СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО
МАТЕМАТИКА И ИНФОРМАТИКА

SOFIA UNIVERSITY
ST. KLIMENT OHRIDSKI

FACULTY OF
MATHEMATICS AND INFORMATICS

Semantic Technologies in eHealth

(Application of Machine Learning Technologies in Supporting Independent Living of the Elderly and Disadvantaged)

by

Kristin Ilieva Aleksandrova

Abstract

for the purpose of acquiring the educational and science degree "Doctor"

in

the field of 4.6. Informatics and Computer Sciences,

Doctoral Program "Information systems" - Knowledge-based systems

Advisor: Professor Maria Nisheva-Pavlova, Ph.D.

Sofia, 2024

1 INTRODUCTION

In recent years, we have observed that the worldwide aging population has been growing at an unprecedented rate, posing significant challenges to healthcare systems and social support networks, which in term has economic consequences (Akhter, M. & Kamraju, M., 2023), (Santos, E., 2023). The concept of Ambient Assisted Living (AAL) has emerged as a promising solution to address the needs of this aging population. AAL systems aim to improve the quality of life and independence of older adults by seamlessly integrating smart technologies into their living environments. However, despite the increasing interest and potential benefits of AAL systems, there is a gap in their adoption and worldwide availability which poses the need for further research and development in discovering and mitigation the limiting factors which would enable the creation of effective and user-friendly prototypes.

1.1 Background of the problem

In the last year a growing worldwide topic has been the rate of demographic aging and the challenges it poses for all countries in regards to healthcare and social systems. The same can be seen in the messaging of the World Health Organization (WHO): “By 2030, 1 in 6 people in the world will be aged 60 years or over. At this time the share of the population aged 60 years and over will increase from 1 billion in 2020 to 1.4 billion. By 2050, the world’s population of people aged 60 years and older will double (2.1 billion). The number of persons aged 80 years or older is expected to triple between 2020 and 2050 to reach 426 million.” (Steverson, M., 2022). Currently personal care, nursing homes and hospitals prove to be both expensive and unable to handle the prognosed number of people in the upcoming years. This is putting more focus on coming up with ways for elderly people to live by themselves, with minimized assistance from caretakers, family, or doctors.

The United Nations (UN) General Assembly declared 2021–2030 the UN Decade of Healthy Ageing and asked WHO to lead the implementation. The UN Decade of Healthy Ageing is a global collaboration bringing together governments, civil society, international agencies, professionals, academia, the media, and the private sector for 10 years of concerted, catalytic and collaborative action to foster longer and healthier lives. (Steverson, M., 2022) details of which can be seen in the accompanying plan of action (WHO, 2022). There we also see the requirement of developing “assistive technologies, while ensuring that use of these services does not cause the user financial hardship” and “Encourage use of safe, affordable, effective digital technology in integrated care. Analyse the labour market and conduct needs-based planning to optimize current and future workforces to meet the needs of ageing populations”. This is where Ambient Assisted Living (AAL) Systems come into the picture. Their sole purpose is to improve the independence and quality of life of people in need of assistance, whether that is in a nursing home or in their own home environment. AAL systems have shown immense potential to improve the quality of life of not just elderly people, but also of people with disabilities.

At present we can aggregate the available care options for elderly people in three main categories, each coming with its own raising costs. “The cost of long-term care for the elderly, including both cost of nursing home and home health agency, reached 61 billion euro in 2019. Half of these spending are for nursing homes while only about 22.5% of beneficiaries use these institutions. Out-of-pocket spending differs greatly between modes of care. Out-of-pocket expenditures make up only about 7% of total expenditures for home care. In nursing homes, 41% of expenditures are out-of-pocket payment.” (Geyer, J. et al., 2023)

- Institutional care. For example, nursing homes, which are medical institutions, employing medical personnel that simultaneously take care of many people, there is no focus on making the environment resemble a home one.
- Home. In this case, the person keeps living in their home and are taken care of by a caretaker, that usually is someone from their family like a partner or children. Occasionally, it’s possible that a caretaker with a background in medicine is employed.
- Community-care, homes specially build for the elderly, etc. Some countries and regions, like Germany have housing that is dedicated to the elderly. These homes are made accessible and are employed with emergency buttons and easy access to emergency services. They are an in-between option that allows the person autonomous living while also introducing some form of monitoring and care.

Meanwhile, we observe a strong preference of elderly people to stay at home. A practice usually referred to in literature as “aging in place”. “Older people want choices about where and how they age in place. “Aging in place” was seen as an advantage in terms of a sense of attachment or connection and feelings of security and familiarity in relation to both homes and communities. Aging in place related to a sense of identity both through independence and autonomy and through caring relationships and roles in the places people live.” (Janine L. W. et al., 2012)

Research in China, showed that the preference of 91.9% of respondents was home-based elderly care, followed by community-based care and medical-nursing care with institutional care being the least preferred alternative (Du, J. et al., 2023). Similar conclusion was seen in a study in Japan, where “Among the 10,119 responders, 61% chose their home as the most desirable place to spend their final days.” (Saito, T. & Konta, T. & Kudo, S. & Ueno, Y., 2023). One contributing factor to this is the perceived quality of care people receive in institutions, which has been observed in other research (Fahley T, et al. 2003), (Mukamel, D. et al., 2023), (Borsa, A. et al., 2023). The question on what acceptable care options would be, is also unclear as shown in “Contestations in coping with elderly care: an intersectional analysis addressing family caregivers in Germany” (Auth, D. & Leiber, S. & Leitner, S., 2023).

