adding additional authentication in the system. The method is based on authentication of sampled value

message of the IEC 61850 Protocol sent between microprocessor devices of relay protection and automation. This is suggested to minimize the sequential power outages caused by cyber-attacks [12].

- Unsupervised or supervised machine learning for recognizing data injections in a smart grid [2].

4. RESEARCH QUESTIONS

The research question this paper investigates is: "How can we detect cyber-attacks by looking at the physical state of the network?". To answer the main research question two subquestions need to be answered:

- What ways are there to detect cyber-attacks?
- How efficient is the detection system for cyber-attacks?

5. METHOD

To answer the above mentioned research questions, the DEMkit simulator that has been developed by the CAES Chair of the EEMCS department will be used to implement and simulate cyber-attacks [4, 5, 6]. In the DEMkit simulation a house is simulated. The simulation of a house includes an electric vehicle (EV), battery, dishwasher, thermostat, washing machine and solar panels. The electric vehicle, dishwasher, thermostat and washing machine all need electricity. The solar panels generates electricity and the battery can store and release electricity. The prices for electricity is organised in the DEMkit as well. The prices are determined by an auction system. Each house communicates a bid that specifies how much power the house will consume/produce for a given price during a time interval.

Then the auctioneer determines a price and this results in a different power level. The whole auction system uses time intervals of 15 minutes. An auction is performed each time interval.

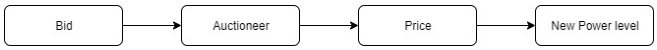

Figure 1. Visualisation of auction system

The auction class in the DEMkit simulation will be used for cyber-attack detection, because if a cyber-attack would take place and the cyber-attack changes the prices it could have a big impact on the whole smart grid, or on individual houses. The cyber-attacks will influence the prices and therefore the power in the system. So the two variables that are used to detect cyber-attacks are the electricity price and the amount of power. The electricity price is a variable integer between -1000 and 1000. Where 1000 is the highest price for buying electricity and -1000 the highest price for selling electricity. The amount of power is also an integer and is measured in Watt.

5.1 Implementation

As is shown in figure 2 the price has influence on the power consumption of houses through the auction system, if the price increases (decreases), the power demand decreases (increases). Subsequently, the increased demand for power results in a voltage decrease (increase). And something that should be prevented from happening is that the voltage decreases lower than 207 Volts and higher than 253 Volts. These values are the boundaries to which the voltage is allowed to deviate by regulations in the Netherlands. So if the price changes dramatically the power and voltage will also do so, this can lead to an overvoltage or undervoltage in the system. So for the implementation in the DEMkit simulation uses the electricity price and the amount of power.

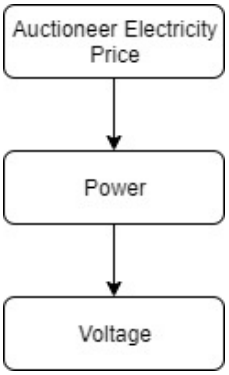

Figure 2. Influence of price in the smart grid

There are a lot of cases that the implementation takes care of, but in essence there are only 3 cases that should be taken care of:

1. The difference in current price and past prices.
2. The correlation between price and power.
3. If the points above points are all checked.

In the first case the difference between the current price and the past prices should not be to high, because a big fluctuation between prices could lead to peaks in power and voltage.

As mentioned in the beginning of this subsection, if the price increases the power decreases and voltage increases. If the price increases more and the power was already quite low it would mean that the voltage increases more as well. This could lead to an overvoltage, also the other way around would be an possibility and this could lead to an undervoltage.

And the last case is if everything is under control the system will use the price that has been determined by the system, because nothing is wrong.

All these cases do not necessarily have to be from cyber-attacks, but can also be caused by the smart grid itself which could cause problems, so the implementation can also help in these cases and not only during cyber-attacks.

