UCLouvain

École polytechnique de Louvain

Rule-based expert system for energy optimization

Detection and identification of relationships between rules in knowledge base

Author: **Magali Legast**
Supervisor: **Axel Legay**
Readers: **Laurent Francis, Thibaut Piraux**
Academic year 2020–2021
Master [120] in Computer Science and Engineering

Abstract

This master thesis presents the design of a rule-based expert system that uses specific energy observations to provide energy recommendations to its users. The aim of this thesis was to improve a similar recommendation AI developed by WeSmart, a company that provides support to energy communities. Since the quality of the recommendations issued are highly dependent on the rules in the knowledge base, the main focus of the work was to develop a method to help the human experts in the design of such rule sets.

To reach this objective, I have produced an algorithm that identifies different types of relationships between rules, basing my work on an existing technique created for firewall policies. I have also implemented a prototype of tool using that algorithm. This program, Relationship Identification Tool (RIT), allows to manage a rule set by identifying the different relationships between the rules as well as providing support for the creation and modification of the rule set.

This manuscript starts with an explanation of the context of energy communities and a presentation of WeSmart. Then comes an assessment of the existing AI developed by WeSmart and suggestions for improvements. After that, a review of literature from different fields is presented alongside the conclusions made regarding the development of the chosen solution. Those fields are associative classification, expert systems and firewalls. After this description of the process that lead to it comes the presentation of the results.

Acknowledgement

First, I would like to thank my adviser, Axel Legay, for giving me the opportunity to work on this interesting subject, for his guidance through the year and for his support and advice, both for this thesis and my future projects.

Thank very much also to Thibaut Piraux, my contact at WeSmart, who followed my work, gave me interesting feedback and helped take my thinking further.

Thank you to WeSmart and François Brodes for allowing me to do my work in collaboration with the company and welcoming me in your office.

Thank you to my third reader, Laurent Francis, who enthusiastically accepted to read my work.

Thank you so much to my parents, who have always been there to support and encourage me no matter what, this years and all the previous ones. And thank you to my grand-parents, Bonne Mamy, Timamy, Bon Papa and Titane, for their encouragements and their moral and material support.

Thank you to my friends and the rest of my family who have been there for me. I'm especially grateful to Limor who has been by my side from the choice of the subject to the very last day. Special thanks to Tom too, for his friendship throughout the year.

Thank you very much to my godfather who has proofread this manuscript and helped me improve it.

Thank you to the professors, assistants and tutors who thought me at UCLouvain, but also to all the teachers that came before them and who helped me love science and believed in me, because this thesis is not only the accomplishment of this year's work, but also the outcome of many years of learning and teaching before that.

Contents

Introduction

At the highest level, the context of this thesis is the current climate crisis. One of the key factor in the degradation of our environment is the consumption of energy. Ways to reduce its impact on the environment include reduction of the consumption and renewable energy production. As many changes, those goals come with several challenges. One of them is the difficulty of storing that type of energy, with which comes the need to distribute it efficiently when it is produced. Another challenge is to combine the maximum reduction of consumption with the minimal impact possible on human activity and comfort.

In response to those challenges, the idea of energy communities has emerged and is becoming a reality in Belgium and across Europe. The goal of such communities is to bring together different actors so they can produce and consume renewable energy together in a more efficient way. Indeed, the excess renewable energy produced locally by one actor can directly be consumed by another one within the community. Only when no member of the community needs that energy is it sent to the general grid. It also limits the amount of energy the community has to buy from power supplier. This allows for a better fit between the global consumption and global production of the community, which increases its energy autonomy. This in turns leads to a more efficient energy use and reduces the price of energy for all parties involved.

The company WeSmart assists such communities in their organization and management using a digital platform the start-up has developed. One of the services it offers is the monitoring and billing of the energy consumed and produced by each member of the community. This service is also enhanced with an artificial intelligence (AI) that raises alerts and provides users with personal recommendations based on their individual consumption. A first version of this AI has already been developed and takes the form of a rule-based expert system with manually crafted rules.

The aim of this master thesis is to provide suggestions on ways to improve the existing AI developed by WeSmart and implement a prototype for one of the propositions. Possible enhancements can for example include improvement of AI results, or support to facilitate the management of its knowledge base.

In order to achieve this goal, the first part of my work has been to analyze the needs of WeSmart (Chapter 1, 2). I could then formulate adequate and innovative enhancement propositions for the AI, as well as select one to implement (Chapter 2). In the second part, I have reviewed scientific literature in order to explore different potential implementation solutions and find the most appropriate one (Chapter 3). Finally, for the third part, I have adapted the selected theoretical solution to my application (Chapter 4) and have implemented it in a prototype (Chapter 5). Those different steps of my work are presented in that order in this report.

Chapter 1

Context

This master thesis consisted in developing a solution for WeSmart, a start-up that develops a digital platform to setup and manage energy communities. The company also helps the communities in their creation process and work with business that want to cut down on energy consumption. In order to understand the needs of WeSmart and the interest of the application it develops, it is import to understand the context in which this company is set and the challenges it takes up. This involves understanding energy communities, what kind of support WeSmart offers and what digital solutions has been developed for that. This chapter will go over these subjects in order to present an overview of the context.

1.1 Energy communities

1.1.1 General concept

The world is currently facing different ecological challenges that include global warming, excess of green house gas (GHG) emissions and decrease of fossil energy resources. There has been various reactions taken at different level in response to this situation. In this context, the European Union (EU) has fixed different objectives and developed several action plans. This includes the ratification of the Paris Agreement [13], the adoption of the European Green Deal [9], the update of legal energy policies through the Clean energy for all Europeans package (CEP) [10] in 2016 or the recast of Renewable Energy Directive (RED II) [15] in 2018. Among the different objectives, the EU wants to achieve carbon neutrality by 2050 and reduce its GHG emissions by at least 55% by 2030.

In order to reach these goals, actions in the energy sector are very important. Indeed, according to the European Environment Agency [19], the fuel combustion is responsible for over three-quarter of the GHG emission. Out of this, about two third is used for energy industries, for manufacturing and construction, and for households, commerce, institutions and others.

Taking this into account, the updated Renewable Energy Directive [15] now requires that the energetic needs of the EU would be fulfilled by at least 32% of renewable energy by 2030. The target for 2020 was to use at least 20% of renewable energy by 2020 and it has been reached. However, efforts must be continued in order to reach the future targets and many challenges need to be faced. Of course, the energy production from renewable source needs to be increased. This requires investments from authorities, but private investments would also be of great benefits. This means giving incentives to companies and citizen to invest in renewable energy. Another challenge is the growing tendency of citizens to oppose large renewable energy projects from traditional actors that are usually not willing to include the concerned local communities in the project nor to share the socio-economic benefits with them [17]. Renewable energy production is also often dependent on external factors like wind or sunshine, which makes it less steady than fossil resources. Since electricity is difficult to store, this means that most of it needs to be distributed and used when it is produced. The grid will thus need to have "flexibility" to accommodate for varying supply and demand of energy.

To promote the use of renewable energy and bring different actors together to consume it more efficiently, the EU is counting, among other things, on energy system integration [11]. This refers to "the planning and operating of the energy system “as a whole”, across multiple energy carriers, infrastructures, and consumption sectors, by creating stronger links between them with the objective of delivering low-carbon, reliable and resource-efficient energy services, at the least possible cost for society." To reach a more integrated system, different actors, including citizens, need to be allowed and encouraged to participate in the production, sharing, selling and self-consumption of renewable energy.

One way to work towards such a system is by the development of local energy communities (LECo). Such communities bring different actors together so they can produce, consume and sometimes store energy in a more efficient way. When an individual user can produce its own energy, for example using photovoltaic panels, any surplus that isn’t directly consumed is injected back to the national grid. Conversely, any extra energy needed has to be bought from the power supplier. In an energy community, participants can exchange energy between each other, before buying it or selling it to the power supplier if needed. This means that the excess energy produced locally by one member can directly be consumed by another participant of the community, usually at a lower price that if it had to be bought from the grid. That energy exchange is typically done through the general electricity network and using precise meters for each participants. So, in many cases creating or joining an energy community requires little to no infrastructure adaptation, making it easy and accessible.

One of the reason this is particularly interesting is because LECo can bring together members that have complementary production and consumption profiles. Indeed, some users produce more than they consume while others consume more than they produce, in general or at specific times. Schools often have large flat roofs that are ideal for photovoltaic panels, but they consume almost nothing during evenings and weekends, nor during the summer, when production is the most important. Most offices and stores also have the majority of their consumption happening within a specific schedule. On the other hand, households typically consume mostly in the evenings, weekends, and during the holidays if there are children home, but their production capacity vary greatly from one another. Places holding cultural events, restaurants or leisure resorts also have specific and distinct consumption profiles. By connecting the right actors to each others, there can be a good fit between the global consumption and global production of the community. That way, energy communities can maximize their energy efficiency and autonomy in a way that would not have been possible for each member individually.

Such an organization offers multiple advantages [12], [17], [18] and addresses the challenges presented above. First, it allows the energy consumption to be more efficient because it is produced and consumed locally. Thanks to that, the energy waste that happens during its transport through the grid because of the Joule effect can be avoided. As a mean to integrate more local actors and connect them to each other, LECo are also a way to increase the flexibility of the electricity system through demand-response and storage, making it more resilient to an intermittent renewable energy production. This has of course a positive environmental impact, but also an economic one. Indeed, it reduces the price of electricity for the LECo members since they can sell and buy energy in a more advantageous way within their community.

As a second interesting advantage, it encourages citizen to take an active role in the electricity system, which is one of the main goal of the EU and has several benefits. Since anybody can be an active participant, it should make it easier for the public to accept such renewable energy projects, as opposed to the ones they have no say in. It should also be an incentive for citizens to invest in renewable energies. One reason for this is that being part of a community makes the investment more financially interesting. The energy surplus produced can indeed be sold to other participants at a better price than to the power supplier. Another reason can simply be the feeling of involvement that may arise from their engagement in the community. This can also help them in modifying their consumption habits towards the reduction of their energetic impact on the environment.

On top of that, it allows people who rent their home or who don't have the financial means to invest in renewable energy on their own to integrate a community and get benefits they wouldn't have been able to access otherwise. LECo can thus help with social inclusion and the reduction of energy precariousness. Furthermore, the deployment and management of the communities is likely to create jobs and have a positive impact on the local economy as a whole.

1.1.2 Legal ground and current situation

European Union

In order to promote their development, the European Union has introduced a legal definition for such communities in the Clean Energy for all Europeans Package [10]. Two types of energy communities are specified. There are the "renewable energy communities" (REC), defined in the revised Renewable Energy Directive in article 2, and the "citizen energy communities" (CEC), defined in the Electricity Market Directive (EMD) in article 2. The latter doesn't require the energy produced and exchanged to come from a renewable source. The legal definitions are the following [24]:

'**renewable energy community**' means a legal entity:

- (a) which, in accordance with the applicable national law, is based on open and voluntary participation, is autonomous, and is effectively controlled by shareholders or members that are located in the proximity of the renewable energy projects that are owned and developed by that legal entity;
- (b) the shareholders or members of which are natural persons, SMEs or local authorities, including municipalities;
- (c) the primary purpose of which is to provide environmental, economic or social community benefits for its shareholders or members or for the local areas where it operates, rather than financial profits;"

'**citizens energy community**' means a legal entity:
which is based on voluntary and open participation, effectively controlled by shareholders or members who are natural persons, local authorities, including

municipalities, or small enterprises and microenterprises. The primary purpose of a citizens energy community is to provide environmental, economic or social community benefits for its members or the local areas where it operates rather than financial profits. A citizens energy community can be engaged in electricity generation, distribution and supply, consumption, aggregation, storage or energy efficiency services, generation of renewable electricity, charging services for electric vehicles or provide other energy services to its shareholders or members;"

The EU has thus established two different types of LECo that mainly differ by the requirement or not for the energy production to be renewable and by the type of actors that can participate. The legal definitions specified in those directives need to be interpreted by each EU member state and implemented into their respective legislation.

Belgium

In Belgium, the Brussels-Capital Region, the Flemish Region and the Walloon Region each have to implement the EU directives in their own legislation since environment and energy are regional matters (except for nuclear which is not of concern here).

Wallonia has been a forerunner with energy communities and is among the first places in the EU to have provided legal ground for them through the adoption of a decree on the 2nd of May 2019 [16]. It revises the 2001 Decree on the organization of the regional electricity market, the 2002 Decree on the organization of the regional gas market and the 2017 Decree on tariff methodology applicable to gas and electricity distribution system operators. It partially implements the CEP legal disposition and focuses on renewable energy community (*Communauté d'Energie Renouvelable*, CER) and of collective self-consumption (*électricité autoconsommée collectivement*).

In this decree, a **renewable energy community** is defined as a legal entity formed by a group of participants in order to share, through the public distribution network or local transmission network, electricity produced solely using renewable energy sources or high-quality cogeneration by production units and, if applicable, storage units, owned by the said legal entity, within a local perimeter where it exerts its activities and that has as primary objective to provide environmental, economic and social benefits to its participants rather than aiming at financial profit.

The **local perimeter** is defined as an area where the junction points for take-off and injection are found downstream from one or several public medium/low-voltage transformer stations. The said junctions points also have to be located within a geographical perimeter such that the portion of network is technically, socially, environmentally and economically optimal in order to promote the local collective self-consumption of electricity.

Any physical person, local authority or small and medium-sized enterprise that is located within the defined local perimeter is free to join a REC. Each member of the community has to conclude an agreement that states the rights and obligations of each party and that regulates the electricity exchange among the participants. The electricity that is produced by the REC and consumed by the participants within the same quarter hour period is considered as **collective self-consumption**.The REC needs a license to operate on the local distribution or transmission network, but not for electric exchanges that are part of the collective self-consumption.

This decree is however not applicable yet because it requires executive measures to be taken by the Walloon Government. More legal texts are on the work, like a draft decree that was accepted by the Walloon Government on the 16th of December 2020 and is expected to be definitively adopted mid-2021. As of this writing, it hasn't been adopted yet. With the time needed for the Government to do the necessary accreditation, the first official REC should begin in 2022. [18] There are however several pilot projects that are currently ongoing.

In the Brussels-Capital Region, there is no general legal disposition that allows for REC. However, a legal ordinance passed on the 23rd of July 2018 [1] specifies that the energy market regulator, Brugel, is allowed to give derogation to specific projects so they can act as energy communities.

More precisely, Article 90 of this ordinance specifies that Brugel is allowed to adopt, for a limited period of time, specific market and tariff rules for areas that are geographically or electrically delimited. Those areas are developed specifically by the achievement of innovative pilot projects and in particular for the development of solutions to the issue of the connection of decentralized productions to the distribution network.

As of this writing, three projects are currently being reviewed by Brugel and two have received an exemption [7]. The first one to receive that authorization is *Les Bambins*, a project in which a community of about 15 people share the energy produced by photovoltaic panels located on a school and by a private individual. The second one is *Greenbizz.energy*, a project in which the electricity produced by the photovoltaic panels by Greenbiz, an incubator for sustainable projects, is shared with 19 businesses located within Greenbiz. The project leader is WeSmart and this REC uses its digital platform, the company and its platform will presented in more detail in Section 1.2.

In the Flemish Region, there is ongoing work to implement a regulatory framework for local energy communities [14]. Some pilot projects already exists and their experiences are taken into consideration to create appropriate legislation that would promote the development of such communities. One of the existing pilot project is the Beacon ECCO Bocholt [5], which is centered around local landscape management with biomass heating. It is organized as a cooperative and involves various actors such as farmers, schools, municipalities and management organizations. As a Beacon ECCO, it is part of the EU project Interreg North-West Europe ECCO [17], which aims to promote the development of local Energy Community Co-Operatives (ECCO).

We can thus see that the legislation relative to energy communities are evolving fast and will probably continue to evolve in the next few years, both in Belgium and the rest of the EU. WeSmart is following this evolution and is working now in order to be an active actor in the evolution of energy communities.

1.2 WeSmart

WeSmart [42] is a young company that has for main focus to support energy communities and help its clients in all its different aspects. They work with any type of communities, be them REC, CEC, or outside of those precise definition. The company is located in Brussels, but they also work internationally.

1.2.1 Activities

As exposed in Section 1.1.2, energy communities don't have a specific legal recognition in Belgium yet, but it is still possible to have pilot projects taking different forms. So, one of WeSmart activities is to assist its clients in the process of joining an existing energy community or creating a new one. In the latter case, the WeSmart team can do all the work needed in order to set up the community. They help their client understand the applicable legislation and analyze their needs to figure out what would be the best form for their project. In case a legal exemption is needed, as in the Brussels-Capital Region, they prepare the application with the client. They also assist them for other aspects, like to determine the best way to organize the community, to establish the tariff scheme for the collective-self consumption, to find new members...

Whether a new community is created or a user is joining an existing one, WeSmart also tries to bring together members that have complementary production and consumption profiles. This is indeed a very important factor to maximize collective-self consumption. It is also a challenge since it requires finding participants that are energetically complementary, reside in the same local area (for which the criteria may vary depending on the situation) and willing to engage in an energy community. So, in order to achieve that goal, WeSmart individually matches each user to the community that is most appropriate for them. They also actively look to motivate new participants through workshops and other events.