In other research we can see that “Slightly more than half of the studied sample (57%) defined their current quality of life with positive evaluations, whereas 18% presented a negative evaluation of it. A group 0f 25% defined their current lives as neutral or having both values (positive and negative) … The main source of reported daily well-being was the involvement with rural or domestic activities. Among the interviewed, lack of health was the main source for not presenting well-being, although there was interpersonal variability regarding what each subject considered as loss of health.” (Xavier, F. et al., 2003)

Dementia is a common disease, affecting in different severity many people. By 2050 it is projected that there will be 154.8 million cases, compared to 57.4 million in 2019 (Ghith, N., 2022). “Dementia alters eating behaviors, hunger and thirst cues, swallow function, ability to self-feed, and recognition and interest in food. There is significant variation in the reported prevalence of malnutrition among older people who live in long-term care. … The prevalence of malnutrition ranged from 6.8 to 75.6%, and the risk of malnutrition was 36.5–90.4%. The pooled prevalence of malnutrition in those with dementia in long-term care was 26.98% (95% CI 22.0–32.26, $p < 0.0001$, $I2 = 94.12\%$). The pooled prevalence of the risk of malnutrition in those with dementia was 57.43% (95% CI 49.39–65.28, $p < 0.0001$, $I2 = 97.38\%$). Malnutrition is widespread in those with dementia living in long-term care.” (Perry, E.& Walton, K. & Lambert, K., 2023)

Additionally, we can observe a routine present in elderly people’s life, and this is common with the decrease of cognitive and psychological abilities, as concluded by Bergua, V. et al., 2006:

“Although routine activities are important to normal functioning across all phases of life, their expression in older people may be associated with cognitive and psychological vulnerability. The relationship between these variables was explored in 235 elderly French participants from the PAQUID cohort study. Cross-sectional positive associations were found between preferences for routines, anxiety and depression levels, and cognitive complaints. General cognitive decline over a three-year time span was also associated with a greater desire for routines at the end of this period. The progressive routinization of behaviors and activities in older people is discussed as a marker of affective and cognitive vulnerability, and its understanding has potential for improving the early detection of adaptation difficulties and overall care in this population.” (Bergua, V. et al., 2006)

The same is especially the case for dementia patients, where maintaining daily routines and environmental familiarity is crucial to the person’s mental clarity and stability and the quality of the care they receive. “We demonstrate how metastability provides an understanding of the ever-changing rhythms of every day and allows us to move beyond the immediacy of arrhythmic breaks and explore the subtle changes that occur in (poly)rhythms. Thus, eurhythmia as a metastable equilibrium allows us to explore the gradual and subtle development of, and changes to, dementia care and other routine practices in health geography.” (Osborne, T. & Lowe, T. & Meijering, L., 2023)

1.2 Problem statement

The current state of the problem space is putting more focus on coming up with ways for elderly people to live by themselves, with minimised assistance from caretakers, family or doctors. From the perspective of computer sciences, to contribution to a solution is contained are Ambient Assisted Living (AAL) Systems, which encapsulates all technology advancements that find application in improving the autonomy and quality of life of elderly people. This explains the growing amount of research in the AAL space, nevertheless despite the growing demand very few of the initiatives and prototypes find productive and widespread application, which asks the question on why this is the case. We can hypothesise that in previous prototypes of AAL systems, the focus was on enabling technical functionality at the expense of security, data privacy, etc., (Schomakers, E.M. & Ziefle, M., 2022) as these systems can increase the quality of life of elderly people and can provide vital support in their daily activities.

One potential reason for the limited availability could be the cost of support per person in the system. This is a combination of a multitude of factors. To name a few: the cost of the infrastructure for running and supporting the system; maintaining and securing up to date personal information; the time and engagement required from a human caretaker. One way to reduce the cost of running a large-scale system is the implementation of Machine Learning algorithms. One potential space is the detection of deviation in the standard person’s behaviour. For example, recognizing if they are taking their medication regularly, have they missed a dose or took a double dose because they forgot, are their sleeping patterns affecting their cognitive ability, etc. Having this knowledge would allow for the same caretaker to sustainably monitor more people that currently, as they can easily react to emergency situations, while getting a daily report on the normal behaviour of each person in their care.

Human behaviour is a complex space, in this work we are focusing on elderly people and especially dementia patients, which is a space where we commonly see a routine present, with deviations pointing to larger problems and potentially dangerous episodes for the person, like getting lost or confused. In this work we pose the hypothesis that a solution of this problem is to consider outlier detection machine learning models, in this case we can narrow the problem down to establishing a pattern of the normal behaviour of a person and considering any outliers of this pattern a break in routine, that needs to be evaluated and reacted on by a person.

1.3 Research questions, hypothesis, and goal

With that in mind, let us formulate several research questions that we aim to answer.

1. Can we create a cost-conscious Ambient Assisted Living (AAL) System?
   When talking about cost aware there are several parameters on which this can be judged, which we strive to optimize. This includes the computational resources for running the AAL System and for training the ML behavioural models; the cost of development and support of the system and models; the cost per user of the system for initial onboarding to the system; the cost per person for providing a functional AAL system and up to date machine learning models.
2. Can we create a data privacy compliant AAL System, more specifically a GDPR-aware one? Also, what would be the implication of GPPR on the trained machine learning algorithms?
3. Would an AAL System benefit from enhancing it with a personalized machine learning algorithm, trained on the collected data that aims to identify outliers in the person's behaviour and raise the appropriate alert to their caretaker? If so, what would be the best approach to tackle the problem, without compromising on the cost of the system and the established data privacy requirements.

Based on the questions above, we can formulate the following hypothesis:

"We can reuse open-source smart home middleware software to create a cost- and data privacy-aware AAL system, extend it with machine learning algorithms in a useful manner and prove that association rule mining (ARM) algorithms can be used for human behavioural recognition and they would be the better choice compared to standard outlier detection approaches for an AAL system as they are overall cheaper, easier to conform to data privacy regulations and they have explainable results."

The goal of this thesis is to prove the hypothesis as a whole and in its derivative parts by answering the defined research questions.

2 SYSTEM ARCHITECTURE AND IMPLEMENTATION

In this work, we propose the stipulation that ready-made open-source middleware platforms are more suitable for the creation of Ambient Assisted Living Systems, then their domain specific counterparts, due to their wider adoption in a multitude of fields and openness to new use cases nature. Additionally, we hypothesise on the wide application of machine learning algorithms in the recognition of behavioural patterns in elderly patients with light cases of dementia. To truly verify and illustrate those points, we create an AAL system prototype, conscious of the required effort to create and use the system and aiming to keep the total cost of operation (TCO) of the resulting solution to a minimum. We then simulate the work of said prototype and evaluate its usefulness in the proposed use-cases. In the following sections, we go over the different analysis and development decisions, that were made as part of the creation of said prototype.