As shown in Algorithm 1 on the next page there are two arrays that are filled when the final price and final amount is power is known. The arrays are used later in the else-if statements and also to get older prices in the case of a cyber-attack. The difference and difference2 are used in the else-if statement this is used for case 1. The first if-statement and the first else-if-statement is too prevent case 2 from happening. And case 3 is the last else case, this case is only used if all the if-else-statements beforehand are not true and the original price will be used, because then there is no cyber-attack detected and the system is safe. The efficiency of the implementation is determined by the amount of times it correctly detects a cyber-attack, So only the true positives. This means that there should be a positive effect visible on the amount of power that is used.

```
Require: price, power
    pastPrices = [ ]
    pastPower = [ ]

    difference = abs(price - pastPrices[-1])
    difference2 = abs(pastPrices[-1] - pastPrices[-2])

    if  power >= 0 and price < 0 then

      i = -2

      while pastPrices[i] < 0 do
        i -= 1
      end while

      price = pastPrice[i]
      pastPrices.append(price)
      pastPower.append(demandForPrice(price))

    else if power <= 0 and price > 0 then

      i = -2

      while pastPrices[i] > 0 do
        i -= 1
      end while

      price = pastPrice[i]
      pastPrices.append(price)
      pastPower.append(demandForPrice(price))

    else if (-250 < power < 250) and ( -250 <
    pastPower[-1] < 250) and ( -250 < pastPower[-2]
    and difference < 500 and difference2 < 500 then

      price = price
      pastPrices.append(price)
      pastVoltages.append(demandForPrice(price))

    else if (-250 < power < 250) and ( -250 <
    pastPower[-1] < 250) and ( -250 < pastPower[-2] <
    250) and ( difference > 500 or difference2 > 500)
    then

      price = pastPrices[-1]
      pastPrices.append(price)
      pastVoltages.append(demandForPrice(price))

    else
      price = price
      pastPrices.append(price)
      pastVoltages.append(demandForPrice(price))

    end if

    return  price
```

Algorithm 1: Cyber-attack detection and prevention method

5.2 Simulation of cyber-attacks

The implementation mentioned in section 5.1 is tested by implementing some test cases where cyber-attacks are simulated. In total 4 tests are written. The were four categories for the test cases are:

1. An attack where prices and powers are completely random.
2. An attack where the prices steadily increases/decreases and stays at the maximum/minimum price
3. An attack where the prices steadily increases (or decreases) and if it reaches the maximum (or minimum) price the price drops (or lifts) to the minimum (or maximum) price
4. An attack which at certain timestamps drops/lift the prices.

5.2.1 Implementation of cyber-attacks

For the first case a hacker can simply send random data and does not have to know anything about the system, but if the system is not prepared for an attack it could do a lot of damage. In algorithm 2 can be seen that it was also an easy implementation to simulate.

```
Require: Min, Max, Length
    iterator = 0
    RandomArray = [ ]
    while iterator < Length do

      RandomArray.append(random.randrange(Min,Max))

      iterator += 1

    end while

    return  RandomArray
```

Algorithm 2: Pseudocode for random vector

The second case was harder to detect for the system, but the system should look at the correlation of price and power, because it steadily increases and not with big differences as can be seen in algorithm 3. The decreasing vector is almost the same, but just the different way around.

In the third case the differences between the current price and the last price are used, because it could see a big change in prices. The code for the steadily increasing price and the sudden drop is shown in algorithm 4, the sudden lift after decreasing is almost identical, but a few things are different.

The fourth case does not have a specific code for vectors, but if a cyber-attacker sees that a household it needs energy to finish a task from an electric device it could put the price at a maximum and when it wants to discharge it puts it at the minimum price, so there are big spikes in the power supply, which could cause damage.