Beyond the creation of new communities, WeSmart also provides its services to businesses or to already established communities. Their main product is a digital platform to track and manage the production and consumption of energy. For communities, it also organizes the distribution of energy between each member and with the national grid.

First, using energy sensors and smart meters, it gives precise information on consumption. This includes how much energy was consumed, at what time, where and how much it has cost. That information is topped up by suggestions of action that would help with energy reduction and are given by an AI. The goal is to permit users to better understand their energy production and consumption so they can better act on it to reduce consumption or better adapt it to the production. Additionally, the AI can generate alerts in case something suspect is detected, like high consumption in an office on a day off. It also gives prediction on the future energy production so users can plan ahead and for example program energy-intensive tasks during the higher production periods.

Community managers can also use the platform to set up key performance indicators and targets, for example lowering consumption to a certain level. They can then track the evolution of their community towards those goals.

For energy communities, the platform tracks how much energy was produced and consumed by each participant. It can thus calculate for each user how much was part of collective self-consumption or was offtake consumption[1]. The invoicing is then done according to the specific energy community rules. The platform automates that accounting and invoicing.

[1]Offtake consumption refers to the energy that was consumed but not self-produced. It thus need to be bought from a power supplier.

Chapter 2

Project development

As introduced in Section 1.2.1, the WeSmart platform includes an artificial intelligence (AI) that provides recommendations to reduce energy consumption. WeSmart wants to modify that AI in order to increase its impact and better take the particularities of each user into account.

My goal as part of this master thesis was to work on possible improvements for that recommendation AI. WeSmart gave me a large freedom on what those improvements could be and how they could be implemented. I was also encouraged to propose creative solutions that were different from the ones already used in the first version. This includes a complete modification of the AI if appropriate.

After apprehending the context (see Chapter 1), the next step in my work has been to understand WeSmart needs regarding the AI, analyze how it could be improved and formulate different suggestions. That process took about two months. During that time, I held several meetings with WeSmart to present my ideas, get feedback and adapt accordingly. Unfortunately, I could only go to WeSmart office once during that period due to the coronavirus pandemic, so most of our exchanges have happened virtually. After the initial improvement propositions, I have maintained regular contact with WeSmart to present the different stages of advancement of the project and adapt my work to their feedback.

2.1 Analysis of existing AI and challenges

For confidentiality reasons, the presentation of WeSmart prototype will not go into much details and the actual rules and recommendations crafted by WeSmart are not going to be used in this manuscript. Therefore, all examples presented in this *document have been created by me and have not been taken from an actual situation. They still remain plausible and useful to the discussion.*

The AI developed by WeSmart takes as input data on the energy consumption and production of a user and returns recommendations. It works with a rule set containing several rules that links conditions on input data to an appropriate recommendation. High-level examples of such rules can be found in table 2.1.

Condition to validate	→	**Recommendation(s)**
Yearly consumption is higher than yearly production by more than 300kWh	→	Add one or more photovoltaic panels
There was a high peak of consumption[1]	→	Try to spread use of energy intensive devices through time in order to avoid costly high peak
Afternoon production is higher than consumption and morning consumption is higher than production	→	Run devices with high consumption during afternoon if possible; Join energy community
Production is not sufficient to cover consumption	→	Add production units; Join energy community
Average production is higher than average consumption	→	Disable some production units when production is highest; Join energy community
Consumption is high at night	→	Reduce number of devices running at night; Add windmill

Table 2.1: Example of rules that could be used for the AI

There are currently rather few rules, but more are expected to be added in order to encompass for more situations and provide users with better advice. Those rules can be viewed as the critical part of the recommendation AI since their relevance and accuracy have a direct impact on how useful the recommendations will be. For those reasons, I decided to focus my work on them.

The creation of the rule set has to be done manually because it requires expertise in the energy field and careful data-analysis in order for the rules to be relevant. There are different challenges in the design of a good rule set, which include

[1]High peak of consumption refers to an energetic demand that is significantly higher than average consumption. High peaks put a strain on the electric grid and power suppliers typically charge more for energy consumed during a peak.

- keeping rules consistent with one another (no rule contradicts another one within the same set),
- avoiding unnecessary repetitions (distinct rules contains the same information),
- setting appropriate threshold values for each rule,
- adapt a rule set to a specific user,
- handling a growing number of rules.

2.2 Design of tool assisting rule creation

To answer the challenges described above and take into account the need for a human expertise, the solution I propose is the creation of a tool that would help humans in the design of rule sets. By helping them design more accurate rules with no error, it would result in more relevant rule sets and improve the overall performance of the recommendation AI. This section describes the general design I conceived for an efficient tool.

To best accomplish the goal of support for the rule creation process, such a tool that assists the creation of a rule set should include two main features. First, this tool needs to reduce the difficulty of keeping rules consistent within a large set of rules. This should make their crafting easier and less error prone. Second, it needs to help in finding the best threshold values for a condition triggering the rules. (Ex: The daily consumption could be considered abnormally high above a threshold of 40kWh.) This part can be extended in order to provide different threshold values and different recommendations depending on the specificity and preferences of the consumer.

Feature 1 : *Verify the coherence of a given set of rules*
This feature aims to facilitate the creation of a large set of rules by providing automatic verification of the compatibility and coherence between the different rules within the set. This verification is done by checking if two rules could apply to the same situation and highlighting such cases. This allows the user to easily see and fix a rule that would not yield the wanted result.

Here are some example cases of such situations:
Example 1 :
Rule 1: Daily consumption is between 80 and 150 kWh → Recommendation 1

Rule 2: Daily consumption is between 70 and 120 kWh → Recommendation 1
The conditions of rule 1 and 2 are both validated in the range of value 70 to 120.

Example 2 :
Rule 1: Daily off-take consumption is between 80 and 120 kWh → Recommendation 2
Rule 2: Daily off-take consumption is between 80 and 120 kWh and Today is Sunday → Recommendation 3
The situation needs to be clarified for days that are not Sunday, otherwise the conditions for both rules are validated for those days.

For this feature my original suggestion included automatic correction of a rule set in the cases of redundancy (e.g. if two rules can cover the same situations and have the same recommendation) or contradiction (when two rules can cover the same situations and have contradictory recommendations). However, after advancing in the development of this solution and further discussions with WeSmart, it came to light that automated modifications of the rule set was not desirable. The detailed discussion of the reasons for this is presented in Section 3.4.7. The complete evolution of the solution from the original idea to the final is explained throughout Chapter 3.

The tool should thus highlight the situation that requires potential action so humans experts know they may have to address those cases. By relieving them of the weight of verifying themselves what the relationships between each pair of rules are, it should make the rule crafting process faster and easier as well as reducing errors in the rule set.

Feature 2 : *Find the best thresholds values for conditions to be considered met*
This feature aims to optimize the values in the conditions in order for them to be considered met by the input data and thus the recommendation to be applicable. For example, a peak of consumption could be considered high at different values depending of the network and the consumer. It is necessary to find the appropriate threshold to avoid false alarms or missing situation that should be notified to the consumer.

For this, an optimization criteria has to be defined. It could for example be the financial gain, the reduction of energy consumption or a ratio between the two. The chosen values could be learned from past production and consumption data or data representative of the same type of consumer (ex: family of four, small office,...).

Making the thresholds more accurate will result in more relevant recommendations given by the AI, which will allow the consumers to better adapt and reduce their consumption.

Feature 2 - Extension : *Customize rule set for a specific user*
WeSmart wants to fit the needs of different types of users, such as households with different numbers of inhabitants, offices, stores of various sizes, production sites,... Those various users have different profile types, different orders of consumption and different goals. Their consumption habits or production capacity could also change with time. The more the recommendations are adapted to their specific profile and needs, the more accurate and useful they can be. It would therefore be necessary to have rule sets that can easily be adapted to different users needs and to have a tool that supports that.

Here are different ways in which the recommendations could be customized to better fit the needs and situation of different users:

- Adapt the condition threshold values depending on the type of user

  Since users have very different consumption profiles and production capacity, most of the condition thresholds have to be adapted to that. The tool should thus make it easy to modify a rule set to fit new needs.

- Adapt and optimize the condition thresholds and recommendations depending on the specific goals and priorities of the user

  Some users may only care about the financial aspect and not the environmental one, so they may only want recommendations that have a positive financial impact. Others may have more fund and be concerned about the environment above all, so they may want the best recommendations on reducing the environmental impact of their energy consumption, even if there is no expected financial gain. It would also be possible to have a company that absolutely wants to reduce the energy consumption to a certain level, for example to obtain a specific certificate, even if there is a consequent cost to it, but that doesn't care to reduce the energy consumption further once that goal has been reached. Accommodating for those different priorities in the rule set would provide each user with more adapted recommendations.

- Allow users to set goals and adapt the rules and recommendations to those goals and the level of advancement towards them

  The threshold for alerts could be adapted to specific goals and users could get notifications to let them know how close or not they are to reaching them. Several intermediate targets could be established and the notifications

adapted for each milestone. Those targets as well as notifications could also be used for the gamification of energy reduction process. This can help better encourage participants to adapt or limit their consumption, for example in the case of households within the same community or employees in a company.
Examples of possible notifications :
- "Your offtake consumption is x kWh more than your goal, you should add more PV unit."
- "Your offtake consumption is only y kWh more than your goal, you can adapt your behavior to reduce that small different and reach your goal."

- Allow users to give information about their personal situation and life habits

  Many elements can have an impact on consumption habits and energy production. The more complete and individualized the information about the users are, the more adapted the recommendation can be, which make them more useful for the user.
  Examples of relevant personal information :
  - I am unable to add new solar panel (so that recommendation shouldn't be given anymore).
  - Nobody is home during work hours of weekdays (so consumption should be very low during that time).
  - Someone in the household works from home (so higher consumption than if there were nobody is normal).
  - My work shift changes weeks by weeks (so frequent changes of consumption profile is expected).

- Allow users to give feedback to recommendations and adapt the rule set accordingly

  With time or with the user action, a recommendation may no longer be useful or another one may become more adapted.
  Examples of such situations :
  - The user has already checked all of their devices for abnormal consumption and is not willing to buy new ones (so a recommendation to check or replace appliances shouldn't be given anymore and the base consumption considered can be increased).
  - The user has installed new solar panels (so the corresponding recommendation shouldn't be given anymore before some time, an increase in production is expected and some previous threshold may need to be recomputed).

2.3 Feature further developed

The previous section presented the different features a very complete tool should have. The full development and implementation of such tool would be outside the scope of a master thesis. Thus, in agreement with WeSmart and my adviser, we decided that I were to further develop the first presented feature only, the verification of the relationships between different rules. At this stage, the automated correction for some problematic rules is considered desirable and should be studied too. The second aspect presented will thus not be discussed further in this document.

The objective is now more specific, it is to design and implement a smaller tool that detects the relevant relationships between rules so it can highlight the conflicting situation in which the human expert may need to make modifications and potentially make automated corrections. This could involve a redesign of the rules formulation and of the general functioning of the recommendation AI if it is beneficial for the rules management.

To achieve this, different questions still needs to be answered. The remaining tasks include to:

- determine the best representation for the rule set and its rules,
- analyze what potential adaptation to the existing AI would be relevant,
- establish what the relationships of interest are,
- find the most appropriate algorithm to detect such relationships,
- study the possibility and relevance of automated modification,
- implement a prototype of the tool.

The three following chapters of the master thesis will be dedicated to the development of that solution. Chapter 3 presents a review of literature on potentially useful fields and the evolution of the answers to the above questions. Chapter 4 is focused on the theoretical presentation of the solution, including a description of the final version of the recommendation AI. Chapter 5 describes the implemented tool.

Chapter 3

Review of literature and project evolution

Now that a more precise objective has been defined, the next step is to find a satisfying way to implement it. As stated in Section 2.3, the goal is to make sure that a rule set of energy rules, associating recommendations to energy observations, doesn't contain contradiction nor redundancy. Those problems should then be corrected, which would improve the rule set and the quality of the recommendations given to users.

To design a good approach to accomplish this task, some questions needs to be answered:

- How should the rule set and the rules be represented ?
- How should the recommendation AI interpret the knowledge contained in those rules ?
- What are the relationships of interests ?
- How to spot those relationships in a rule set ?
- How to automate the correction of problematic relationships ?

The solution I want to develop has not been previously implemented to the best of my knowledge, so there is no preexisting answer to those questions. However, rules and identification of their relevance and the relationships between them is something that is used in many applications. Similar problems have thus been explored in scientific literature. So, the methodology I have used to find the most appropriate solution has been to study various approaches used in other fields. I have then analyzed if and how they would also be applicable to the case of energy rules. I have reviewed the analysis and use of rules in three different domains: associative classification, expert systems and firewalls. This chapter presents the different techniques I have reviewed in those fields and the conclusion I reached from their study regarding my application. This chapter thus also describes the evolution of the proposed solution from the original assumptions to the final choices.

3.1 Original assumptions

The task of the recommendation AI is to associate a set of data to a particular recommendation. This can be viewed as classification, which aim is to predict the class label of a certain data object [39]. For the recommendation AI, each class label corresponds to a different recommendation. The classification problem is central in machine learning and has been widely studied.

There exist several machine learning algorithms that create a variety of rules and then select the useful ones out of the pre-established set, including by the detection of redundancy or contradiction. If they can be adapted to the case of energy rule sets, such an approach could provide detection and automated correction of problematic rules.

Even though energy rules require human expertise to be created, they are tied to energy data. WeSmart has access to such data through the information collected by its platform and it could be possible to exploit that for rule analysis. Analyzing and adapting methods used in machine learning thus seems to be an appropriate solution, so I have started by studying literature covering that kind of problems.

3.2 Associative classification algorithms

3.2.1 Associative classification and energy rule set

Associative classification (AC) is a branch of data mining, which is itself a branch of machine learning. It combines classification with association rules discovery, which aims to find correlations between items in a transactional database. AC algorithms discover class association rules (CARs) in a training data set, then use them in order to build a classifier, which will be used to predict the class of unseen data. [39]

A **training data set** T of size $|T|$ has m distinct **attributes** $A_1, A_2, ...A_m$ and contains a number of training objects. Those **training objects** are each defined by a combination of attribute names A_i, attributes values a_{ij} and a **class** c_j. The list of all classes is denoted C. [39]

Class association rules associate a consequent on their right hand side to the antecedent on their left hand side. The antecedent is a set of **items**, which can each be described with an attribute name A_i and a value a_i. The set of items is called an **itemset** and corresponds to a set of disjoint attribute values in a training object. It is denoted $\langle(A_{i1}, a_{i1}), ..., (A_{ik}, a_{ik})\rangle$. The consequent is a class. [39]

A CAR can be represented in the following form:

$$(A_{i1}, a_{i1}) \wedge ... \wedge (A_{ik}, a_{ik}) \rightarrow c$$

This form is also well suited for energy rules, as shown in the following examples, where rules are expressed first in written form then in the CAR format:

- Average night consumption is high → Reduce use of energy at night

  $(AvgNightCons, [a, b]) \rightarrow$ *{Reduce use of energy at night}*

- Afternoon cons. is lower than production, and morning cons. is higher than production → Run devices with high consumption during afternoon if possible; Add storage system

  $(AvgAftCons, [a, b]) \wedge (AvgAftProd, [b+thresh1, c]) \wedge (AvgMornPord, [d, e]) \wedge (AvgMornCons, [e - tresh2, f])$
  $\rightarrow$ *{Run devices with high consumption during afternoon if possible; Add storage system}*

- A large proportion of energy produced is reinjected and the user is not in an energy community yet → Integrate energy community

  $(AvgOfftake, [a, b]) \wedge (InCommunity, False) \rightarrow$ *{Integrate energy community}*

Other important terms and concepts in CA include the following [39], [41]:

- A **ruleitem** is the combination of an itemset and a class c, denoted $\langle itemset, c\rangle$.
- The **actual occurrence** ($actoccr$) of a ruleitem r is the number of training objects in T that matches the itemset of r.
- The **support count** ($supp$) of a ruleitem r is the number of training objects in T that matches the itemset of r and belong to the class c of r.
- The **occurrence of an itemset** i ($occitm$) is the number of training objects in T that match i.
- The **confidence of a ruleitem** r corresponds to the ratio of its support count over its actual occurrence, $conf(r) = supp(r)/actoccr(r)$.

- An itemset i is a **frequent itemset** if, for a defined *minsupp* threshold, $occitm(i)/|T| \geq minsupp$.
- A ruleitem r is a **frequent ruleitem** if, for a defined *minsupp* threshold, $supp(r)/|T| \geq minsupp$.