Previously we looked closely at the details of four of the most prominent candidates for a middleware platform, on which to base a prototype. There were many other options that failed to cover half the criteria and as such were excluded from this comparison. We defined nine criteria on which the platforms were judged, and we summarized the conclusions for each one of the prominent candidates – openHAB, ThingsBoard, universAAL, OpenRemote. OpenRemote fully facilitates the desired scenario and provides opportunities for further development and extension, at a reasonable trade-off amongst the benefits and costs. Therefore, in this work we build a prototype based on OpenRemote, that would exemplify the functionality and potential of this type of system. We must not rule-out universAAL completely, instead an additional goal of the prototype would be to display the strengths and weaknesses of an OpenRemote-based AAL system, in turn justifying or discouraging further research to create a similar prototype, based on universAAL.

2.1 First version of the prototype based on OpenRemote

Now that we have selected our middleware platform, we can start developing our AAL system prototype. There are several questions we need to answer and validate in the very first prototype we test, that would serve as a foundation for future versions: How is an OpenRemote distribution packaged and it's development environment configured; How are custom modifications implemented an how are they packaged into a finished product; How are sensors modelled and what are the possibilities to create rules based on them; Can we create a virtual simulation of one of the target scenarios we previously defined, without connecting the system to real sensors and without compromising the system's functionality? This is the premise of this section.

As OpenRemote is an open-source solution, there are two ways to get access to the source code and start developing. One is to clone the GitHub repository and build the system locally. The other option would be to utilize the created Docker container images for OpenRemote, published in Docker Hub. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime and in the case of Docker containers – images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software always runs the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space . Containers take up less space than VMs, they are typically tens of MBs in size, can handle more applications and require fewer VMs and Operating systems. (Use containers to Build, Share and Run your applications, n.d.) This ensures the ease of distribution, we initially defined. The resulting AAL System Prototype is also packaged and distributed as a containerized application. If we manage to build it as an abstraction layer to the current OpenRemote containers, we can reuse all updates and improvement to the platform.

In order to create the containers, there are two things we need. A Docker Engine and a docker-compose file. The Docker Engine uses a long-running daemon process, called dockerd, to create and manage Docker objects, including containers. It acts as a client-server application with the daemon process running on the server side and a command line interface (CLI) that interacts with the daemon. There are several ways to obtain a Docker Engine, but one of the easier methods is the Docker Desktop application. Docker Desktop includes the Docker Engine we need, Docker CLI client, Docker Compose, Docker Content Trust, Kubernetes, and Credential Helper.

Let us focus on Docker Compose. It is a tool that was developed to help define and share multi-container applications. This is realized with the creation of a specific YAML files, also referred to as Docker compose files, there all services, networks, and volumes for a Docker application are defined, and a container is spun up or teared down based on that information. Said file is situated at the root of the app project and usually named “docker-compose.yml”. In the case of OpenRemote, the file is standard and allows us many modifications, based on the described parameters.

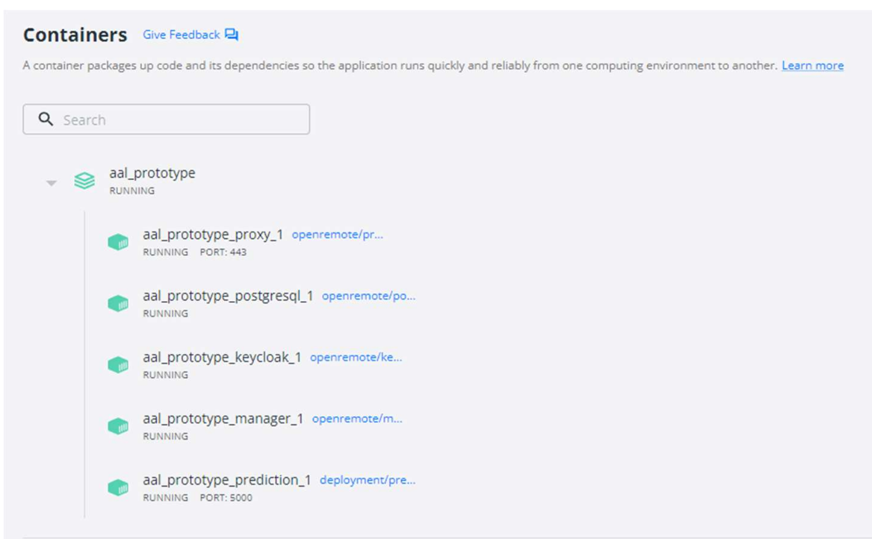

Figure 1. *Custom OpenRemote installation in Docker Desktop*

The new container we name ‘prediction’. For its creation we need several files. To start we create a train.py fail that contains all the logic about the model training, including getting the data from the postgresql container, cleaning and splitting it. We finalize this file’s contents in the results section, after we compare the accuracy of different approaches. We create a “requirements.txt” file, that describes all the needed python packages, that need to be installed, so that the training is possible. We also mentioned that this container needs to be able to handle incoming requests and make inquiries to the model about them, this is the role of the api.py file. In order to create a 'predictions' container, we need to create a 'Dockerfile', that describes the type of container and the operations that need to be performed. It can be split into four main sections, first we create a container from the image of a Jupyter[1] container. Afterwards we copy the requirements file and use python's pip to install all of the needed modules. Afterwards we create a file structure and environmental variables, that allow the api code to find the model files and executables and derive the needed predictions. Finally, we copy the two python files, and we train the model and run the server, respectively. The resulting containerized structure can be seen in Figure 1.

2.1.1 Customizations

As we already saw in the previous section, the OpenRemote deployment is based on Docker and many modifications can be achieved when specifying the parameters for spinning up the needed containers. For example, initially the master password of the installation is "secret", this is based on two parameters, that must match in value, one is in the definition of the keycloak container, named KEYCLOAK_PASSWORD, and the other is in the definition of the manager service, named SETUP_ADMIN_PASSWORD.