5.3 Machine Learning

Besides the implementation in DEMkit to try to detect and prevent cyber-attacks from happening machine learning can also be used to detect cyber-attacks. In DEMkit a library [10] is used to put all prices and power-levels and time-intervals into Excel. Then the CSV (Comma-Seperated values) file is the input for a machine learning programm called WEKA [9]. And then from the input a classifier can be build via cross-validation.

```
Require: Min, Max, Length

    iterator = 0
    RandomArray = [ ]
    while iterator < Length do

      if iterator == 0 then

        firsPart = math.floor(Min)
        secondPart = math.floor((Min+ (0.05 *
        (Max-Min))))
        randomInt = random.randrange(firstPart,
        secondPart)
        RandomArray.append(randomInt)

      else

        firsPart = math.floor(RandomArray[-1])
        secondPart = math.floor((RandomArray[-1] +
        (0.05 * (Max-Min))))
        randomInt = random.randrange(firstPart,
        secondPart)
        RandomArray.append(randomInt)
      end if

      if randomInt < Max then

        RandomArray.append(randomInt)
        iterator += 1
      end if

    end while

    return  RandomArray
```

Algorithm 3: Pseudocode for increasing vector

```
Require: Min, Max, Length

    iterator = 0
    RandomArray = [ ]
    levelIterator = 0
    while iterator < Length do

      if iterator == 0 then

        firsPart = math.floor(Min)
        secondPart = math.floor((Min+ (0.05 *
        (Max-Min))))
        randomInt = random.randrange(firstPart,
        secondPart)
        RandomArray.append(randomInt)

      else

        firsPart = math.floor(RandomArray[-1])
        secondPart = math.floor((RandomArray[-1] +
        (0.05 * (Max-Min))))
        randomInt = random.randrange(firstPart,
        secondPart)
        RandomArray.append(randomInt)
      end if

      if randomInt > Max - (Math.abs(Max-Min)*0.005
      then

        levelIterator += 1
      end if

      if randomInt < Max and levelIterator < 4 then

        RandomArray.append(randomInt)
        iterator += 1

      else

        random.Array.append(Min)
        levelIterator = 0
      end if

    end while

    return  RandomArray
```

Algorithm 4: Pseudocode for increasing vector then sudden drop

6. RESULTS

6.1 Implementation

After the implementation the simulations of DEMkit were used to see the difference between cyber-attacks and the implementation. All three points discussed in subsection 5.1 work and prevent big spikes in power consumption.

The difference between the implementation and the cyber attack can be found in figure 3 and figure 4. The price and power are measured in a timespan of 24 hours with a time interval of 15 minutes.

Figure 3. Test without implementation

Figure 4. Test with implementation

As can be seen in figure 3 the minimum and maximum are far apart and the difference is around 15000 in comparison with the test with implementation it has a peak of around 5000. Even though this is the most extreme case all other tests had less difference between maximums and minimums. Besides this result more tests got the same result. In all cases the cyber-attacks were detected and tried to be prevented. In a 91% of the cases the peaks and valleys decreased just like the figures above. As can be seen in table 1 the implemented algorithm did not always prevent the cyber-attacks, but this only happened for case 3 and 4 as discussed in subsection 5.2. The reason it did not always prevent the cyber-attacks in case 3 is because it might not be done with processing the last cyber-attack. And for case 4 it did not get worse, but at least it filtered out the big difference in power, but it did not prevent a big peak, so it is not registered as a prevention.

Attack type	#Cyber-attacks	#Prevented cyber-attacks
Case 1	1	1
Case 2	2	2
Case 3	22	20
Case 4	7	6

Table 1. Detected and prevented cyber-attacks per attack type

6.2 Machine Learning

As mentioned in section 5.3 it is possible to generate a classifier. The classifier that got the highest percentage of cyber-attack detection was the classifier REPTree. REPTree is used because after a series of tests and comparing every classifier this classifier gave the highest accuracy. As can be seen in table 2 it got an accuracy of around 94% and the difference between the amount of cross-validation folds did not change more than 0.5%. The other classifiers all got different results from around 50% to 93%.

a	b	<–classified as
5920	124	a = Cyber Attack
481	4211	b = Not a Cyber Attack

Table 2. Confusion matrix for cyber attack detection

7. CONCLUSION

After using the implementation and the four test cases there is one general conclusion. In all cases in general it filters out the maximums and minimums, which could lead to problems in the smart grid. And as shown in subsection 6.1 it can detect cyber-attacks and also in all the other test cases the cyber-attacks were detected and prevented from happening, by the implementation, which lead to a 91% accuracy detection and prevention implementation.