The **classifier** that will label new data objects takes the form of a set of CARs. In [39], its construction using AC is divided into the four following steps:

1. Discover all the frequent ruleitems
2. Produce all confident CARs using the frequent ruleitems found during Step 1
3. Select a subset of CARs from those created during Step 2
4. Evaluate the quality of the classifier formed by those rules

This procedure is rather different than the one used for the creation of an energy rule set, since in the latter the rules are manually crafted rather than learned through data mining. However, the verification of the coherence of an energy rule set, which is the problem of interest, seems comparable to the third step. Indeed, they both require to analyze the rules and determine which ones are relevant, redundant or contradictory. A new set of rules can then be produced from the original one. The problem of understanding the rules and resolving the problematic relationships between them has been the subject of many researches [8]. The process of selecting some rules and discarding others is referred to as **pruning**. Even though the third step is the one focusing on the selection of the rules that will be used in the classifier, pruning usually also happens in the first and second steps too.

I have reviewed a selection of four pruning techniques that were presented in [39] to see if and how they can be adapted for the case of energy rules. The evaluation of performance could also be interesting in both cases and could be studied if this approach is confirmed to be successful.

3.2.2 Pruning techniques

χ^2 testing

χ^2 testing is a statistical method that allows to determine whether two variables are correlated or independent [36]. For population of subjects, the independence test indicates if they are positively correlated or not, using comparison of expected frequencies and observed frequencies. For AC algorithm, it allows for example to test if the antecedent of a rule is positively correlated with its class [39]. The rule can then be kept if it is the case or discarded if it isn't, like in CMAR [41].

Redundant rule pruning

It is possible to have rules that share the same items in their antecedent, which causes redundancy. The antecedent of a **general rule** is a subset of the antecedent of the **specific rule**. If the rule r is general with regard to the rule s, redundant rule pruning discards s if it has lower confidence than r. This pruning technique was introduced in CMAR [41]. In has also been used by other CA algorithms, like in [4] or [3].

Database coverage

This pruning technique is defined in [28]. It requires the rule set to be sorted, with all rules having different precedence. The general idea is to select a set of high precedence rules from the set S of generated rules in order to cover the training data D. The algorithm goes through the rules from higher to lower precedence. It marks all the rules that correctly classify at least one training object in D and remove the correctly classified (covered) objects until there is no rules or no training object left. The rules that have not covered any training objects are then discarded and the remaining ones are selected to form the classifier. A calculation of the errors made by the classifiers can also be made and the rules causing too many errors discarded. Other uses of this technique for CA algorithms can be found for example in [29], [37] or [41].

Pessimistic error estimation

The pessimistic error estimation, which is a measure based on training objects, can be used to calculate estimated errors for a certain rule r with or without removing a specific item i from its condition [39]. If the estimated error of the newly obtained rule r' is lower than that of r, r' should replace r in the rule set. This estimation has been used in AC algorithms such as the ones in [29] or [40].

3.2.3 Classification of data

In all the rule-based AC algorithms, the rules are sorted before they are used to predict the data [39]. The sorting procedure is often also used to select the rules that will be used for the classifiers, like for the algorithms proposed in [28], [37], [38], [40], [41]. For example, we have seen in 3.2.2 how the ranking can be used for the database coverage pruning technique. The ranking is usually evaluated according to different parameters, such as the confidence, the support count, the number of items in the antecedent or the order in which the rules are created.

Once the classifier has been built with the sorted rules, there are two main methods to predict the class label of a new object, according to [39]:

- Maximum likelihood based prediction : The rule with highest precedence that matches the new object is used to label it.
- Multiple rules based prediction : Several rules matching the new object are taken into consideration to label it. The selected label can be chosen according to different techniques that won't be described here.

In both cases, the new object is assigned one unique class.

3.2.4 Discussion and conclusion for energy rule set and AI

AC pruning methods

The expectation before starting the analyze of AC and its pruning techniques was that some of those methods could also be adapted to manually crafted rules, even though they were created for rules generated from a training data set. When studying those techniques, we see that they all rely on information that can only be obtained by the use of a labeled data set. Indeed, χ^2 testing uses expected and observed frequencies. Redundant rule pruning relies on confidence. Database coverage requires a data set to be executed. The pessimistic error estimation is measured based on training objects.

WeSmart does have access to energy data, but the available data sets are not labeled. Labeling them by hand would require an enormous amount of work and careful analyses of each object, which is unfeasible. Even if it was possible, the aim of this project is to ease the work of the human experts crafting the rules. It would thus be counterproductive to shift the work from verifying the coherence of energy rule sets with no assistance to labeling extremely large data sets in order to make that assistance possible. Labeling the data set in an automated way would require to have an algorithm for that, which would need to be created using expert knowledge. This actually corresponds to the problem of associating recommendations to energetic observations. Since the objective of this thesis is to make this problem easier to solve, an automated labeling of the data sets wouldn't help with regard to that goal.

In response to the lack of labeled data, unsupervised learning techniques could be considered. However, mining algorithms generally aim to find recurring structures in data set, which is not an approach I recommend for energy rules. When creating the rules, the aim is to formulate energy recommendations and associate them with energy observations, which is usually not something that can be deduced simply from patterns in data, but actually requires expert knowledge. Beyond the creation of the rules, relying on training data samples for their verification and correction doesn't seem to be a relevant approach either. Indeed, many interesting rules may cover abnormal or rare observations that would be absent or scarce in a training set. I thus think it is better that the rule creation process is dissociated from specific training sets and keeps on being done manually.

Another problem of the comparison with AC rule sets is that they need to be sorted, both to be used as classifiers and to apply certain pruning methods, like database coverage for example. Energy rules can't be sorted using the same indicators than CARs. One reason is the lack of access to labeled training data set. Another is that the objects in a training set would probably not provide useful information regarding the relevance and precedence of rules anyway. Another way of ranking rules would need to be used and would most likely require to be manually done by human experts. This would thus add a difficult task for them and would increase the rule crafting task complexity, instead of providing some relieve.

Classification

Studying the way the AC classifier is conceived also brought new considerations regarding the original assumptions made in Section 3.1.

The AC classifier, like all classifiers, associates one unique class to each considered data object. This means that, if the recommendation AI uses classification, only one recommendation can be associated to each energy observation. The recommendations thus each need to be very precise, which would quickly lead to a large number of recommendations.

As example, let's consider the three different following pieces of advice:

- Rec1 : Run devices with high consumption when production is higher
- Rec2 : Integrate an energy community
- Rec3 : Add storage system

All those three pieces of advice can apply to situations in which there is a mismatch between the consumption and the production, meaning they are times when the production is higher than consumption and other times when it is lower. However, it may sometimes be more appropriate to give the user only one of them, while other times better to give two of them or even all three. The right advice to give may depend on different factors, like how important is the mismatch, whether the user is already in an energy community, what is the ratio of the average production and consumption,...

In order for an AI using classification to return the most adapted recommendation possible, there would need to be 7 classes to cover for all possibilities.

- Class 1 : Rec1
- Class 2 : Rec2
- Class 3 : Rec3
- Class 4 : Rec1 + Rec2
- Class 5 : Rec1 + Rec3
- Class 6 : Rec2 + Rec3
- Class 7 : Rec1 + Rec2 + Rec3

Classification thus leads to a very large number of very precise classes, containing redundant information. This would make it harder for the human experts to handle those classes and ensure that they are all coherent with each other. The problem of contradiction and redundancy would thus shift from handling a large number of rules to handling a large number of classes. It would also increase the difficulty to verify the quality of the rule set every time a new rule, a new class or a new attribute is added. Regarding those elements, I don't think anymore that a classification model would be very relevant for the recommendation AI.

Two alternative to the association of a unique class to each observation can be considered:

A first possibility would be to associate a class to a rather large category of situations, then perform further checks on each data object to determine which precise recommendation should be selected. New rules should thus be implemented for each category of recommendation in order to select the actual recommendation given as output. This would also require all the precise recommendations to be dissociated in different categories, which could put an important constraint on the different possible recommendations and may lead to redundancy between categories. For example, it seems logical that two distinct categories of situations would be (1) a larger production than consumption and (2) a larger consumption than production. Yet, the recommendation to join an energy community would be interesting in both cases and similar verification would need to be implemented for both categories or dealt with in another way. There is thus no guarantee that this alternative would decrease the complexity of the rule set and its creation.

A second possibility would be to allow objects to be associated with more than one "class", or label. Then, the precise recommendation that may contain several pieces of advice could be split into smaller unitary recommendations. This way, an energy observation can be associated to several unitary recommendations that compose a more complete and precise recommendation. This would reduce the number of labels and rules needed, as well as making each of them less complex and cumbersome.

Let us look at the example already used above. Instead of having 7 different classes and at least as much rules defining very precise situations in their antecedent, there would be 3 classes and 3 or more rules. The rules should also be more simple since they need to cover less specific situations.

- R1: Mismatch between production and consumption $\rightarrow$ Rec1
- R2: Mismatch between production and consumption[1] in range $[a, b]$ and user not in energy community $\rightarrow$ Rec2
- R3: Mismatch between production and consumption in range $[c, d]$ $\rightarrow$ Rec3

If the energy observation for a user who is not in an energy community corresponds to a mismatch between production and consumption in range $[a, b]$ and outside of $[c, d]$, the recommendation produced will be the combination of Rec1 and Rec2.

This alternative to classification seems promising in order to express precise sets of recommendations without putting too much complexity in the rules. I have thus chosen to select this solution for the construction of the energy rule set. This also means that the recommendation AI needs to be able to select several rules and output a set composed of one or more unitary recommendations.

Conclusion

Since all pruning algorithms used in AC I have encountered relied on the possibility to use a labeled training data set, they can't be properly adapted to the case of the energy rules used for the recommendation AI.

Beyond AC and its specific techniques, the classification problem isn't the best way to look at the generation of recommendations based on energy observations. Indeed, it would require a very large number of classes that would be difficult to

[1]Mismatch corresponds to the addition of the energy reinjected to the grid and the offtake bought from the grid.

manage for the human experts crafting the rules. Instead, a better solution would be to associate several unitary recommendations to the same observation, which should lead to less numerous classes and more compact rules.

The main challenge that come with this solution is making sure that all the recommendations that are given simultaneously are complementary and don't present redundancy nor contradiction. This question goes beyond the verification of the rule set and is also related to the way the recommendation AI handles several matching rules. For example, should all matching rules be selected ? Should there be a selection of certain matching rules ? What would it be based on ? That question will be further analyzed in Section 3.3.

Since AC techniques proved to not provide good solutions for the problem at hand, a new approach needs to be considered. CARs and there creation was too different from the energy rules to be of use, so looking for a solution in a domain that uses manually crafted rules could be a more successful approach.

3.3 Recommendation AI as an Expert system

After the realization that machine learning techniques could not be efficiently adapted for the verification of energy rule sets, as explained in Section 3.2.4, my following approach was to look for a domain that has more similarities with energy rules. In particular, I considered the use of manually crafted rules as an important criteria.

One such application is the study of firewalls since their rules are also created by domain experts. It is however a very specific field and there are also differences between firewall and energy rules. There is thus no prior guarantee that it uses techniques that are adaptable to energy rules. In particular, the recommendation AI needs to select and return the results obtained from several matching rules (see Section 3.2.4), while the firewall policies only select one matching rule to be applied.

So, before diving in that precise field, I have first done a brief review of the more general field of expert systems, which are applications using expert knowledge to solve particular problems. Firewalls are an example of expert systems. My first goal in this study was to determine if there is another application that seems to be a better parallel to energy rules before I start analyzing firewalls. My second goal was to see how expert systems use their rule set and use that new insight to help me in the design of the recommendation AI, with a particular focus on the simultaneous selection of several rules.

3.3.1 Expert systems and recommendation AI

According to [33], an expert system (ES) is a "computer application that performs a task that would otherwise be performed by a human expert". It allows to transfer knowledge from human experts to the application, so it can use it to make inferences with regard to a specific situation [35]. The ES can then reach a conclusion that can be given to the user, much like a human consultant. In the case of energy recommendations to improve one's consumption, the recommendation AI provides users with advice that could have been otherwise given by an expert in energy. It thus fit the definition perfectly.

There are many different kind of expert systems, [35] divides them in eleven categories: rule-based systems, knowledge-based systems, neural networks, fuzzy expert systems, object-oriented methodology, case-based reasoning, system architecture development, intelligent agent systems, modeling, ontology, and database methodology.

In rule-based systems, the information is represented as a set of IF-THEN rules with an antecedent and a consequent. The rules are then used to perform operations on data, make inferences and reach a conclusion. The knowledge is thus represented explicitly through the rules rather than as procedural code. [22], [30], [35] The recommendation AI is thus obviously a rule-based system since all its knowledge is stored using such rules. This is also the case of firewalls. Many other examples of rule-based ES can be found in [30].

More specifically, the recommendation AI is also a forward chaining system, because it starts with a fact and uses the rules in order to reach a conclusion about that fact [22]. This is opposed to backward chaining systems that are given a hypothesis and use the rules to determine whether it is correct.

The functioning of forward chaining systems is represented in Figure 3.1, from [22]. The *facts* to consider are stored in the *working memory*. The *inference engine* determines which rules from the *rule base* those facts are matching. All the matching rules form the *conflict set*. A *conflict resolution strategy* is used in order to determine which rule from the conflict set will be applied (fired). If the firing of the selected rule has lead to new facts, they are added to the working memory. The process is stopped when a termination criterion is met, for example if a conclusion has been reached.

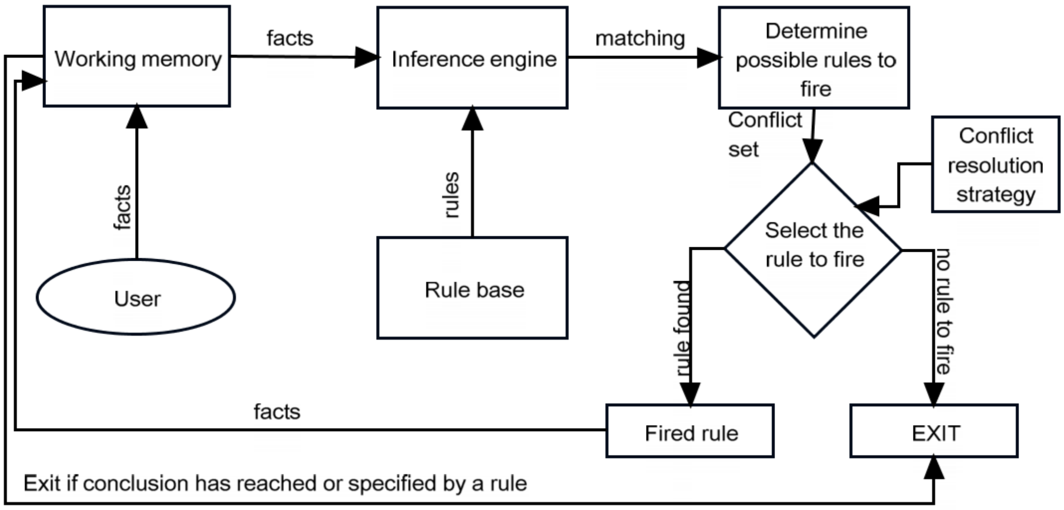


Figure 3.1: Forward chaining rule-based expert system diagram, image from [22]

3.3.2 Discussion

Since the rule representation chosen at this stage requires to fire (select) several rules at the same time (see Section 3.2.4), it would be interesting to find an example of ES that also fires several rules simultaneously. However, the general procedure for rule-based expert systems explained in [22] only intends for one rule to be fired at a time. Article [30] gives a list of existing rule-based systems found in the literature before its publishing in 2019, with the aim to be as exhaustive as possible. Out of the 78 systems considered, the presentation of none of them indicates the use of several rules simultaneously. As I could not find a system firing several rules simultaneously, I chose to adapt the general functioning of forward chaining rule-based expert system to fit the wanted behavior for the recommendation AI, which was first presented in Section 2.1 then further refined in Sections 3.1 and 3.2.4. Its final behavior and functioning will be explained in Section 4.1.

Since this review of expert systems didn't allow me to find an option that appeared better suited, I decided to proceed with the analyze of firewalls and the creation of their policies in order to try and find adaptable techniques for the verification of energy rule sets. The outcome of that study will be presented in Section 3.4.

3.3.3 Recommendation AI design

It has been established in Section 3.2.4 that the recommendation AI, that associates a recommendation to an energy observation, can fire several rules simultaneously in order to provide comprehensive and precise recommendations. The precise strategy to determine what rules need to be fired still needs to be decided on. This strategy needs to guarantee that all the recommendations returned simultaneously are coherent, meaning that the advice they contain are not contradictory nor redundant. Several strategies could be used, which are not necessarily all mutually exclusive :

1. In addition to the rules associating energy data to recommendation, additional rules could be created to indicate that some recommendations can't be applicable at the same time.
2. Rules could be assigned priorities, then only the rule(s) that have higher priority compared to others would be selected. (adapted from [22])
3. In case of contradiction, the choice of which rule to select could be made depending on how specific they are (i.e. depending on how many attributes they have in their condition). (adapted from [22])
4. Each rule could be given a weight that could be used to help in the choice of which rule to select or not. (adapted from [22])
5. Rules could all be created in such a way that rules with redundant or contradictory recommendations can never match the same situation, eliminating the need for a conflict resolution strategy.