In addition to adjustments done via the Docker Compose file, we can further customize the OpenRemote visuals and user experience. All of this is done in a structured manner, that has been properly described in the official OpenRemote documentation, that is hosted on GitHub. Let us start by taking a look at the manager_config.json file. This is a JSON, that allows modifications on the look and feel of the manager application, which is also the location of the UI. As we already discussed in the initial middleware evaluation, OpenRemote supports multitenancy in the form of different "realms" for the different users. In line with that concepts UI components can be specific per realm. The tab name of the application is defined by the 'appTitle' parameter, in the styles parameter we see the main application colors. In this example, we have defined the default preset for realms if nothing else has been specified and additional configurations for a realm with a technical name 'ivan', that is representing of the home of our example persona Ivan.

Similarly, we can exchange the map, to display a region of our choosing. In this prototype, we take a map of Bulgaria and position the simulated example home nearby the building of the Faculty of Mathematics and Informatics of the Sofia University "St. Kliment Ohridski". The requirement towards maps is that they come as a vector tile data, more specifically Mapbox GL format. Other formats require additional effort but are not impossible to implement. Additionally, there is support for raster maps (Mapbox JS), but they require an additional map container to be running, that then serves the map data to the manager application. Nevertheless, for our scenario a vector map of the Sofia city is sufficient, as we

[1] https://jupyter.org/

aim to represent several households simultaneously and have the possibility to further extend the system's functionalities outside of a person's home.

There are two files we need in other do define the new map to the application: mapsettings.json and mapdata.mbtiles. The mapdata.mbtiles contains the vector map data, while the mapsettings file contains settings related to the map tiles source data and also UI rendering settings, like the center point of the map, as well as levels of zoom. Additionally, in this JSON file it is possible to define a separate setting for different realms. This allows the same application to focus on different households, depending on the user accessing it. Both files are then located in a folder named map, that is on the same level as the manager folder.

2.1.2 Simulating target scenario

When defining the target persona for our use case we stressed the importance of monitoring electrical appliances, as an elderly person can be distracted and forget that they were cooking, in turn creating a risk of a fire in the house. Therefore, the first task in the prototype creation was creating a simulation of the scenario in question. If the approach of the middleware for handling these cases is not suitable to the persona we have in mind, it is only reasonable that we look into alternatives for the foundation of the prototype. With that in mind, let us model the easiest scenario to monitor – namely has the stove been left unsupervised and turned on? The same question about an electrical appliance like the oven is different, as many oven recipes anticipate a longer cooking time, and without feedback from the person cooking, it's hard to judge if the oven should have been on for 3 minutes or 2 hours, not no mention all bakes require a pre-heated oven, which would conflict with any rule about monitoring for an empty turned on oven. Now for the stove, it is easier, as there is no concept of preheating, so any time the stove is on and there are no pans directly on it is a cause for concern. Also, it is not recommended that the elderly person leaves the stove unattended for long. In other words, this problem can be derived to the question is the stove on and is there any presence in the kitchen?

Therefore, the first step would be to create assets representing the Kitchen, including the motion detection sensor and a stove sensor, indicating if the stove is on or off. The room asset has no additional attributes and only serves as an organizational entity for all devices located in the kitchen. The door asset has one mandatory attribute, which is the position of the door in a Boolean format, in Figure 2 the value is true, meaning the door is open. If we had such sensors equipped to the door this would be the place to connect the values in the system. For the sake of the simulation the door asset is not used.

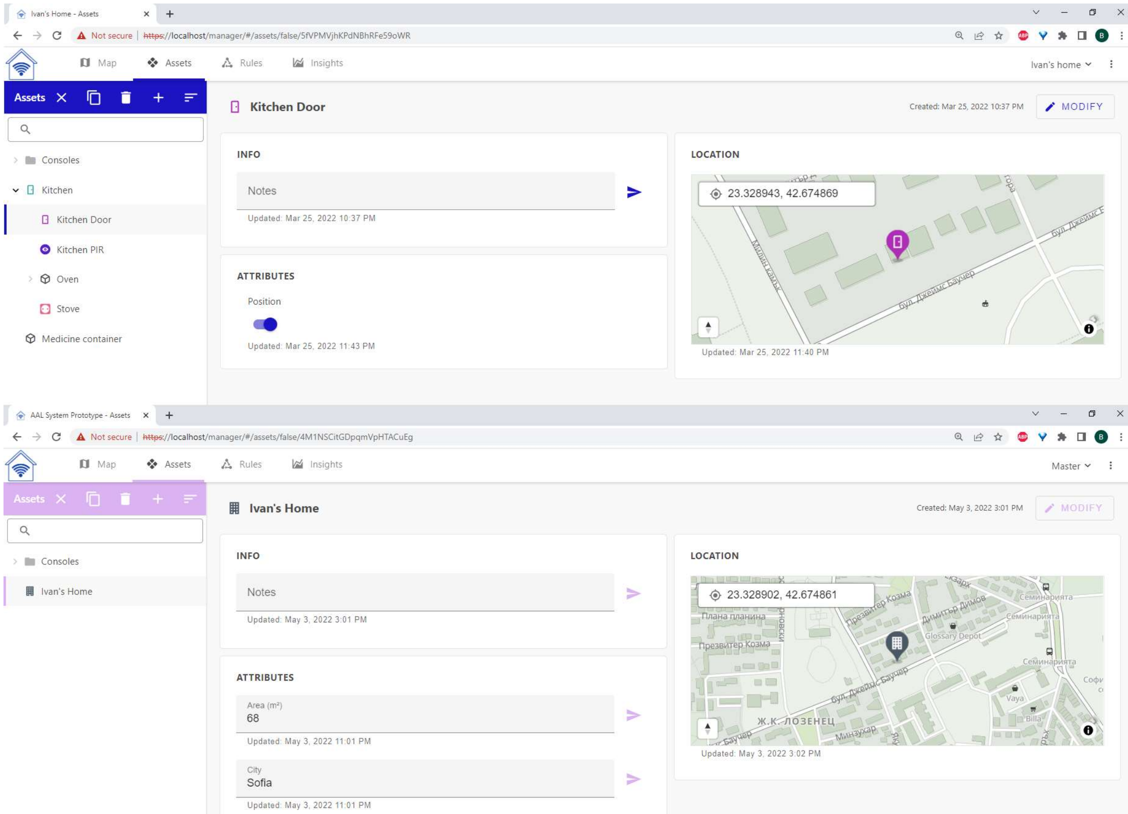

Figure 2. *Assets modeled in the realms "Master" and "Ivan's home" for the sake of scenario simulation*

We also see the same asset page, but from the "Master" realm. As we are logged in as an administrator user, we have edit rights on all realms and we are free to switch as we choose. As we expected, assets are specific to the realm, they have been created in. Therefore, in the "Master" realm there is only one asset and that is the building of Ivan's apartment, as an indication of the maintained households, the system is currently supporting. All the kitchen created devices are restricted to the realm of "Ivan's home". This proves the initial claim of multi-tenancy and allows us to further enhance the system in that direction.