The Machine learning classifier is also efficient in detecting cyber-attacks it correctly detects if it is a cyber-attack or not in 94% of the cases.

So both the implementation and the machine learning classifier are effective in detecting a cyber-attack when we look at the physical state of a network.

8. DISCUSSION

The system that is used in this paper is novel because it, as mentioned in section 5, uses the power and the price to determine if a cyber-attack is happening in the smart grid. This is useful in the future because more and more smart grids are likely to be created and cyber-attackers could try to attack these systems and thus it is smart to do more research into possible methods to protect these grids.

8.1 Future Work

The classifier that is build via WEKA can give a java class as output, but the whole simulation is written in python, so the java class has to be changed to python and if this is done 94% of the cyber-attacks that enter the system can be detected. To improve this, more data can be used to build a better classifier. Besides the classifier the implementation could be tested on a whole street, because the implementation that is build now is made for one household, but could be used for a whole street, but this can be tested in the future as well.

9. REFERENCES

[1] J. Aho, M. Amin, A. Annaswamy, G. Arnold, A. Buckspan, A. Cadena, D. Callaway, E. Camacho, M. Caramanis, A. Chakrabortty, A. Chakraborty, J. Chow, M. Dahleh, C. Demarco, A. Dominguez-Garcia, D. Dotta, A. Farid, P. Flikkema, D. Gayme, and J. Stoustrup. *IEEE Vision for Smart Grid Controls: 2030 and Beyond*. 01 2013.

[2] F. Akbarian, A. Ramezani, M. Hamidi-Beheshti, and V. Haghighat. Intrusion detection on critical smart grid infrastructure. In *2018 Smart Grid Conference (SGC)*, pages 1–6, Nov 2018.

[3] X. Chen, L. Zhang, Y. Liu, and C. Tang. Ensemble learning methods for power system cyber-attack detection. In *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 613–616, April 2018.

[4] DEMkit, the University of Twente. Dec. 2019. https://www.utwente.nl/en/eemcs/energy/demkit/.

[5] G. Hoogsteen. A cyber-physical systems perspective on decentralized energy management. Dec. 2017.

[6] G. Hoogsteen, J. L. Hurink, and G. J. M. Smit. Demkit: a decentralized energy management simulation and demonstration toolkit. In *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pages 1–5, Sep. 2019.

[7] IET. What is a smart grid? *What is a smart grid?*, August. 2019.

[8] M. N. Kurt, Y. Yılmaz, and X. Wang. Distributed quickest detection of cyber-attacks in smart grid. *IEEE Transactions on Information Forensics and Security*, 13(8):2015–2030, Aug 2018.

[9] Machine learning group of Waikato university. January 2020. https://www.cs.waikato.ac.nz/ml/weka/.

[10] J. McNamara. January 2020. https://xlsxwriter.readthedocs.io/index.html.

[11] M. Shabanzadeh and M. P. Moghaddam. What is the smart grid? definitions, perspectives, and ultimate goals. In *28th International Power System Conference (PSC)*, 2013.

[12] T. Sharafeev, O. Ju, and A. Kulikov. Cyber-security problems in smart grid cyber attacks detecting methods and modelling attack scenarios on electric power systems. 2018.

[13] X. Yu. Introduction to ies panel discussion on smart grids. In *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pages 11–14, Nov 2010.