Solutions 2, 3 and 4 would be very appropriate in order to select one rule among several, which is what is done in [22]. However, they are not sufficient in order to make a selection of several rules, nor to determine which rules have contradictory or redundant recommendations. Additionally, for solutions 2 and 4, there is no straightforward way to determine the ranking between two rules or the appropriate weight for each of them. Human experts would need to determine those priorities or weights, which would be an additional difficulty in the creation of rule sets. Those values could also need to be reevaluated each time a new rule is added, which would lead to more work and make the rule sets hard to modify. The problem of extra work put on experts is also present in the 1^{st} solution, since they would need to create those additional rules too. Since each rule is very specific and many situations can happen, it would be hard or impossible to design general rules to break ties, so all those additional rules would need to be very specific. That solution would thus probably not ease the work of human experts.

The 5^{th} solution, on the other hand, would avoid the necessity to add such new rules or to determine another complex scheme to handle the conflicts between recommendations. The responsibility to avoid redundancies or contradictions would still be put on the human experts because this solution adds a new constraint for the rule creation that they have to take into account. However, it should still lead to an easier management of the rule set than with another of the solutions presented above.

Indeed, the condition in which a rule can be selected or not would only be expressed through the rule antecedent. This is of increased simplicity compared to having several different types of conditions on the rules in order for them to be fired or not, for example the combination of the antecedent with a weight or with another type of extra condition. The more complex the condition for a rule to be fired is, the harder is it for human experts to visualize its behavior. It is in turn harder to make sure that behavior is the intended one and that it is coherent with that of all the other rules. Human experts are thus more likely to make mistakes if the process for rule selection is more complex.

If two rules can be applied to the same situation and contain redundant or contradictory recommendations, there condition should be modified in order to make sure they can only match different situations. The rational behind this is that if the recommendations are different, they should be associated to different situations. If they match the same situations, it is the sign of an error or that the rule condition should be more specific. Detecting such situations is precisely the aim of the verification tool developed throughout this thesis and first presented in 2. The tool thus need to very the type of relationships between the rules to see which ones could be applied at the same time, then check whether those that do have complementary or conflicting recommendations.

Solution 5 would lead to more rules, since they would each need to be more precise, but with a rule set easier to design and to understand. With this solution, the recommendation AI doesn't need a conflict resolution strategy, nor to deduce new facts used in following iterations. In fact, the inference engine just need to go through the rule set and select all rules that match the data given as input. Their recommendation can then be combined and given to the user. The final version of the AI will be presented in Section 4.1.

3.4 Firewalls policies

3.4.1 Firewalls policies and energy rule set

Firewalls are key elements of network security that secure the connection between a trusted private network and an un-trusted external network [25]. It works by verifying each packet coming to or from the private network against a list of rules that determine which packets can go through and which should be discarded.

Each rule associates an action to a specific type of packet, much like energy rules associate a recommendation to an observed situation. Together, those rules make up the firewall policy. In order to know what behavior to adopt for a specific packet, the firewall checks all the rules in the list until it finds one that matches the packet. The order in which the rules are listed has thus an impact on the decision taken, as opposed to the case of energy rule set where all rules need to be considered for every observed situation. The number of rules and the position of the first matched rule in the network have a direct impact on the efficiency of the firewall which in turn directly impacts the network performances.

Different organizations may have different needs which may also change over time. Firewall rules thus have to be adapted and manually crafted. As the number of rules grow, it may become difficult for network administrators to notice possible anomalies that can badly affect the firewall accuracy or efficiency, much like it becomes harder to craft a good energy rule set with an increasing number of rules. Firewall and energy rules also have a very comparable structures, with a condition composed of an ensemble of attributes with specific values and a decision that is respectively an action[2] or a recommendation. Those are very important similarities that firewall policies share with energy rule sets. The main difference is that the order in which the rules are listed for firewalls plays an important role in the result since only the first matching rule is considered, while the order is irrelevant in an energy rule set since all rules needs to be checked against. Another smaller difference is the importance of the speed in the look up process. While the number of rules impact the speed for both firewall and recommendation AI, this factor is much more crucial for firewalls.

Tables 3.1 and 3.2 show examples of a firewall rule list and an energy rule set presented in the same fashion to highlight similarities. In those tables, the symbol '$*$' represents a wild card accepting any value of appropriate format and can also be used to express intervals of values, such as in the case of IP addresses [21].

[2] The decision of a firewall rule can also include other elements, like logging options, [27] but this was omitted for simplicity.

No.	Protocol	Source IP	Source Port	Dest. IP	Dest. Port	Action
1	UDP	1.1.*.*	*	*	80	deny
2	TCP	1.*	*	1.*	90	accept
3	TCP	2.*	*	2.*	20	accept
4	UDP	1.*	*	*	*	accept
5	*	*	*	*	*	deny

Table 3.1: Example of firewall policy adapted from [21]

Recommendation	OfftakeMorning	OfftakeAfternoon	MaxProd	InCommunity
Add production units	$[200, inf]$	$[200, inf]$	False	False
Add production units	$[0, inf]$	$[0, inf]$	False	True
Join energy community	$[200, inf]$	$[200, inf]$	True	False
Run highly consuming devices during afternoon	$]0, inf]$	$[-inf, 0[$	$*$	$*$

Table 3.2: Example of energy rules in same format as for firewall rules

Given the importance of the rules quality both for the security and speed of the network, there has been intensive research regarding the rules in firewall policies. Part of it focuses on the detection of anomalies by analyzing the list of rules, which mostly corresponds to the detecting which rules could be applied for the same situations. That task is thus very similar to my objective for this thesis. As firewall rules and energy rules have important similarities and a few differences, summarized in table 3.3, adapting an algorithm meant for anomaly detection in firewalls to my application could be a promising possibility. I have thus reviewed some of those techniques and will describe them and their potential adequacy to fit my purpose in the rest of this section. Other techniques have been studied to increase the accuracy and efficiency of firewalls, for example the analysis of network statistics to optimizing the rule order like in [31] or the use of data mining as in [32]. Since those other strategies are not relevant to the energy application, I won't discuss them.

3.4.2 Relationships and anomalies

To detect anomalies in the firewall list of rules, the general technique is to compare rules to one another and, considering the relationship between them, to determine which pairs cause anomalies. In order to do that, most authors precisely define the

	Firewall	Recommendation AI
Procedure	Check packet against every rule until first match	Check situation against every rule to find all matches
Result	1 action	0, 1 or more recommendation(s)
Possible rule consequent	Deny/Accept	Any imaginable recommendation
Creation	Manual	Manual
Rules order	Impact result	Doesn't impact result
Nbr of rules	Impact speed, very important	Impact speed but not very important
Possible values	Number, range of numbers, predefined type (ex: protocols, flags,...)	Number, range of numbers, predefined type, boolean

Table 3.3: Comparison of firewall and recommendation AI

different possible relationships and/or anomalies they consider for their approach. It is interesting to study them to see if they have a potential to be adapted for energy rules. The definitions given will also be useful for the analysis of the approaches taken as examples.

Relationships

The relationship between two rules essentially corresponds to the relationship between the set of packets that matches those rules [2]. The action field thus doesn't have an impact on the definition of the relationships. Following are most of the relationships I have encountered:

Disjunction/Difference : Two rules r and s are disjoint, or different, if the values of at least one of their respective fields are disjoint [2], [6], [34]. There is thus no packet that can be matched by both rules.

Exactly Matching/Equality : Two rules r and s are exactly matching, or equals, if the values of all of their respective fields are equal, as defined in [2], [6], [34]. They thus match exactly the same traffic.

Inclusively Matching : A rule r is inclusively matched by another rule s if, for all respective fields, r's values are either a subset of or equal to s's values, with at least one field for which it is a subset [2], [34]. s thus matches all the traffic matched by r as well as other packets.

[6] breaks this down further and defines **generalization relationship** where r as defined above is generalized by s and **shadowing relationship** where s as defined above shadows r.

Correlation : According to [2], [34], two rules r and s are correlated if they are not disjoint nor the subset of the other. [6] states that those two rules are correlated if there is at least one shadowing relation and at least one generalization relation among their respective fields. Both definitions correspond to the same situation in which some traffic is matched by both r and s while some other is only matched by r and some only matched by s.

A direct parallel can be made between the relationships amongst firewall rules that match specific sets of packets and energy rules that match specific sets of energetic observations. Those relationships can thus be transposed almost directly to the energy rules.

Anomalies

Relationships are usually used to detect and express anomalies. Even though some of them are widely recognized, the choice of anomalies of interest and their definitions vary more among authors than for relationships. With anomalies, the aim is not just to take the relationships between matched packets into account, but to consider the impact of the rules on the security policy. Following are some of the anomalies I encountered in the literature:

Redundancy : In [2], [6], [25], [34], redundancy exists between two rules r and s if they both match the same set of packets and yield the same action, which happens when two exactly matching rules have the same action value. [2] and [25] specify that the redundant rule is the second one to occur. Its removal from the rule list doesn't have an impact on the firewall policy.

Redundancy can also be defined to encompass all configurations in which the removal of the redundant rule doesn't affect the security policy, as in [27]. This paper defines two kind of redundancy. A rule r is **upward redundant** if it isn't the first match for any packet. A rule r is **downward redundant** if, for each packet for which r is the first matching rule, the first matching rule below r has the same decision as r.

Generalization : For [25], [34], a generalization happens when there is a following rule s and a preceding rule r such that r matches all the traffic of s and apply a different action.

According to [6], a generalization occurs when there is a generalization relationship between two rules with different actions. The position of the rules in the list is thus not taken into consideration. This article also distinguishes generalization from **contradiction**, which exists between two rules that match exactly the same packets (equality relationship) but have different actions.

Some authors [2], [25] decide not to consider generalization in their detection algorithms because it can be an intended and relevant design choice.

Shadowing : For [2], [25], [34], a rule r is shadowed by a rule s if s precedes r in the policy and s can match all the packets matched by r. So r is never activated. [2] also requires for the two rules to have different actions.

As in the case of generalization, [6] doesn't take the position of the two rules into consideration, but only the relation between their respective fields. A rule shadows another if there is a shadowing relationship between them and they have different actions.

Correlation : For [2], [6], [34], there is a correlation between two rules r and s if they have different actions and if they each match packets not matched by the other rule as well as a common set of packets matched by both rules (correlation relationship).

Detecting redundancy is relevant in the case of energy rules since such a situation would increase the size of the rule set, which makes its crafting and the detection of errors more difficult, without adding new information. However, the upward and downward nature of redundancy is not relevant for energy rules since their order in the set is of no importance. Since this distinction and the order in which the rules are listed are essential in article [27], it will not be of much relevance for the energy application and I won't discuss this particular paper further.

Adapting generalization, shadowing and correlation to energy rules and detecting those cases would be very interesting because it allows to see where two different recommendations can be applied to the same situation. When such cases happen, it is important for the human expert to verify that it is indeed the wanted behavior and that the recommendations that can be yield at the same time are complementary.

The aim of Sections 3.4.3, 3.4.4 and 3.4.5 will thus be to analyze how those anomalies are detected in the mentioned articles and to see what technique could best be adapted for detection of similar situations in energy rules. The conclusions will be given in Section 3.4.6.

3.4.3 Creation of anomaly free lists

Article [2] presents an algorithm that detects and resolves anomalies in a given rule list and generates a new list of rules that is anomaly free. That new list is built incrementally, considering each rule in the old rule list and inserting them in the new one in a way that doesn't introduces anomalies. The anomalies considered are shadowing, correlation and redundancy.

The algorithm maintains two lists of rules, *old_rules_list* for the original policy and *new_rules_list* for the new anomaly free list. The rules are represented as ordered tuple. The values of the fields are defined as ranges of values which are represented and analyzed as sets.

Each rule in *old_rules_list* is considered one after another and is inserted in *new_rules_list*. The rule r to be inserted is checked against each rule that has already been added to *new_rules_list*. Considering a pair of rule (r, s), the different possible cases are handled as follow:

- If r and s are equal, one of them is discarded.
- If r is a subset of s, r is inserted before s.
- If r and s are correlated, the rules are broken up into disjoint rules.

  In the three above cases, the loop on the rules in *new_rules_list* is broken and the algorithm moves on to the next rule in *old_rules_list*.
- If r is either disjoint or a superset for every rule s in *new_rules_list*, r is added at the end of the list.

Once all rules in *old_rules_list* have been considered, *new_rules_list* is scanned for redundancy anomalies. Any rule that is a subset of a subsequent rule with the same action is removed. Then a merging algorithm is applied to merge rules that have fields with consecutive ranges and the same action in order to reduce the number of rules in the list.

The resulting *new_rules_list* now contains a list of firewall rules that is free of shadowing, correlation and redundancy. A proof of correctness that those anomalies have been removed by the algorithm is included in the paper but is not of interest for the present discussion.

Article [34] expands [2] to handle non-ordered lists and to take generalization into account as a conflict to resolve. It thus considers all four anomalies presented. It also presents an algorithm to generate rules with non-overlapping attributes so they can be easily matched if they are inserted in a tree structure, which increases the firewall speed.

The conflict resolution algorithm considers a list of rules and examines it to detect anomalies and rewrite any two conflicting rules. For each possible situation, the action taken is described in the paper, but the precise functioning of the algorithm is not given.

3.4.4 Optimization using Directed Acyclic Graphs

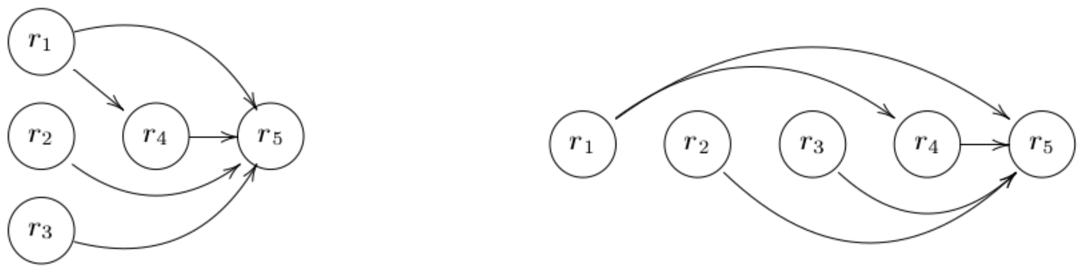
Articles [21] and [25] both address the optimization of firewall policies using Directed Acyclical Graphs (DAG). The strategy in [25] is to remove unnecessary rules, while [21] focuses on rule reordering and presents the DAG representation of a firewall policy used in both papers. While the reordering of the rules is not of relevance in the case of energy rules, the detection and removal of redundant rules could potentially be interesting and the DAG representation in itself even more so.

In the modeling described in [21], a rule is represented as an ordered tuple of sets, $r = (r[1], r[2], ...r[k])$ where each tuple $r[l]$ is a set that can be specified as a range. Two rules are defined as intersecting when the intersection of every corresponding tuple $r[l]$ is non-empty. A precedence relationship exists between two rules r_i and r_j if $i < j$ and the rules intersect.

Still in [21], a firewall policy is represented as a DAG, the rules and their precedence relationships are respectively represented as vertices and edges. The edges thus represent the constraints that the precedence relationships are. An example of such graph can be viewed in Figure 3.2a.

A linear arrangement is "a list of DAG vertices where all the successors of a vertex appear in sequence after that vertex" [21], guaranteeing that the precedence relationships are respected. A linear arrangement for the example in Figure 3.2a is given in Figure 3.2b. It has been proven in [20] that any linear arrangement of a policy DAG maintains its integrity, which refers to the absence of change in the original policy. By using a linear arrangement where rules that are more often matched are at the beginning of the list, the number of average comparison can be minimizes and the performance (speed) of the firewall improved. A sorting algorithm to solve this optimization problem is also presented in paper [21], but will not be discussed here.

Article [25] presents a rule optimization software, FIRO, that uses the same DAG structure and removes unnecessary rules without changing the firewall policy. It takes into consideration shadowing and redundancy anomalies, as well as another



(a) DAG policy

(b) Example of a possible linear arrangement

Figure 3.2: Two DAG representations for a 5 rules firewall policy, images from [21]

situation in which a rule can safely be removed.

In the first pass through the rule list, FIRO creates the corresponding DAG. It also remove redundant rules that correspond to one of the two following conditions:

- rule r is shadowed
- rule r is followed by another rule s such that:

$$\begin{cases} s \text{ and } r \text{ have the same action} \\ s \text{ matches at least all the packets matched by } r \\ \text{there is no rule between } r \text{ and } s \text{ that has a different action} \end{cases}$$

To verify the redundant rules for the conditions above, it is only necessary to consider rules that are related (intersected). The DAG representation is interesting in that regard, because it keeps track of intersections between rules, which allows to speed up the process. When a rule is removed, the references (edges representing precedence relationships) are transferred to another rule, if it doesn't already contain them.

After that rule removal step, a merging stage happens. When two rules r and s have the same values for all fields except one and the values for that field can be joint, those rules are merged into a new rule t. t inherits the references of r and s. The references to r and s rules are updated to point towards t. Once the rules have been merged, the procedure to remove the anomalies is started again because the merging may have created new situations in which a rule can be safely removed.