At the current stage of prototype development, we are trying to visualize a concept of the system's functionalities and approach to problem resolution. In line with that we do not have a reference home or devices to obtain real-time data from, so we need to create virtual devices, that behave in a similar manner to their real counterparts. For the defined scenario we need a virtual motion detector PIR device, situated in a person's kitchen and a stove, that can notify the system, when it has been turned on or off.

> One approach to create this type if a device is via OpenRemote's Groovy rules. One the asset for a stove was created, one of its attributes a stove is named 'onOff'. This is a Boolean value that is true, when the stove is turned on. Instead of connecting this asset to an actual device, we can use a Groovy rule to simulate the values. OpenRemote has a specific class structure, that allows to find the needed asset by their ID and set the

value to our desired one. In Figure 3 we can see the rule that simulates that the stove has been turned on. You also might notice that there is a repetition to this rule, namely Mondays, Tuesdays, Thursdays, Fridays, and Sundays at noon from 12:00 to 13:00 o'clock. This combined with the set frequency of 10 second, ensures that every day for one hour our virtual oven will report that it is turned on. The virtual stove off rule is the same code, but with a different attribute value of false and a reoccurrence of once every mentioned day at exactly 13:00 o'clock.

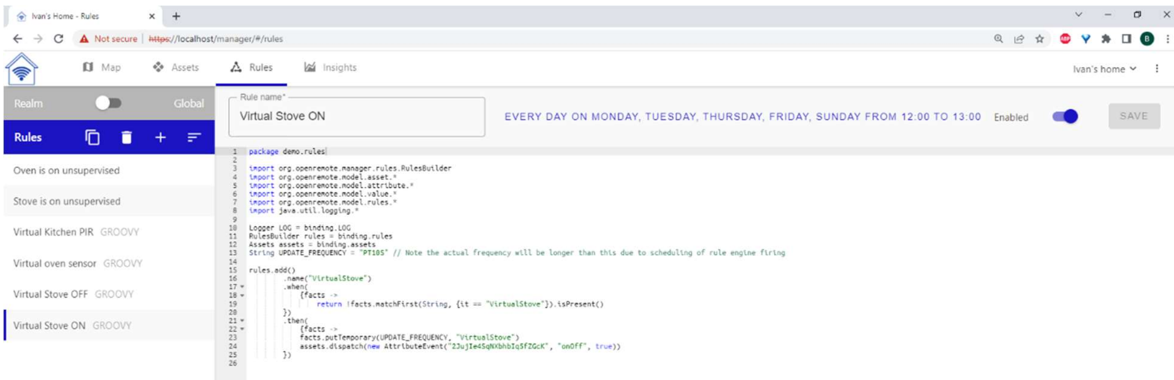

Figure 3. *"Virtual stove ON" Groovy rule*

The same applies for the Virtual PIR sensor. It operates every day from 8:00 to 19:00 o'clock and reports a presence in the Kitchen at random intervals. Initially this was realised by randomizing the intervals at which a presence is reported. (Figure 4)

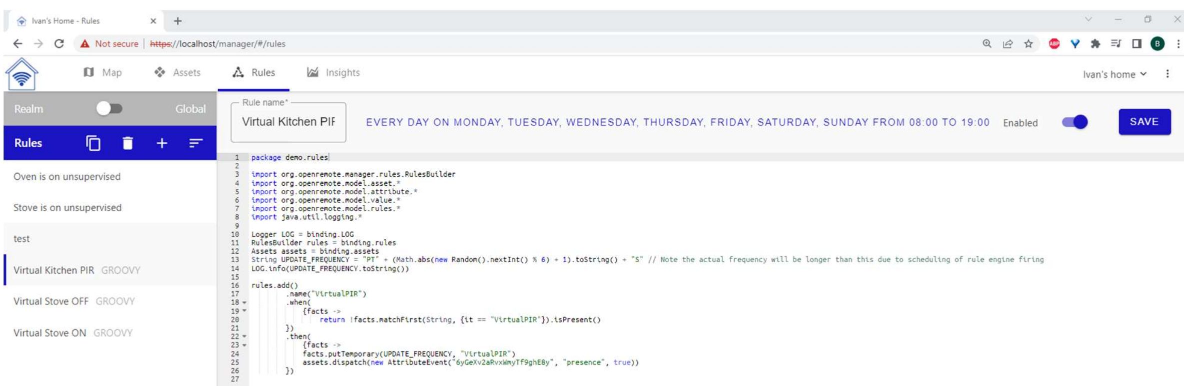

Figure 4. *"Virtual Kitchen PIR" Groovy rule*

In Figure 5, the virtual sensors have been combined by a straight-forward when-then rule, that basically says, that if the stove is on and there is no presence detected in the kitchen, the person's caretaker should be notified via email. This was later replaced by a more sophisticated groovy rule, that takes into account the elapsed time since a presence was detected. Additionally, a notification is sent to the caretaker's phone via the Pushsafer application, that is a forward of the email or a direct POST request from the Groovy script.

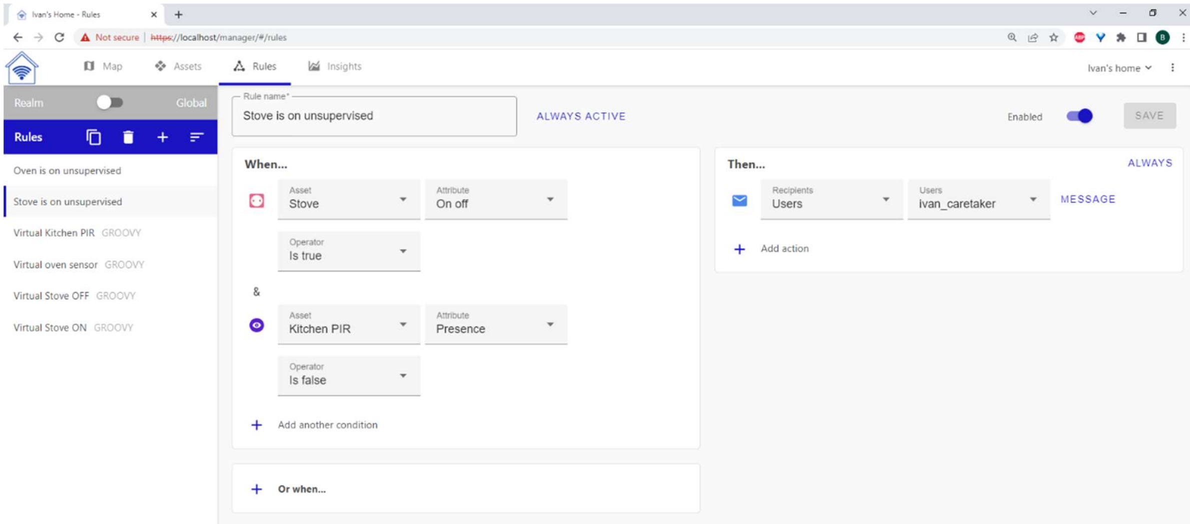

Figure 5. *"Stove is on unsupervised" When-Then rule*

3 GDPR REQUIREMENTS AND COMPLIANCE

One foundational flaw of AAL systems, that hinders their adoption is the securing of personal data. As AAL systems benefit from their status of systems with impact on public health and are often part of research projects, they are not obliged to conform to any legislation or security standard. In turn very few do so, which hinders their wider adoption and development. In this work we want to argue, that ensuring data privacy can be easily achieved, when said compliance is considered at the start of product development. As reference in this work, we are using the EU General Data Protection Regulation (GDPR). One of the foundational principles of this legislation is Privacy by Design and Default. In our case this would translate into understanding GDPR and with that in mind defining the system functionalities. Many times, this has proven to be the leading approach, when introducing security into a system, as with time and the project's growth the cost of reconstruction for the sake of data protection exponentially grows. Therefore, in a separate analysis we have gone in detail of the implications of GDPR on an AAL system, that aims to combine traditional inobtrusive data collection with machine learning algorithms to derive insights about the physical and mental condition of the person in the system's care. As a result of that work, we derived 15 requirements towards an AAL system prototype and its behaviour:

1. GDPR compliance needs to be verified by a legal team. They are the ones, that determine whether the processing is fair, legal, and transparent; what are the lawful reasons for processing; for which data and to what extend is each data subject request (DSR) applicable; are there results of automated processing personal data and as such subject to the GDPR, etc.
2. When designing the processes, we need to ensure that at all times when data is processed it is minimized, including when data is shared with an additional processor.

3. Before constructing the architecture of the AAL system, based on the security risks there needs to be a “Data Protection Impact Assessment” done.
4. Data needs to be decoupled in such a way, that if a data subject wants to see their own data, we are able to provide it without compromising the confidentiality of all other data subjects in the system, or when data needs to be deleted it does not interfere with the overall system performance.
5. We need to establish sufficient means of verifying the requester of DSRs. There are many means of authentication, and we can choose a suitable option for the AAL system.
6. We need to analyse the expected amount of DSR related requests, so that we can ensure their processing is done within 30 days, this also includes cases of data deletion or restriction.
7. We must draft and publish a Privacy notice, based on the system’s use-case and functionality.
8. We need to establish means of notification for data subjects in case of data breaches and as a potential channel for communication on the topic of purpose and consent.
9. Based on the legal evaluation, it needs to be possible to select and delete only personal data, that is subject to the right to erasure, which is defined by the legal team.
10. We need to have a way to flag data, so it is not used in processing in cases where processing has been restricted as per the DSRs.
11. We need to have a sufficient strategy for Backups, as per GDPR we are not obliged to delete data from backups, we need to be aware of the possibility, that we restore personal data which is restricted or deleted.
12. Depending on the algorithms used we need to evaluate anonymization and pseudonymization. If the former is feasible for all personal data, we are GDPR compliant, the latter is recommended by GDPR as a security measure.
13. End-to-end encryption is also referred in GDPR, as a necessary security measure.
14. Depending on how critical the system functionality is, we need to evaluate the possibility to invest in a high-availability or disaster recovery setup.
15. Identity management - each AAL system needs to establish proper role definitions and access levels, so that personal data is exposed in a need-to-know manner, with the appropriate purpose defined.

4 BEHAVIOURAL PREDICTIONS

In the previous sections we have established the creation of an Ambient Assisted Living system prototype, that supports the daily activities of elderly people with custom sensors and rules, so that the system is tailored closely to the person it is intended for. In the initial premise we considered the application of this system to elderly people with light cases of dementia. The premise there is that we can design a general set of rules that can pinpoint if a person is deviating from their usual behaviour, this can be cases where the patients forget they have eaten and easts multiple times per meal, or the opposite they believe they have already eaten and in turn they forgo food for the whole day. We showcased how these simple scenarios can be modelled via a few simple rules for the already connected sensors, but when we are talking about deviations from standard behaviour, an obvious approach would be the implementation of various machine learning algorithms, that would create behavioural models. The idea behind those models would be to train them on a standard set of data, containing various activities of daily living (ADL), and ask them to judge if a current observation is standard behaviour for this person or not.