FIRO also encompasses for special cases, like log rules or limit options, but they are too specific to firewall to be transposed to energy rules.

3.4.5 Anomaly detection using matrices

Article [6] uses matrices to represent the list of filtering rules and the relationships between those rules, and to detect anomalies. The anomalies considered are redundancy, contradiction, generalization, shadowing and correlation. As stated in Section 3.4.2, the definitions given in this paper only take the relationships between fields into account and not the ordering of the rules in the list, which is interesting regarding an adaptation to energy rule sets.

First, the set of rules is represented by a matrix F where the n rows correspond to the rules and the m columns represent the fields. As stated, the relationship and anomalies between two rules are defined based on the relationships between their respective fields, which can be viewed as the difference between the values of two fields. So, for each field, an Inter-Difference Matrix (IDM) E is defined and contains the relationships/difference for that field for each pair of rules. Its size is thus $n \times n$ since the n rules are compared to all other rules. The relationships are represented using a predefined Inter-Difference Coding (IDC) that attributes a numerical value for each kind of relationship as defined in table 3.4.

Relationship	IDC code non-action field	IDC code action field
Shadowing	3	-
Generalisation	2	-
Equality	1	1
Difference	0	-1

Table 3.4: IDC for the different possible relationships

Those IDM can be used to build a model with superposed layers, where each one of them corresponds to the IDM for a specific field. We can define an Inter-Difference Vector (IDV) R_{ij} with the elements (i, j) of each IDM. R_{ij} then corresponds to the comparison between two rules R_i and R_j. Figure 3.3 shows a visual representation of the IDM layers model and R_{ij}.

Thanks to IDC and the IDM layers model, the anomalies can easily be formally defined using the IDC values found in IDV's. In their definitions, the authors use two vectors H_{xy} and R_{xy} such that $H_{xy}(i)$ and $R_{xy}(i)$ correspond to the IDC for the rules x and y and the i^{th} field, and $R_{xy}(Action)$ corresponds to the IDC of those rules for the action field. H_{xy} has length m which corresponds to the number of fields and R_{xy} has length $m + 1$. Following are two examples of such definitions

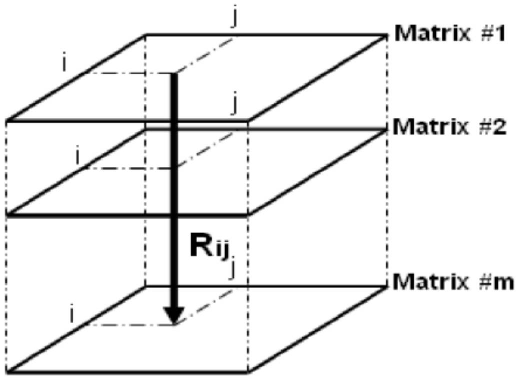


Figure 3.3: IDM layers model, image from [6]

for the absence of anomalies and for generalization, the other definitions being of similar form:

- Absence of anomalies: $\exists\ i \in \{1, ..., m\},\ H_{xy}(i) = 0.$
- Generalization Anomaly: $\begin{cases} R_{xy}(Action) = -1, \\ \exists \text{ at least } i \in \{1, ..., m\},\ H_{xy}(i) = 2 \\ \forall\ e \in (\{1, ..., m\} - i),\ H_{xy}(e) \in \{1, 2\}. \end{cases}$

With those definitions, the existence of anomalies can easily be proven by calculating the product p_{xy}^H of the elements of the vector H_{xy} or the product p_{xy}^R of the vector R_{xy}.

The paper gives the following theorem

An anomaly exists between the rules R_x and R_y if and only if $p_{xy}^H \neq 0$.

and its corollary

An anomaly exists between the rules R_x and R_y if and only if $p_{xy}^R \neq 0$.

The products p_{xy}^R can also be used to identify anomalies, which can be formulated with regard to this product, as in the two following examples :

- Absence of anomalies: $p_{xy}^R \prod_{i=1}^{m+1} R_{xy}(i) = 0.$
- Generalization Anomaly: $\begin{cases} R_{xy}(Action) = -1, \\ p_{xy}^R \prod_{i=1}^{m+1} R_{xy}(i) = -(1^{m-j} * 2^j), \end{cases}$
  where j is the number of generalized fields, and $m - j$ is the number of equal fields.

The $n \times n$ matrix of products, P, holds the products p_{xy}^R for each pair of rules, allowing to easily verify the existence or absence of an anomaly between two rules. It is also easy to identify the type of anomalies using modulo operations on the values in P. Table 3.5 shows how to determine the type of an anomaly based on p_{xy}^R.

Condition on $P[i,j]$	Anomaly
$P[i,j] = 0$	Absence of anomalies
$P[i,j] = -1$	Contradiction
$P[i,j] > 0$	Redundancy
$P[i,j] \bmod 6 = 0$	Correlation
$\begin{cases} P[i,j] \bmod 6 \neq 0 \\ P[i,j] \bmod 2 = 0 \end{cases}$	Generalization
$\begin{cases} P[i,j] \bmod 6 \neq 0 \\ P[i,j] \bmod 3 = 0 \end{cases}$	Shadowing

Table 3.5: Identification of anomaly between rules i and j using P matrix

The complexity, C, of the algorithm that builds all three types of matrices and detects the anomalies can be written as follow

$$C(m,n) = \frac{3}{2} mn^2$$

where m is the number of fields and n the number of rules.

The presented method allows to detect anomalies in a list of firewall rules, but also to store the relationships between them in matrices. If the rules are changed, for example to correct anomalies, the matrices can be updated with no need to repeated the whole process. It also allows for a numeric treatment of the anomalies which is more adaptive to fast algorithm than usual firewall rules representations.

3.4.6 Discussion and method choice

The comparison between firewall policies and energy rules and the analysis of different methods used on firewall rules show that they have many similarities between them. This makes it interesting to use one of those methods to analyze energy rules, provided some adaptations are made to take the differences into account. The most important ones are the facts that energy rule sets are not ordered, that there can be many different types of recommendations and that all the matching rules are considered in the AI result.

The method proposed in [2] and presented in Section 3.4.3 is highly based on the assumption that the order in which the rules are listed matters, which is true in the case of firewall but not for energy rules. It seems to be possible to build an algorithm inspired by the method in [2] for the case of energy rules. However, that would require to redefine the relationships of interest without using ordering and to make major changes to the algorithm. Further, since the algorithm takes a list and creates a new one without storing any information about relationships, the whole process would have to be executed each time a rule is modified or a new rule is added or deleted, which would be inefficient.

The second method [34] presented in that section takes generalization into account and claims to handle non-ordered list, which could potentially make it more adaptable to energy rule sets. Yet the management of non-ordered list is not detailed in the paper and the whole algorithm would also have to be repeated for each change to the rules.

In comparison, the use of DAG presented in Section 3.4.4 would be more interesting because it provides a way to store some information on the relationships between the rules. A detection process would thus not have to be restarted from scratch every time there is a modification to the rule. The edges still only contain information on the fact that two rules are related, but not how so. The exact type of relationship would thus have to be analyzed again for every change, which is not optimal. Still, the structure could be modified to hold more specific information and mitigate this limitation.

The algorithm in article [25] that is also adapted to DAG seemed somewhat interesting because it considers and fixed several type of redundant situations. However, it is also highly dependent of the order of the rules and would thus need to be almost completely rewritten to be adapted for energy rules.

The matrix representation of [6] presented in Section 3.4.5 has the advantage of storing precise information on the relationship between each pair of rules, including which type it is. In order to account for a change in the rule set, it thus suffices to update the corresponding matrices. The algorithm presented in this paper also only take into account the relationships between the values of respective fields in order to define the different type of anomalies and relationships. It is thus easier to adapt it to energy rules since it never relies on a specific ordering.

Given all these advantages and the downsides present in the other methods, I chose to adapt and use the algorithm in [6] to implement the solution developed for this thesis. The detail description of the solution using an adapted version of this algorithm is the subject of Chapter 4.

3.4.7 Refinement of tool functionalities

Another particularity of the chosen paper is that it gives a detailed explanation of the identification of each type of anomalies, while being more succinct on their corrections. The other articles reviewed, on the other hand, were more expansive on the correction of anomalies than on the method used to detect them.

So far, I have considered that an automated correction of certain relationships between the rules was desired. As I refined the design of the AI and made progress in my review of potential solutions, I kept on working with rules, including by creating various examples to help with my analysis. I also had several discussions with WeSmart throughout the whole process to better adjust my propositions to the company's needs. This allowed me to gain a better understanding of the rule creation process and the variety of situations that may be represented in the rules. It came to light that since all the rules were different with different reasoning behind them, the corrections potentially needed are very different from one case to another. It is thus very difficult to automate them.

Let's take the example of two rules that have the same recommendation but different conditions. Sometimes it is preferable to delete one of the rules. Other times it is better to keep both. In yet other cases it is best to create a different new rule to replace the two original ones. So there is no one solution that can be applicable to all pair of rules that have the same relationship. Also, knowing whether a particular situation is a mistake or was intended requires expert knowledge and may depend of the particular intention of the human expert who crafted the rules. It is thus not something that should be done automatically. Giving more freedom to the human expert rather than implementing automated corrections also corresponds to WeSmart's request.

Since the focus of the tool is now only on the identification and not the correction of the different relationships, the method in [6] is a very good choice regarding to that aspect too.

Chapter 4

Detection and identification of relationships between rules

As first expressed in Section 2.3, the objective here is to facilitate the creation of energy rules used by a recommendation AI through the creation of a tool that verifies the relationship between them. In order to accomplish the detection and identification of those relationships, I am going to adapt an algorithm used for the detection of anomalies in firewall policies, as explained in Section 3.4.6. The reference paper is [6] and is summarized in Section 3.4.5. This chapter contains the detailed solution I have developed in order to adapt that work to the case of energy rule sets.

Since the rules and their creation process are closely tied to the functioning of the recommendation AI, it is necessary to define the latter to some degree in order for the rules to be adapted to the future AI. As exposed in Chapter 3, the formulation of the rules and their use by the AI has evolved from my original idea and is also different from WeSmart's version. This chapter will thus start with the general outline of the AI.

4.1 Recommendation AI and its rules

The recommendation AI is an expert system with a knowledge base containing rules that allow to provide behavioral recommendations based on energy observations. It also stores user information so the recommendations given can be adapted not only on the consumption and production data taken as input, but also on each user's specificity.

The **observations** taken as input are energy consumption and production data as well as some potential contextual information like the time, the day or the weather. A **situation** is the combination of data found in an observation and other relevant information regarding the user specificity or preferences, for example their potential involvement in an energy community, whether they have solar panels, their usual schedule for being home or away,... A situation is expressed using **attributes**. They can be any element that can serve as a criteria to differentiate a situation from another, like energy data, relation between different data, time, weather characteristic, user specificity... Their values can be ranges of numbers, numbers (which can also be used to represent other types of predefined values) or boolean. They can be viewed as sets. Considering situation that can include user data on top of energy observations is an addition to WeSmart AI, which includes less adaptation to the different users in its rules.

The rules present in the knowledge base are used by the AI to associate situations to the appropriate recommendations. **Rules** are composed of a condition and a recommendation, which is similar to firewall rules [27].

$$\langle \textit{condition} \rangle \rightarrow \langle \textit{recommendation} \rangle$$

The **condition** is a set of attributes and their associated values that define a specific set of situations. It can be seen as a boolean expression that is $True$ when the situation considered is the set described by the condition and $False$ otherwise. When a situation s satisfies the condition of a rule r, s and r are said to **match**.

The **recommendations** are pieces of advice that can be given to a user in the specific situations described by the rule condition. Each recommendation should contain a single piece of advise. It is possible that several of them can fit the same situation (ex: "Reduce your consumption" and "Add new energy production units" are both possible recommendations to a situation in which the user consume more energy that they produce.) In this case each recommendation need to be expressed in its own rule.

An example of a rule set in textual form and its equivalent in formal form can be respectively found in Tables 4.1, and 4.2. The symbol '$*$' is used as a wildcard for unspecified values, like in [21], and can be interpreted as the set of all the possible values for that attribute.

When the recommendation AI takes an observation as input, a preliminary step needs to be executed to pre-process the data given as the input observation to compute the corresponding situation. This includes calculating the value for certain attributes where necessary (e.g. attributes that correspond to a relation between different data) and to add the user information.

The inference procedure of the AI is very straightforward. The situation considered is checked against each of the rules in the knowledge base to identify all the matching ones. The recommendations associated to the matching rules are then aggregated and given as output. The complexity of this algorithm is thus directly proportionate to the number of rules in the rule set. $C = \Theta(n)$ Given that this number should never be very high and that the execution time is not crucial in this procedure, the number of rules in the rule set should not become a concerned.

Condition	$\rightarrow$	**Recommendation**
Yearly offtake is between 350 and 700 kWh, there is no time of day during which production is much higher than consumption, and user has not reached maximum possibility for production units	$\rightarrow$	Add photovoltaic panel
Yearly offtake is between 700 and 1200 kWh, there is no time of day during which production is much higher than consumption, and user has not reached maximum possibility for production units	$\rightarrow$	Add 2 photovoltaic panels
Yearly offtake is over 1200kWh, there is no time of day during which production is much higher than consumption, and user has not reached maximum possibility for production units	$\rightarrow$	Add 3 or more photovoltaic panels
Afternoons production is higher than consumption and morning consumption is higher than production	$\rightarrow$	Run devices with high consumption during afternoon if possible
Afternoons production is higher than consumption, morning consumption is higher than production, and user is not in energy community	$\rightarrow$	Join energy community
User has not reached maximum possibility for production units and is in energy community	$\rightarrow$	Add maximal number of photovoltaic panels
Consumption at night is 40% of daily consumption	$\rightarrow$	Reduce use of energy at night

Table 4.1: Example of rule set in textual form

Recommendation	YearlyOfftake	OfftakeMorning	OfftakeAfternoon	RatioNightDayCons	MaxProd	InCommunity
Add 1 photovoltaic panel	$[350, 700[$	$]-2, inf]$	$]-2, inf]$	$*$	False	$*$
Add 2 photovoltaic panels	$[700, 1200[$	$]-2, inf]$	$]-2, inf]$	$*$	False	$*$
Add 3 or more photovoltaic panels	$[1200, inf]$	$]-2, inf]$	$]-2, inf]$	$*$	False	$*$
Run highly consuming devices during afternoon	$*$	$]0, inf]$	$[-inf, 0[$	$*$	$*$	$*$
Join energy community	$*$	$]0, inf]$	$[-inf, 0[$	$*$	$*$	$*$
Add maximal number of solar panels	$*$	$*$	$*$	$*$	False	True
Reduce use of energy at night	$*$	$*$	$*$	0.4	$*$	$*$

Table 4.2: Example of rules in their formal form
YearlyOfftake: Total offtake (energy bought from the grid) for the year in kWh
OfftakeMorning: Average offtake for the morning period in kWh
OfftakeAfternoon: Average offtake for the afternoon period in kWh
RatioNightDayCons: Ratio of the average total consumption during the night and the day
MaxProd: Whether the user has reached its maximal production potential
(e.g. True when roof is field with solar panels and no other production possibility)
InCommunity: Whether the user takes part in an energy community

4.2 Detection and identification of relationships

Since the rules are at the heart of the AI, the quality of the rule set is crucial in order to return relevant recommendations, as explained in Chapter 2. In particular, it is very important that the set of recommendations given simultaneously for the same situation are complementary. The human experts designing the rule set thus need to be easily able to notice the cases in which redundant or contradictory recommendations could be given in order to fix the problem. (see Chapter 3) We will thus see in this section how to automatically detect the cases that require extra verification.

4.2.1 Relationships definitions

Those cases that require extra attentions are the ones in which two rules could be applied to the same situation. In order to detect those cases, we need to look at the relationships between each pair of rules. The relationships between two rules are considered regarding the relationships between the set of situations that can be matched by each rules. This is similar to the way firewall rules are considered regarding the set of packets they match [2].

Two rules can either be **disconnected**, if there can be no situation they both match, or **connected**, if there exist at least one possible situation they can both match. The different relationships between two rules are listed bellow :

Disjunction : Two rules r and s are disjoint if the set of situations that are matched by both rules is empty, as show in Figure 4.1a. The values of at least one of their respective attributes are disjoint. The words *disjoint* and *disconnected* can be used interchangeably.

Equality : Two rules r and s are equal if all the rules matched by r are also matched by s and all the rules matched by s are also matched by r, as show in Figure 4.1b. The values of all of their respective attributes are equal.

Inclusion : A rule r is included in a rule s if all the situations matched by r are also matched by s and s also matches situations that are not matched by r, as show in Figure 4.1c. For all respective attributes r's values are either a subset of or equal to s's values, with at least one attribute for which it is a subset.