4.1 Neural Networks

Neural networks have a wide application in anomaly detection (J. E. D. Albuquerque Filho, 2022), as they are capable of learning complex patterns in high-dimensional, time-varying data. This makes them suitable for application on real world raw data, where the volume and complexity of the data can grow very quickly out of the range of handling for other algorithms. Neural networks are also suitable for usage in our case of anomaly detection in human behaviour, as they do not require training on a dataset with identified anomalies, instead they are capable of extracting the relevant features from raw data and identifying the patterns of input. In this manner they anomalies are identified, based on the difference between their prediction of the patterns and the actual observed values. This mechanism of pattern recognition allows neural networks to perform better than the usual statistical methods, as unlike them neural networks do not assume a specific data distribution. They can learn and detect anomalies without requiring prior information and also derive contextual information about the location and severity of the anomaly. Building onto that, one of the strengths of neural networks is their ability to learn intricate and non-linear patterns from data, making them effective in capturing complex relationships between variables. Neural networks can adapt their internal parameters and structure during the learning process. This flexibility enables them to generalize well to diverse types of anomalies, making them suitable for various anomaly detection tasks.

4.1.1 Autoencoders

Autoencoders are a feedforward neural network where the input shape is the same as the output one. The idea is that the input is compressed into a lower-dimensional code and then the output is reconstructed from this representation. Therefore, each record would be encoded and then decoded by the neural network to its previous state. If the decoding is successful, we can conclude that this is an entry the neural network is familiar with and categorize it as normal behaviour. If the neural network cannot reconstruct the record, it is considered an anomaly. In this section we look closer at autoencoders (Jordan, J., 2018) and their application in recognizing anomalies in human behaviour.

The easiest approach would be to provide the dataset to the tokenizer without any additional intervention. However, the columns ‘Room’, ‘Object’ and ‘Sensor’ only serve to clarify the sensor data to the end user, the autoencoder would not benefit from them and they would just add additional levels of complexity, therefore we can drop them right away. Additionally, we do not expect people to act in the same second or millisecond, we are more interested in the part of the day each activity is performed, therefore it only makes sense that we round the times to the nearest hour and consider a region around minutes if additional accuracy in the real use case is needed. The resulting model is trained in 16m 40.8s with smaller loss than the previous one, but with an unmoving accuracy of 51.54% in all the 50 epochs (Figure 6).

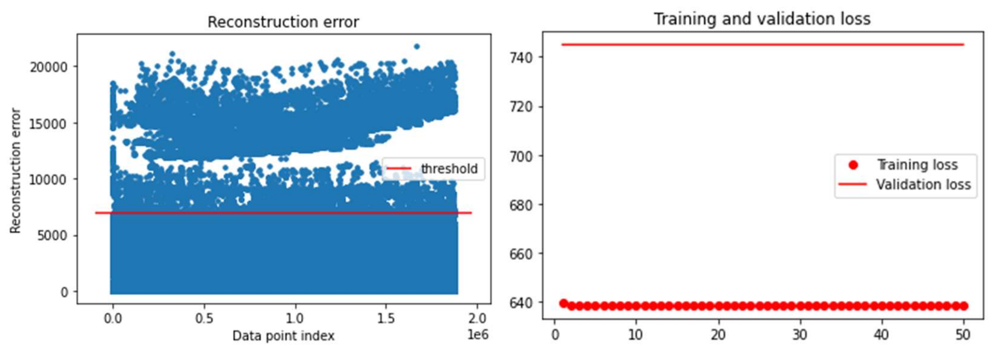

Figure 6. *Autoencoder reconstruction error and training and validation loss for OpenRemote data.*

Nevertheless, there could be an application of autoencoders in our scenario. While they cannot handle the number of features and dependencies we have in our dataset, autoencoders can prove quite useful when looking at a single sensor data. For example, if we take the light sensor in the Motion Sensor Package. From the data description we know that while the light sensor can report values from 0 to 100, usually we observe values between 0 and 10 because of the cover and stickers on the sensor package. The only spikes in the value occur when the lid is removed to replace the battery. While we are familiar with this anomaly there can be many outliers like this one, that have no meaningful information and there is no need to use them when training any model as they just worsen the quality of the data and results. For this purpose, we can train an autoencoder per sensor, that can quickly judge if the recorded data is an anomaly from the standard recordings. For the autoencoder itself we change the loss function from mean squared error to binary crossentropy, the learning rate to 0.0000001 and we train 50 epochs in batches of size 256. The resulting autoencoder is trained in 17.3s and has -0.07 loss with 100% accuracy.

This approach for anomaly detection is not expected to bear meaningful results in the context of recognizing the relationship between different ADLs, as each record is processed independently, which is clearly seen from the preliminary training results. However, we showed that it could recognize anomalies on the technical level for the sensors we are using. and it can be used as a pre-processing to other algorithms to clean up any noise in the data such as hardware malfunctions. To improve on this approach to neural networks in the next section we look at LSTM, which is better equipped to handle multivariate data.

4.1.2 LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that aims to overcome the limitations of traditional neural networks in capturing long-term dependencies in sequential data. By definition this makes it a more suitable approach than the autoencoders for the purpose of this work. LSTM units are composed of a memory cell, input gate, output gate and a forget gate. The memory cell remembers values over arbitrary time intervals, the forget gates filter and discard pieced of the input information for each iteration and the output gate decides which information from the current state will comprise the final output. This selectivity in the output allows LSTM to maintain useful, long-term dependencies and make, predictions in the current and future time-steps. This is why they find applications in anomaly detection (Lindemann, B. et al., 2021).

For the architecture of the model, we have 2 encoding layers each, a repeat vector that replicates the encoded features and returns the data into a 3-dimensional array that serves as input for the decoding layers and as such bridge the encoding and decoding. Followed by 2 decoding layers and finally we have a TimeDistributed layer. The parameter n_features is 36 in our case and the timesteps is our lookback, which is 5. We use the adam optimiser with a learning rate of 0.001, train 100 epochs with a batch size of 256. The training time for the hh101 dataset is 24m 47.9s with accuracy of 0.9162, loss 0.0194 and validation loss 0.0189. We can also plot the change in the loss over the epochs (Figure 7).