(a) Disjunction Disconnection (b) Equality Connection (c) Inclusion Connection (d) Overlap Connection

Figure 4.1: Representation of the sets of matched situations for pairs of rules with different relationships

Overlap : Two rules r and s overlap if there can exist at least one situation that is matched by r and not by s, at least one situation that is matched by s and not by r and at least one situation that is matched by both r and s, as show in Figure 4.1d. For all respective attributes, the values of r and the values of s can't be disjoint and at least one of the two following sufficient conditions must hold:

- There is at least one attribute for which the values of r are a subset of the values of s and at least one attribute for which the values of s are a subset of the values of r.
- There is at least one attribute for which the values of r overlap the values of s, meaning the intersection between the two set of values is not empty, not equal to the set of values of r and not equal to the set of values of s.

Equality, inclusion and overlap are different types of connection, while disjunction is the only type of disconnection. The definitions of disjunction, equality and inclusion have been partially inspired by the firewall rules definition in [2], [6]. Unlike in firewalls, the attribute values of energy rules can be boolean or single values (as opposed to ranges of values). In that case, the values for two rules can just be either equal or different, and one can't be the subset of the other. This doesn't have an impact on the relationships definitions.

Relationships can be specified further by indicating if the recommendations are equal or different for both rules. I will not define anomalies to take the conclusion of the rule into account like is often done for the detection of firewall rules. Doing so would create redundant definitions in the case of energy rules. The word *anomaly* would also be misleading because connections between rules can be necessary in order to associate several complementary rules to the same set of situations. It is the role of the human expert to differentiate intended connections from mistakes.

4.2.2 Identification using matrix representation

Rule set and relationships between attribute values

The set of rules can be represented in a matrix S where the rows represent the n rules and the columns correspond to $m+1$ fields, which include the m attributes as well as the recommendation, which is located in the first field. The element v_{ij} thus represents the value of the j^{th} field for the i^{th} rule. The value `NaN` (not a number), the wildcard '$*$' or an empty entry are used for unspecified values. They don't represent a constraint on the situation and should be interpreted as the set of all possible values for that attribute when compared to other values. Thus two unspecified values are equal and a specified value is always a subset of an unspecified value.

Since the relationships between rules can be defined with regard to the relationships between their respective fields, the latter are represented in Inter-Difference Matrices (IDM). For each attribute, there is a corresponding IDM R that represents the relationships between the values of this attribute for each pair of rules. There is also one IDM for the recommendation, which is an addition to the original model in [6]. For a rule set of n rules with m attributes, there is thus $m+1$ IDM's of size n x n. Since the relationship between attribute values of rules R_i and R_j are reciprocal and the diagonal would correspond to a relationship between values of the same rule, R is a strictly upper triangular matrix[1]. The information for a relationships between R_i and R_j is thus found in the entry (i, j) if $i < j$ and in the entry (j, i) if $j < i$. For simplicity, it will always be considered that $i < j$ for the remaining of Section 4.2.

Together, the IDM matrices can be considered as several layers of a 3D tensor, creating an IDM layers model. The 0^{th} layer is associated to the recommendation. The other layers, 1^{st} to m^{th}, correspond to each of the attributes. Like for firewall policies, we can define an Inter-Difference Vector (IDV) R_{ij} that represents the relationship between the rules R_i and R_j, where $i < j$. Its elements are the elements (i, j) of each IDM, which includes the recommendation IDM in the case of energy rules, so its length is $(m+1)$. A visual representation of the IDM layer model and an example of vector R_{ij} can be seen in Figure 4.2.

In order to represent the relationships between attribute values in a way that will facilitate the detection of relationships between rules, they are encoded in the IDM's using a specific Inter-Difference Coding (IDC). The IDC associates a different number to each type of relationship. The coding used for energy rules is

[1]In a strictly upper triangular matrix, only the entries above the diagonal can be non-zero. The diagonal and all the entries below it are zero.

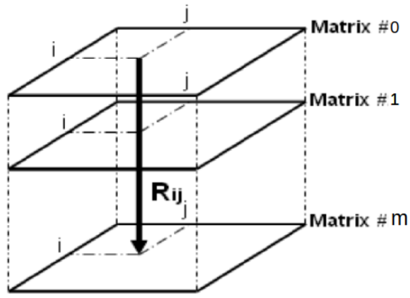


Figure 4.2: IDM layers model for energy rule set, modification of image from [6]

given in table 4.3 alongside the definitions of the different possible relationships between attribute values. $\Re(v_{ik}, v_{jk})$ corresponds to the relationship between R_i and R_j for the k^{th} attribute, which is represented in the k^{th} layer of the IDM layers model.

$\Re(v_{ik}, v_{jk})$	Definition	IDC code $1 \leq k \leq m$	IDC code $k = 0$
Difference	$val(v_{ik}) \cap val(v_{jk}) = \phi$	0	-1
Equality	$val(v_{ik}) = val(v_{jk})$	1	1
Inclusion ij	$val(v_{ik}) \subset val(v_{jk})$	2	-
Inclusion ji	$val(v_{ik}) \supset val(v_{jk})$	3	-
Overlap	$\begin{cases} val(v_{ik}) \cap val(v_{jk}) \neq \emptyset \\ val(v_{ik}) \not\subset val(v_{jk}) \\ val(v_{ik}) \not\supset val(v_{jk}) \end{cases}$	6	-

Table 4.3: Relationships between attribute values of two rules: Definitions and IDC

Relationships between rules with regard to IDC

The IDC above (Table 4.3) has been created such that it is easy to deduce the type of a relationship between rules using R_{ij}. The definitions of those relationships with regard to R_{ij} are as follows[2]:

- **Disjunction :** $\exists\ x \in \{1, ..., m\},\ R_{ij}(x) = 0$
- **Equality :** $\forall\ x \in \{1, ..., m\},\ R_{ij}(x) = 1$

[2]This is similar, yet more straightforward, that what has been done in [6].

- **Inclusion of R_i in R_j :** $\begin{cases} \exists \text{ at least } x \in \{1, ..., m\},\ R_{ij}(x) = 2 \\ \forall\ e \in (\{1, ..., m\} - x),\ R_{ij}(e) \in \{1, 2\} \end{cases}$

- **Inclusion of R_j in R_i :** $\begin{cases} \exists \text{ at least } x \in \{1, ..., m\},\ R_{ij}(x) = 3 \\ \forall\ e \in (\{1, ..., m\} - x),\ R_{ij}(e) \in \{1, 3\} \end{cases}$

- **Overlap :** *Either one of these conditions needs to be satisfied:*

$$\begin{cases} \exists \text{ at least } x \in \{1, ..., m\},\ R_{ij}(x) = 6 \\ \forall\ e \in (\{1, ..., m\} - x),\ R_{ij}(e) \in \{1, 2, 3, 6\} \end{cases}$$

or

$$\begin{cases} \exists \text{ at least } x \in \{1, ..., m\},\ R_{ij}(x) = 2 \\ \exists \text{ at least } y \in \{1, ..., m\},\ R_{ij}(y) = 3 \\ \forall\ e \in (\{1, ..., m\} - (x, y)),\ R_{ij}(e) \in \{1, 2, 3, 6\} \end{cases}$$

All the connections above can be further specified to determine whether the rules have the same recommendation or not, with the following definitions:

- **Connection with same recommendation :** $R_{ij}(0) = 1$

- **Connection with different recommendations :** $R_{ij}(0) = -1$

With those definitions, we can prove the existence of a connection between the rules R_i and R_j, with $i < j$, using the product of the elements of the IDV R_{ij} of the rules. Let us define that product as follow:

$$p_{ij} = \prod_{x=0}^{m} R_{ij}(x)$$

Theorem 1 : *A connection exists between rules R_i and R_j, with $i < j$, if and only if $p_{ij} \neq 0$.*

Proof: According to the definitions of Section 4.2.1, a connection exists when the rules are not disconnected, which corresponds to them not being disjoint. We have established that, in the case of a disjunction between two rules R_i and R_j where $i < j$, we have

$$\exists\ x \in \{1, ..., m\},\ R_{ij}(x) = 0.$$

So, in the case of a connection,

$$\forall\ x \in \{1, ..., m\},\ R_{ij}(x) \neq 0.$$

In turn,

$$\prod_{x=0}^{m} R_{ij}(x) \neq 0 \text{ and } p_{ij} \neq 0.$$

This theorem and its proof follow the same reasoning than in [6].

Relationships between rules with regard to p_{ij}

The product p_{ij} can be used to detect the existence of a connection, but, thanks to the way the IDC is built, it also allows to identify its type, as well as whether the recommendations of the two rules are the same or not. From the definitions in the previous subsection, we can express the relationships as follow:

- **Disjunction :**
$$p_{ij} = \prod_{x=1}^{m} R_{ij}(x) = \prod_{x=0}^{m} R_{ij}(x) = 0$$

- **Equality with same recommendation :**
$$p_{ij} = \prod_{x=0}^{m} R_{ij}(x) = 1^{(m+1)}$$

- **Equality with different recommendations :**
$$p_{ij} = \prod_{x=0}^{m} R_{ij}(x) = -1 \times 1^{m}$$

- **Inclusion of R_i in R_j with same recommendation:**
$$p_{ij} = \prod_{x=0}^{m} R_{ij}(x) = 1 \times (1^{(m-x)} \times 2^{x})$$

- **Inclusion of R_i in R_j with different recommendations :**
$$p_{ij} = \prod_{x=0}^{m} R_{ij}(x) = -1 \times (1^{(m-x)} \times 2^{x})$$

- **Inclusion of R_j in R_i with same recommendation :**
$$p_{ij} = \prod_{x=0}^{m} R_{ij}(x) = 1 \times (1^{(m-y)} \times 3^{y})$$

- **Inclusion of R_j in R_i with different recommendations :**

$$p_{ij} = \prod_{x=0}^{m} R_{ij}(x) = -1 \times (1^{(m-y)} \times 3^y)$$

- **Overlap with same recommendation :**

$$p_{ij} = \prod_{x=0}^{m} R_{ij}(x) = 1 \times (1^{(m-x-y-z)} \times 2^x \times 3^y \times 6^z)$$

- **Overlap with different recommendations :**

$$p_{ij} = \prod_{x=0}^{m} R_{ij}(x) = -1 \times (1^{(m-x-y-z)} \times 2^x \times 3^y \times 6^z)$$

  where, in the above equations, m is the total number of attributes, x is the number of attributes with an inclusion ij between the values of R_i and R_j, y is the number of attributes with an inclusion ji and z is the number of attributes with an overlap.

All the products p_{ij} of a rule set are held in the Product Matrix P that is generated from the $(m+1)$ IDM. Like the IDM's, P is a strictly upper triangular matrix. This matrix P allows to easily detect the type of a relationship between two rules using modulo operations on its elements since, for every $i < j$, $P[i, j] = p_{ij}$. The operations that permit to deduce the type of relationship from P are listed in Table 4.4.

Discussion on matrix approach

The method presented in this section thus allows to determine the type of a relationship between two rules using matrix representations, IDC and mathematical properties. The creation of an IDM has a quadratic complexity with regard to the number of rules. $C_{IDM} = \mathcal{O}(n)^2$ The complexity to compute the product matrix may depend of the algorithm used, but the naive approach is also quadratic.

The matrix representation described also provides an efficient way to store all the different relationships of interest, which make potential updates easy. If a rule is modified, added or deleted, the different matrices can be updated with no need to start the whole process from scratch. This is especially interesting for a relationship identification tool that is used for the crafting of rules because this process will likely include several modifications and corrections of the rules before the final rule set is created.

Condition on $P[i,j]$	Relationship	Recommendations
$P[i,j] = 0$	Disjunction	irrelevant
$P[i,j] = 1$	Equality	same
$P[i,j] = -1$	Equality	different
$\begin{cases} P[i,j] \bmod 6 = 0 \\ P[i,j] > 0 \end{cases}$	Overlap	same
$\begin{cases} P[i,j] \bmod 6 = 0 \\ P[i,j] < 0 \end{cases}$	Overlap	different
$\begin{cases} P[i,j] \bmod 6 \neq 0 \\ P[i,j] \bmod 2 = 0 \\ P[i,j] > 0 \end{cases}$	Inclusion of R_i in R_j	same
$\begin{cases} P[i,j] \bmod 6 \neq 0 \\ P[i,j] \bmod 2 = 0 \\ P[i,j] < 0 \end{cases}$	Inclusion of R_i in R_j	different
$\begin{cases} P[i,j] \bmod 6 \neq 0 \\ P[i,j] \bmod 3 = 0 \\ P[i,j] > 0 \end{cases}$	Inclusion of R_j in R_i	same
$\begin{cases} P[i,j] \bmod 6 \neq 0 \\ P[i,j] \bmod 3 = 0 \\ P[i,j] < 0 \end{cases}$	Inclusion of R_j in R_i	different

Table 4.4: Identification of relationships between R_i and R_j using P matrix

Adaptations from reference paper

As expressed before, the method presented in this section 4.2 is an adaptation of the method used in [6] for firewall rules. More precisely, most concepts are coming from that paper, namely the matrix representation of the rules, IDM's, the IDM layers model, IDV, the P matrix, the use of an IDC and of the products of the IDV elements in order to detect and identify certain situations (relationships or anomalies). The different matrices (S, R and P) and vectors (R_{ij}) have however been adapted in order to include the recommendation. This makes it easier to define and identify the relationships between rules using only R_{ij} and avoids the need for complementary operations to determine whether two recommendations are identical. IDM and P matrix have also been modified to be strictly upper triangular, which is not the case in [6].

The different methods to define and express the relationships are similar to the successive definitions of relationships and anomalies in [6]. However, they have been lightly adapted to be more straightforward and easier to understand than in the original paper. This was possible because the recommendation have been included in the matrices alongside the other attributes. On the other hand, the reference paper uses different types of vectors and products and requires extra verification on the relationships between recommendations.

The different types of relationships considered and their definitions have been modified to fit the purpose of energy rule set, leading to a new detection and identification model.

4.3 Example of relationship identification

Let us now look at an example of rule set in order to see how the method described in Section 4.2.2 is applied to it. An example should also help to better understand how the detection and identification of connections can help human experts in the crafting of the rules.

For this example, let us consider the rules given in Table 4.5. The matrix representation of this rule set, S, is given in Figure 4.3. In order to improve readability, the recommendations are represented with key words in that matrix.

The IDM's constructed from S and corresponding to each fields are found in Figure 4.5. We can see the IDC for the relationship between the attributes of each pair of rules. As example, we can notice that all the relationships between recommendation values are different (IDC = -1), except for the the 6^{th} and 7^{th} rules (starting counting at 0). We can see that the recommendation is indeed "Join energy community" in both cases and that the IDC at index $(6, 7)$ is 1 in the $0^t h$ IDM. This analyses can of course be done for each IDM.

In Figure 4.4, we can see the products p_{ij} for each pair of rules R_i and R_j and deduce their relationships from it, as explained in the previous section and using the operations in Table 4.4.

Having this indication on the relationships, it is easier to see which relationships should be modified in order to have a coherent ruleset.

Recommendation	YearlyOfftake	OfftakeMorning	OfftakeAfternoon	MaxProd	InCommunity
Add 1 photovoltaic panel	$[350, inf]$	$]-2, inf]$	$]-2, inf]$	False	$*$
Add 2 photovoltaic panels	$[700, inf]$	$]-2, inf]$	$]-2, inf]$	False	$*$
Add 3 or more photovoltaic panels	$[1200, inf]$	$]-2, inf]$	$]-2, inf]$	False	$*$
Run highly consuming devices during afternoon	$*$	$[0, inf]$	$[-inf, 0]$	$*$	$*$
Run highly consuming devices during morning	$*$	$[-inf, 0]$	$[0, inf]$	$*$	$*$
Add solar panels	$[350, inf]$	$*$	$*$	False	$*$
Join energy community	$*$	$]0, inf]$	$[-inf, 0[$	$*$	False
Join energy community	$*$	$[-inf, 0[$	$]0, inf]$	$*$	False
Add maximal number of photovoltaic panels	$*$	$*$	$*$	False	True

Table 4.5: Set of rules considered for the example
YearlyOfftake: Total offtake (energy bought from the grid) for the year in kWh
OfftakeMorning: Average offtake for the morning period in kWh
OfftakeAfternoon: Average offtake for the afternoon period in kWh
RatioNightDayCons: Ratio of the average total consumption during the night and the day

$$\begin{pmatrix} 1\_panel & \text{[350,inf]} & \text{]-2,inf]} & \text{]-2,inf]} & False & * \\ 2\_panels & \text{[700,inf]} & \text{]-2,inf]} & \text{]-2,inf]} & False & * \\ 3\_panels & \text{[1200,inf]} & \text{]-2,inf]} & \text{]-2,inf]} & False & * \\ Run\_in\_after & * & \text{[0,inf]} & \text{[-inf,0]} & * & * \\ Run\_in\_morn & * & \text{[-inf,0]} & \text{[0,inf]} & * & * \\ Sol\_panels & \text{[350,inf]} & * & * & False & * \\ Join\_commu & * & \text{]0,inf]} & \text{[-inf,0[} & * & False \\ Join\_commu & * & \text{[-inf,0[} & \text{]0,inf]} & * & False \\ max\_panels & * & * & * & False & True \end{pmatrix}$$

Figure 4.3: Rule set matrix S

$$\begin{pmatrix} 0. & -3. & -3. & -72. & -72. & -4. & -216. & -216. & -24. \\ 0. & 0. & -3. & -72. & -72. & -8. & -216. & -216. & -24. \\ 0. & 0. & 0. & -72. & -72. & -8. & -216. & -216. & -24. \\ 0. & 0. & 0. & 0. & -36. & -36. & -27. & -0. & -36. \\ 0. & 0. & 0. & 0. & 0. & -36. & -0. & -27. & -36. \\ 0. & 0. & 0. & 0. & 0. & 0. & -108. & -108. & -6. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix}$$

Figure 4.4: Representation of PM for the given example

$$\begin{pmatrix} 0. & -1. & -1. & -1. & -1. & -1. & -1. & -1. & -1. \\ 0. & 0. & -1. & -1. & -1. & -1. & -1. & -1. & -1. \\ 0. & 0. & 0. & -1. & -1. & -1. & -1. & -1. & -1. \\ 0. & 0. & 0. & 0. & -1. & -1. & -1. & -1. & -1. \\ 0. & 0. & 0. & 0. & 0. & -1. & -1. & -1. & -1. \\ 0. & 0. & 0. & 0. & 0. & 0. & -1. & -1. & -1. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & -1. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -1. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix}$$

(a) 0^{th} field: Recommendation

$$\begin{pmatrix} 0. & 3. & 3. & 2. & 2. & 1. & 2. & 2. & 2. \\ 0. & 0. & 3. & 2. & 2. & 2. & 2. & 2. & 2 \\ 0. & 0. & 0. & 2. & 2. & 2. & 2. & 2. & 2. \\ 0. & 0. & 0. & 0. & 1. & 3. & 1. & 1. & 1. \\ 0. & 0. & 0. & 0. & 0. & 3. & 1. & 1. & 1. \\ 0. & 0. & 0. & 0. & 0. & 0. & 2. & 2. & 2. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 1. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix}$$

(b) 1^{th} attribute: YearlyOfftake

$$\begin{pmatrix} 0. & 1. & 1. & 3. & 6. & 2. & 3. & 6. & 2. \\ 0. & 0. & 1. & 3. & 6. & 2. & 3. & 6. & 2. \\ 0. & 0. & 0. & 3. & 6. & 2. & 3. & 6. & 2. \\ 0. & 0. & 0. & 0. & 6. & 2. & 3. & 0. & 2. \\ 0. & 0. & 0. & 0. & 0. & 2. & 0. & 3. & 2. \\ 0. & 0. & 0. & 0. & 0. & 0. & 3. & 3. & 1. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 2. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 2. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix}$$

(c) 2^{th} attribute: OfftakeMorning

$$\begin{pmatrix} 0. & 1. & 1. & 6. & 3. & 2. & 6. & 3. & 2. \\ 0. & 0. & 1. & 6. & 3. & 2. & 6. & 3. & 2. \\ 0. & 0. & 0. & 6. & 3. & 2. & 6. & 3. & 2. \\ 0. & 0. & 0. & 0. & 6. & 2. & 3. & 0. & 2. \\ 0. & 0. & 0. & 0. & 0. & 2. & 0. & 3. & 2. \\ 0. & 0. & 0. & 0. & 0. & 0. & 3. & 3. & 1. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 2. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 2. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix}$$

(d) 3^{th} attribute: OfftakeAfternoon

$$\begin{pmatrix} 0. & 1. & 1. & 2. & 2. & 1. & 2. & 2. & 1. \\ 0. & 0. & 1. & 2. & 2. & 1. & 2. & 2. & 1. \\ 0. & 0. & 0. & 2. & 2. & 1. & 2. & 2. & 1. \\ 0. & 0. & 0. & 0. & 1. & 3. & 1. & 1. & 3. \\ 0. & 0. & 0. & 0. & 0. & 3. & 1. & 1. & 3. \\ 0. & 0. & 0. & 0. & 0. & 0. & 2. & 2. & 1. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 3. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 3. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix}$$

(e) 4^{th} attribute: MaxProd

$$\begin{pmatrix} 0. & 1. & 1. & 1. & 1. & 1. & 3. & 3. & 3. \\ 0. & 0. & 1. & 1. & 1. & 1. & 3. & 3. & 3. \\ 0. & 0. & 0. & 1. & 1. & 1. & 3. & 3. & 3. \\ 0. & 0. & 0. & 0. & 1. & 1. & 3. & 3. & 3. \\ 0. & 0. & 0. & 0. & 0. & 1. & 3. & 3. & 3. \\ 0. & 0. & 0. & 0. & 0. & 0. & 3. & 3. & 3. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{pmatrix}$$

(f) 5^{th} attribute: InCommunity

Figure 4.5: Representation of all the different IDM's for the given example

When analyzing the P matrix, we can see for example that $P[0,1] = p_{0,1} = -3$, so $p_{0,1} < 0$ and $p_{0,1} \bmod 3 = 0$. With the conditions given in Table 4.4, we know that the relationship between R_0 and R_1 is an inclusion. The relationships between R_0 and R_2 and between R_1 and R_2 are also inclusions, since $P[0,2] = p_{0,2} = -3$ and $P[1,2] = p_{1,2} = -3$. This means that there are situations in which R_0, R_1 and R_2 will be return together. If we look at those rule recommendations, we see that they shouldn't be given together as output since they contain redundant information. It is thus necessary to fix this situation. In this case, the good way to do it is to change their values for the attribute *YearlyOfftake* and make them disjoint. With an identification tool that highlight the inclusion relationship, the problem would be easy to see and can thus be corrected.

Other problems can be found in this matrix where redundant or contradictory recommendation could be given simultaneously. A possible way to fix all of them is presented in Figure 4.6. For the reader who wants to explore this example more, it can be found, as well as its corrected version and other examples, in CSV format at [26] alongside the developed tool that will be presented in Chapter 5.

Recommendation	YearlyOfftake	OfftakeMorning	OfftakeAfternoon	MaxProd	InCommunity
Add 1 photovoltaic panel	[350,700[	]-2,inf]	]-2,inf]	FALSE	FALSE
Add 2 photovoltaic panels	[700, 1200[	]-2,inf]	]-2,inf]	FALSE	FALSE
Add 3 or more photovoltaic panels	[1200,inf]	]-2,inf]	]-2,inf]	FALSE	FALSE
Run highly consuming devices during afternoon		]0,inf]	[-inf,0[		
Run highly consuming devices during morning		[-inf,0[	]0,inf]		
Join energy community		]0,inf]	[-inf,0[		FALSE
Join energy community		[-inf,0[	]0,inf]		FALSE
Add max number of photovoltaic panels				FALSE	TRUE

Figure 4.6: Possible correction of the example rule set

Chapter 5

Relationship Identification Tool

The source code of the prototype to implemented to detect and identify connections between rules, the Relationship Identification Tool (RIT), can be found in the following repository [26]:
`https://github.com/Magalii/Relationship-Identification-Tool`

Figure 5.1: Relationship Identification Tool (RIT)

It allows the user, supposedly an energy expert, to manage a rule set trough a graphical user interface (GUI), as can be seen in Figure 5.1. The main feature of RIT are the highlighting and indication of the different types of connections between rules and the possibility to modify and update the rule set using the tool. The user can thus successively analyze the rules and their relationships then modify them until satisfaction. The rule set can be loaded from and saved to a CSV file, as well as being started from scratch in the tool.

The implementation includes a `RuleSet` class that can be used in another program and allows to create objects to store and manage rule sets. It provides an interface for accessing and modifying the data in the rule set, as well as getting the type of the relationship between any pair of rules it contains.

5.1 Tool functionalities

5.1.1 Rule set display and creation

The GUI of the tool displays each rule composing the rule set in a grid that corresponds to its matrix form, with the attribute names indicated above. An example for the rule set already used as example in Section 4.3 is shown in Figure 5.2.

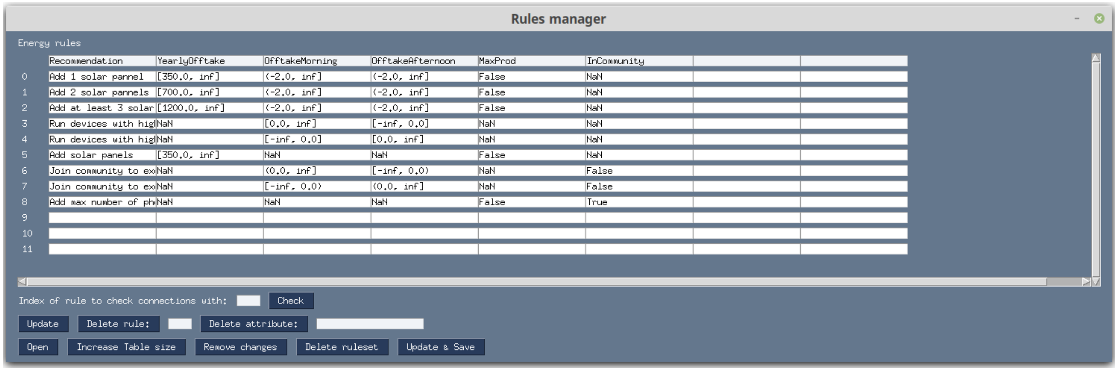


Figure 5.2: RIT GUI displaying rule set

The rule set can be loaded from a CSV file, using the Open button, provided it respects the appropriate format. The first row must contain the attribute names, starting with the recommendation named 'Rec' or 'Recommendation'. The other rows represent the rules. The format of a rule set and its values is described in Section 4.1. More details on the way the values should be represented in the CSV file and the input boxes of the GUI can be found at [26]. A new rule set can also be created in RIT by filling the GUI grid with the wanted attributes, recommendations and values. The same format needs to be used. That new rule set can latter be saved to a new or existing file using the Update & Save button.

5.1.2 Modification of rule set

The value of an existing rule and attribute can be modified by replacing the old value by the new one in the corresponding cell. This also works for attribute names and recommendations. A new rule can be added by writing its recommendation and attribute values in the corresponding cells of the first empty row. A new attribute can be added by indicating its name in the first empty attribute cell. If there is not enough empty space available to add the desired number of rules or attributes, the user can click the Increase Table size button which will refresh the GUI and display a new grid with more rows and columns.

Once all the desired changes have been written in, the button Update must be pressed in order for the changes to be actually applied to the rule set. If all the inputs have an appropriate format, all the modifications made are committed. If there is one or more input that doesn't respect the required format, each error is displayed to the user in a pop up window and no changes are saved. The unsaved modifications are still displayed to allow the user to fix them before trying to save them again. It is also possible for the user to remove all unsaved modifications using the Remove changes button.

The deletion of a rule is done by indicating its number in the input box to the right of the Delete rule button and clicking on it. The deletion of an attribute is similar with the Delete attribute button, except that either the name or the number of the attribute can be given as input. Those changes are directly applied and don't require to use the Update button. It is also possible to delete the complete rule set using the button Delete rule set.

The updates saved are local to the tool and RIT doesn't store information after the program is closed. In order to save them for for further access, it is necessary to store them in a file, using the Update & Save button.

5.1.3 Identification of relationships

RIT allows the user to select a specific rule in order to identify which relationships the other rules have with the selected one. The user needs to input the number of the rule they want to consider as the reference in the input box preceded by the text "*Index of rule to check connections with*" and click the Check button. The GUI then indicates next to each rule the name of the relationship it has with the reference rule. In the case of a connection, the rule is highlighted in color, with each color corresponding to a specific type of connection. The color code is indicated in Table 5.1. A dummy example of relationship identification in which all the different types of relationships and their respective colors can be seen is shown in Figure 5.3.

	Equality	Inclusion	Overlap
Same rec.	Dark green	Medium green	Light green
Different rec.	Dark blue	Medium blue	Light blue

Table 5.1: Color coding for highlight of connections in RIT

Figure 5.3: Identification of relationships in which all the possible types are showcased

5.2 Implementation

The program was implemented using Python 3.7 to take advantage of its different modules for matrices computation and data management. It is also the language most used by WeSmart so the code can be easily used in their existing implementation.

RIT is launched using the script `gui` which builds the GUI and handles the interactions with the user. This script calls the class `RuleSet` to instantiate it and create the `ruleset` object that will be used to hold and manage the rules. It will call its methods in reaction to the user inputs that requires to get information from or make changes to the `ruleset` object. `gui` uses the module `Parser` to convert user inputs and CSV file content into the proper values usable by `gui` and `RuleSet`. The script also uses the enumeration `Connection` in order to associate each enum type with a specific color (according to the color code explained in Section 5.1.3).

The class `RuleSet` contains all the attributes and methods necessary to manage the rule set, including creating it, getting information about it, modify it or saving it. The class uses the enumerations `Connection` and `Relation` to represent the different types of relationships at different stages of the detection and identification process. `Relation` enumerates the different values taken in consideration in the

IDC presented in Section 4.2.2. `Connection`, on the other hand, enumerates the different relationships as defined in Section 4.2.1, as well as the special case of the reference rule and an error value.

The class diagram of the code is presented in Figure 5.4. It follows UML convention [23] since that notation is widely known. It is thus language independent, which is why `Parser` is represented as a Utility class, a class of static attribute and methods that has no instance, even though it is written as a module in this Python implementation. Some minor changes to UML standards have been made in order to avoid redundancy and increase the readability of the diagram. In the case in which a function can accept arguments of different types or may return variables of different types, the possible types are indicated next to each other and separated by a slash. When a specific argument can be of any type, no specification of type is indicated on the diagram.

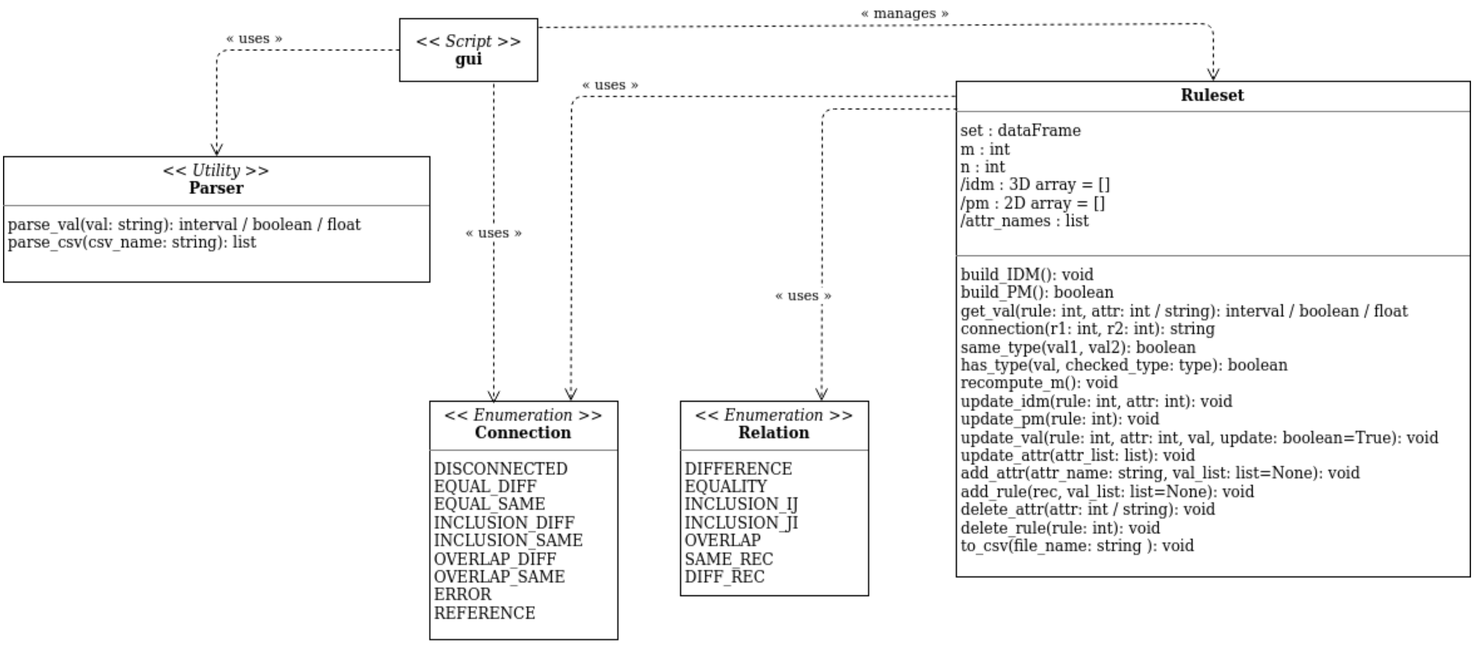


Figure 5.4: Structure diagram of RIT implementation

Conclusion

This thesis presents the design of a rule-based expert system that uses specific energy observation to provide energy recommendation to its users. The main focus of the work was to develop a method to help human experts in the creation of the knowledge base composed of rules of this recommendation AI. To reach this objective, I have produced an algorithm that identify different types of relationships between rules, basing myself on previous work that has been done to help in the crafting of rules used in firewall policies. I have also implemented a prototype of tool using that algorithm. This program, Relationship Identification Tool (RIT), allows to manage a rule set by identifying the different relationships between the rules as well as providing support for the creation and modification of the rule set.

This work was done to answer WeSmart desire to improve their already existing recommendation AI. WeSmart is a company providing support to energy communities, including by the development of a digital platform that allows for energy production and consumption monitoring. The recommendation AI is and will be used in that setting. I have thus started this thesis by presenting energy communities and the context in which they were created, as well as its legal status as defined by the EU and the current situation in Belgium. This was the subject of Chapter 1, with a presentation of WeSmart and its activities. Energy communities connect different actors so they can produce and consume energy together in a more efficient way. The EU is working on promoting their development as a response to the environmental crisis and the challenges that come with a growing need for renewable energy production. In Belgium, the work to specify a legal context for those communities is ongoing, but several pilot projects already exist.

The recommendation AI developed by WeSmart helps energy communities and their participants in the improvement of their energy consumption through the recommendations it provides. I have analyzed the functioning of this AI to suggest a possible way to improve it. This analysis and the suggested improvements were presented in Chapter 2. Since the quality of the AI is strongly tied to the quality of the rules in its knowledge base, I have suggested three ways in which a tool could assist in their creation: First by verifying that there is no redundancy nor contradiction in the rule set; Second by finding the best thresholds values for the attributes used in the rules; Third by allowing more customization of the rules to each specific user. I have then decided to further develop the first aspect.

In order to find an efficient way to verify a rule set for conflicts, I reviewed literature on comparable subjects to find algorithms that could be adapted for energy rules. This analysis was developed in Chapter 3. I have first represented the problem of associating energy data to recommendations as a classification problem and looked for potential solutions in the association classification field, see Section 3.2. This approach proved to be unsuccessful. That analyzis still allowed me to find a better way to represent the knowledge in the rules, by including a single piece of advice in each rule and allowing several rules to be selected simultaneously. That way, the rules are easier to create and understand, and complete recommendation can be formed using the conclusion of several applicable rules.

After realizing that machine learning approaches would not be useful, I studied expert systems (ES) since they rely on manually crafted rules like the recommendation AI. The goal was to understand how ES were inferring conclusions from their rules and see if that insight could help me in the design of the AI and its rule set. Another goal was to determine if firewalls policies would be a good comparison for energy rules or if another type of ES seemed to be an obvious better comparison. This study allowed me to model the recommendation AI as a forward chaining rule-based expert system that is adapted to allow for the selection of several rules simultaneously. I also judged from the review of ES that firewalls would be a field worth exploring for potential solutions adaptable to the detection of energy rule relationships. The presentation of ES and those conclusions were given in Section 3.3.

I then proceeded with the analysis of techniques used to detect and correct conflicts in firewalls policies, which are also knowledge base composed of rules, see Section 3.4. I found that a matrix representation of the rule set and its relationship allowed for easy identification of relationships between two rules. With the work on rules I had done so far and further discussion with WeSmart, I also reached the conclusion that the redundancies and other types of relationships should not be automatically corrected, but solely highlighted. This way, the human experts crafting the rules have more freedom in the potential modifications performed.

Once that algorithm had been found, I have adapted it for an application to energy rules. In Chapter 4, I have presented the final design of the AI and the theoretical solution used for the identification. For that, I have defined relationships of interest for energy rules with regard to the situations they match, namely disjunction, equality, inclusion and overlap. To identify them, I have used matrix representations and a specific encoding of the different types of relationships between the attribute values used in the rules. This allows to easily identify the relationships between rules and to store the information in a way that can be efficiently updated when the rule set is modified.

After the algorithm had been determined, I have implemented it in a prototype, the Relationship Identification Tool (RIT), which was presented in Chapter 5. It allows the management of rule sets by offering support in their creation, modification and storage in CSV files through a visual interface. Of course, it also indicates the different types of relationships between the rules in order to facilitate the work of the user crafting the rule sets. This tool was presented to WeSmart, which had also been included during the whole development process so the solution proposed could be best adapted to their needs and requests. RIT was judged to be an interesting tool to help with the visualization of the rule set and the relationships between its rules.

The algorithm and tool presented in this thesis have been developed for the particular purpose of representing knowledge about energy. However, the rule representation and identification method proposed are generic and they have no intrinsic characteristic that ties them to the energy field. Their usage could thus be extended to other rule-based application in which rules are manually crafted and the relationship between different rules are of importance.

List of abbreviations

- AC: Associative classification
- AI: Artificial intelligence
- CAR: Class association rule
- CEC: Citizen energy communities
- CEP: Clean Energy for all Europeans Package
- DAG: Directed Acyclical Graphs
- ES: Expert system
- EU: European Union
- EMD: Electricity Market Directive
- GHG: Greenhouse gas
- IDC: Inter-Difference Coding
- IDM: Inter-Difference Matrix
- IDV: Inter-Difference Vector
- REC: Renewable energy communities
- RED: Renewable energy directive (from 2009)
- RED II: revised Renewable energy directive (from 2018)
- RIT: Relationship Identification Tool
- SMEs: Small and medium-sized enterprises

List of Figures

List of Tables

Bibliography

[1] *23 JUILLET 2018. - Ordonnance modifiant l'ordonnance du 19 juillet 2001 relative à l'organisation du marché de l'électricité en Région de Bruxelles-Capitale, l'ordonnance du 1er avril 2004 relative à l'organisation du marché du gaz en Région de Bruxelles-Capitale, concernant des redevances de voiries en matière de gaz et d'électricité et portant modification de l'ordonnance du 19 juillet 2001 relative à l'organisation du marché de l'électricité en Région de Bruxelles-Capitale et l'ordonnance du 12 décembre 1991 créant des fonds budgétaires*, Jul. 23, 2018. [Online]. Available: `https://www.ejustice.just.fgov.be/cgi_loi/change_lg.pl?language=fr&la=F&table_name=loi&cn=2018072307` (visited on 05/22/2021).

[2] M. Abedin, S. Nessa, L. Khan, and B. Thuraisingham, "Detection and resolution of anomalies in firewall policy rules," in *Data and Applications Security XX*, E. Damiani and P. Liu, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2006, pp. 15–29, ISBN: 978-3-540-36799-4. DOI: `10.1007/11805588_2`.

[3] M.-L. Antonie and O. R. Zaïane, "An associative classifier based on positive and negative rules," in *Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery - DMKD '04*, Paris, France: ACM Press, 2004, p. 64, ISBN: 978-1-58113-908-2. DOI: `10.1145/1008694.1008705`.

[4] M.-L. Antonie, O. R. Zaïane, and A. Coman, "Associative classifiers for medical images," in *Mining Multimedia and Complex Data*, vol. 2797, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 68–83, ISBN: 978-3-540-20305-6 978-3-540-39666-6. DOI: `10.1007/978-3-540-39666-6_5`.

[5] *Beacon ECCO bocholt, landschapsenergie cvba so.* [Online]. Available: `http://www.agrobeheercentrum.be/Portals/41/Publicaties/20181119_Beacon%20ECCO%20Bocholt.pdf` (visited on 05/23/2021).

[6] A. Bouhoula, Z. Trabelsi, E. Barka, and M. A. Benelbahri, “Firewall filtering rules analysis for anomalies detection,” *International Journal of Security and Networks*, vol. 3, no. 3, p. 161, 2008, ISSN: 1747-8405, 1747-8413. DOI: 10.1504/IJSN.2008.020090.

[7] Brugel. (2021). “Les projets,” Projets innovants brugel.brussels, [Online]. Available: https://projets-innovants.brugel.brussels/projets-innovants/les-projets-405 (visited on 05/22/2021).

[8] F. Chen, Y. Wang, M. Li, H. Wu, and J. Tian, “Principal association mining: An efficient classification approach,” *Knowledge-Based Systems*, vol. 67, pp. 16–25, Sep. 1, 2014, ISSN: 0950-7051. DOI: 10.1016/j.knosys.2014.06.013.

[9] E. U. Commission. (). “A european green deal,” European Commission - European Commission, [Online]. Available: https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en (visited on 05/19/2021).

[10] ——, (Oct. 20, 2017). “Clean energy for all europeans package,” Energy - European Commission, [Online]. Available: https://ec.europa.eu/energy/topics/energy-strategy/clean-energy-all-europeans_en (visited on 05/19/2021).

[11] ——, *Communication from the commission to the european parliament, the council, the european economic and social committee and the committee of the regions, powering a climate-neutral economy: An EU strategy for energy system integration, COM(2020) 299 final*, Brussels, Jul. 8, 2020. [Online]. Available: https://ec.europa.eu/energy/sites/ener/files/energy_system_integration_strategy_.pdf (visited on 05/20/2021).

[12] ——, (Dec. 14, 2020). “Energy communities,” Energy - European Commission, [Online]. Available: https://ec.europa.eu/energy/topics/markets-and-consumers/energy-communities_en (visited on 05/19/2021).

[13] ——, (Nov. 23, 2016). “Paris agreement,” Climate Action - European Commission, [Online]. Available: https://ec.europa.eu/clima/policies/international/negotiations/paris_en (visited on 05/19/2021).

[14] ——, *PNEC 2021 - 2030, partie a - plan national, contexte, objectifs,politiques et mesures*, Dec. 18, 2019. [Online]. Available: https://ec.europa.eu/energy/sites/ener/files/documents/be_final_necp_parta_fr.pdf (visited on 05/22/2021).

[15] ——, (Jul. 16, 2014). “Renewable energy directive,” Energy - European Commission, [Online]. Available: https://ec.europa.eu/energy/topics/renewable-energy/renewable-energy-directive/overview_en (visited on 05/19/2021).

[16] *Décret modifiant les décrets des 12 avril 2001 relatif à l’organisation du marché régional de l’électricité, du 19 décembre 2002 relatif à l’organisation du marché régional du gaz et du 19 janvier 2017 relatif à la méthodologie tarifaire applicable aux gestionnaires de réseau de distribution de gaz et d’électricité en vue de favoriser le développement des communautés d’énergie renouvelable*, May 2, 2019. [Online]. Available: https://wallex.wallonie.be/eli/loi-decret/2019/05/02/2019204063/2019/09/23 (visited on 05/21/2021).

[17] (Feb. 8, 2021). “ECCO - creating new local energy community co-operatives,” NWEurope, [Online]. Available: https://www.nweurope.eu/projects/project-search/ecco-creating-new-local-energy-community-co-operatives/ (visited on 05/23/2021).

[18] S. P. W. Energie. (Dec. 18, 2020). “18/12/2020 : Communautés d’énergie et autoconsommation collective : partageons nos énergies !” Site énergie du Service public de Wallonie, [Online]. Available: https://energie.wallonie.be/fr/18-12-2020-communautes-d-energie-et-autoconsommation-collective-partageons-nos-energies.html?IDD=146181&IDC=8187 (visited on 05/21/2021).

[19] Eurostat. (Aug. 24, 2020). “Climate change - driving forces,” Eurostat, Statistics Explained, [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Climate_change_-_driving_forces (visited on 05/20/2021).

[20] E. W. Fulp, “Firewall policy models using ordered-sets and directed acyclical graphs,” *Wake Forest University Computer Science Department, Tech. Rep*, 2004.

[21] ——, “Optimization of network firewall policies using directed acyclical graphs,” in *Proceedings of the IEEE Internet Management Conference*, Nice, France: IEEE, 2005, p. 4.

[22] C. Grosan and A. Abraham, “Rule-based expert systems,” in *Intelligent Systems: A Modern Approach*, ser. Intelligent Systems Reference Library, C. Grosan and A. Abraham, Eds., Berlin, Heidelberg: Springer, 2011, pp. 149–185, ISBN: 978-3-642-21004-4. DOI: 10.1007/978-3-642-21004-4_7.

[23] O. M. Group. (Dec. 2017). “About the unified modeling language specification version 2.5.1,” OMG, [Online]. Available: https://www.omg.org/spec/UML/ (visited on 06/09/2021).

[24] A. Hannoset, L. Peeters, and A. Tuerk, "Energy communities in the EU -task force energy communities," D3.12.d, p. 137. [Online]. Available: https://www.h2020-bridge.eu/wp-content/uploads/2020/01/D3.12.d_BRIDGE_Energy-Communities-in-the-EU-3.pdf (visited on 05/20/2021).

[25] T. Katic and P. Pale, "Optimization of firewall rules," in *2007 29th International Conference on Information Technology Interfaces*, ISSN: 1330-1012, Cavtat, Croatia: IEEE, Jun. 2007, pp. 685–690, ISBN: 978-953-7138-09-7 978-953-7138-10-3. DOI: 10.1109/ITI.2007.4283854.

[26] M. Legast. (2021). "Relationship identification tool," [Online]. Available: https://github.com/Magalii/Relationship-Identification-Tool.

[27] A. X. Liu and M. G. Gouda, "Complete redundancy detection in firewalls," in *Data and Applications Security XIX*, S. Jajodia and D. Wijesekera, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2005, pp. 193–206, ISBN: 978-3-540-31937-5. DOI: 10.1007/11535706_15.

[28] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," p. 7, 1998.

[29] B. Liu, Y. Ma, and C.-K. Wong, "Classification using association rules: Weaknesses and enhancements," in *Data Mining for Scientific and Engineering Applications*, ser. Massive Computing, Boston, MA: Springer US, 2001, pp. 591–605, ISBN: 978-1-4615-1733-7. DOI: 10.1007/978-1-4615-1733-7_30.

[30] N. Masri, Y. A. Sultan, A. N. Akkila, A. Almasri, A. Ahmed, A. Y. Mahmoud, I. Zaqout, and S. S. Abu-Naser, "Survey of rule-based systems," *International Journal of Academic Information Systems Research (IJAISR)*, vol. 3, no. 7, pp. 1–23, 2019.

[31] I. Mothersole and M. J. Reed, "Optimising rule order for a packet filtering firewall," in *2011 Conference on Network and Information Systems Security*, La Rochelle, France: IEEE, May 2011, pp. 1–6, ISBN: 978-1-4577-0735-3. DOI: 10.1109/SAR-SSI.2011.5931399.

[32] U. Mustafa, M. M. Masud, Z. Trabelsi, T. Wood, and Z. Al Harthi, "Firewall performance optimization using data mining techniques," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Sardinia, Italy: IEEE, Jul. 2013, pp. 934–940, ISBN: 978-1-4673-2480-9 978-1-4673-2479-3 978-1-4673-2478-6. DOI: 10.1109/IWCMC.2013.6583682.

[33] S. S. Naser, "SL5 object: Simpler level 5 object expert system language," *The International Journal of Soft Computing, Mathematics and Control*, vol. 4, no. 4, pp. 25–37, Nov. 30, 2015, ISSN: 22014160. DOI: 10.14810/ijscmc.2015.4403.

[34] A. Papagrigoriou, P. Petrakis, and M. Grammatikakis, “A firewall module resolving rules consistency,” in *2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES)*, Hamburg: IEEE, Jun. 2017, pp. 47–50, ISBN: 978-1-5386-1157-9. DOI: 10.1109/WISES.2017.7986931.

[35] Shu-Hsien Liao, “Expert system methodologies and applications—a decade review from 1995 to 2004,” *Expert Systems with Applications*, vol. 28, no. 1, pp. 93–103, Jan. 1, 2005, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2004.08.003.

[36] G. W. Snedecor and W. G. Cochran, *Statistical methods*, 8th ed. Ames, Iowa: Iowa State University Press, 1989.

[37] F. Thabtah, P. Cowling, and Y. Peng, “MCAR: Multi-class classification based on association rule,” in *The 3rd ACS/IEEE International Conference onComputer Systems and Applications, 2005.*, ISSN: 2161-5330, Jan. 2005, pp. 33–. DOI: 10.1109/AICCSA.2005.1387030.

[38] F. Thabtah, P. Cowling, and Y. Peng, “Mmac: A new multi-class, multi-label associative classification approach,” in *Fourth IEEE International Conference on Data Mining (ICDM’04)*, 2004, pp. 217–224. DOI: 10.1109/ICDM.2004.10117.

[39] F. Thabtah, “A review of associative classification mining,” *The Knowledge Engineering Review*, vol. 22, no. 1, pp. 37–65, Mar. 2007, ISSN: 0269-8889, 1469-8005. DOI: 10.1017/S0269888907001026.

[40] K. Wang, S. Zhou, and Y. He, “Growing decision trees on support-less association rules,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’00*, Boston, Massachusetts, United States: ACM Press, 2000, pp. 265–269, ISBN: 978-1-58113-233-5. DOI: 10.1145/347090.347147.

[41] Wenmin Li, Jiawei Han, and Jian Pei, “CMAR: Accurate and efficient classification based on multiple class-association rules,” in *Proceedings 2001 IEEE International Conference on Data Mining*, Nov. 2001, pp. 369–376. DOI: 10.1109/ICDM.2001.989541.

[42] WeSmart. (2020). “WeSmart,” WeSmart, [Online]. Available: https://wesmart.com/en (visited on 05/22/2021).

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**