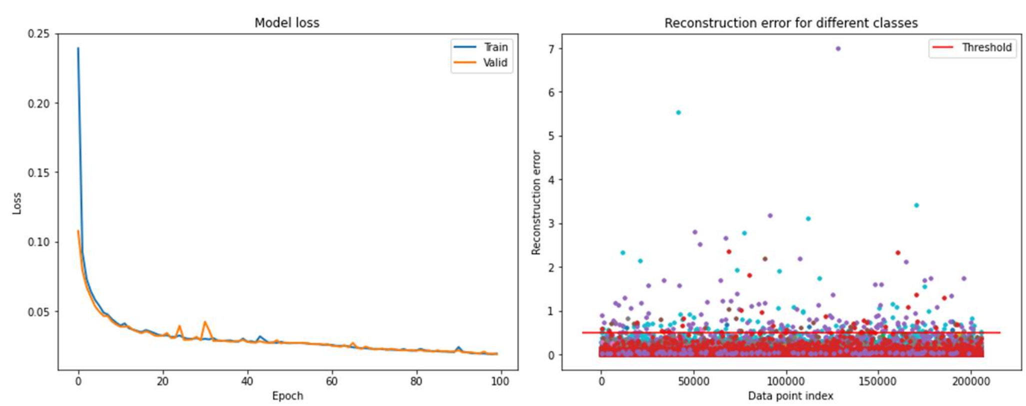

Figure 7. *Model loss and reconstruction of LSTM for hh101*

While a batch size of 32 and learning rate of 0.0001 training takes 117m 18.1s, which is 5 times the training time of a batch size 256. It poses a small improvement in the results, with accuracy 0.9171, which is 0.0050 points better, a loss of 0.0229 that is 0.0017 less and a similar deviation with validation accuracy of 0.9149 compared to 0.9144, and validation loss of 0.0229 compared to 0.0252. This is not a considerable improvement that would justify this much computing time and resources.

4.2 Association Rule Mining

Association Rule Mining (ARM) derives meaningful observations of frequently occurring patterns, correlations, or associations from large datasets, which is why it finds wide application in data mining (Saxena, A. & Rajpoot, V., 2021). We can use the same idea of predicting customer's behaviour in predicting the relationships between different daily activities, part of a routine. This could allow us to understand variations within day, for example the standard routine for Sundays could be very different from the rest of the week. When training a neural network to do the same, we would need to analyse the data before training and separate the Sundays in a separate model, which would not scale in a low-touch approach to assisted living. In addition, Association Rule Mining has also been applied in the field of anomaly detection (Ms. Gargi Joshi, 2014), meaning we can narrow the problem in the same manner as for Neural Networks. The most common ARM algorithms in those spaces are Apriori and FPGrowth and this is what we focus on in this paper.

Association Rule Mining algorithms, expect a list of transactions, containing a sub-list of transactions, based on which correlation is to be derived on the common items and dependencies in each transaction. Obviously, our sensor data does not fit the pattern. Let us consider again the problem we are trying to solve, we are looking for behavior patterns in elderly people, that are prone to abiding by a set routine and any deviations can be treated as a cause for concern as they can be indicative or a harmful to the person episode, as can be the case with dementia. From a behavioral perspective in this case, we can consider two aspects of recognizing behavior, one is on the detailed level of how routines are performed, for example we can break down the dataset based on the subset of sensors that are triggered when <activity> is performed. Long term this provides us with little useful information, because while we are able to recognize that the person has correctly taken their medication, we can't recognize that they have done this for the second time in the past 30 minutes. With that in mind a more reasonable split of the data is based on days. In this case we are looking at the daily routine of a person which fits more to the natural rhythm of routines. We can also easily recognize patterned behaviors that only occur once or twice a week and note them down as a rule with sufficient support. With the daily approach we are of course not able to compare the weekly spread of activities, for example if the monitored person routinely goes for a weekly rehabilitation appointment at the same time, we are not able to recognize if they have missed it completely in a week or attempted to go several times in a week. We can consider additional enhancements in this direction in future works. In this work we are attempting to construct a daily list of timed activities, looking like:

```
[['Other_Activity_02:00:00',
  'Sleep_04:00:00',
  'Sleep_05:00:00',
  'Sleep_06:00:00',
  'Sleep_07:00:00',
  'Other_Activity_07:00:00',
  'Toilet_07:00:00',
  'Cook_Breakfast_07:00:00',
  'Morning_Meds_07:00:00',
  'Dress_07:00:00',
  'Work_07:00:00',
  'Work_08:00:00',
...
  'Other_Activity_22:00:00',
  'Watch_TV_22:00:00',
  'Personal_Hygiene_22:00:00',
  'Sleep_22:00:00',
  'Other_Activity_23:00:00']]
```

4.2.1 Apriori

Apriori is an algorithm, first introduced by Agrawal and Srikant in 1994, used in data mining and machine learning for association rule learning, based on relational databases. It's designed to operate on transactional data and each transaction is seen as a set of items, referred to as itemset. It then identifies frequent itemsets in a dataset and generates association rules based on their occurrence. Candidate generation in Apriori is a "bottom up" approach of extending frequent subsets one item at a time, followed by testing groups of candidates against the data. The algorithm terminates when no further successful extensions are found. Apriori uses a bread-first search and a Hash tree structure to count candidate itemsets. For this type of candidate generation, it's clear that the algorithm has many scans of the database and the number of subsets generated is large. In addition, the time complexity of the algorithm is exponential.

Nevertheless, Apriori is widely used in market basket analysis, where it helps identify patterns and associations in customer purchasing behaviour. Considering we are looking at behavioural patterns in daily life, if we represent the collected activities as transactions, we can attempt to use Apriori for behavioural rule generation.

With the full dataset containing also the Other_Activity ADLs, Apyori times out and can't produce any meaningful result. On the other hand, with the reduced version, where we omit the Other_Activity category we derive exactly 600 rules for the hh105 dataset with support 0.2, confidence 0.8 and lift of 3. For example, we can observe straightforward rules, such as the one bellow. Where we can derive that if the person is working at the table at 16:00, they are also likely to be working at the table also at 17:00. Similarly, we can see some more examples of the rules such as, a person who comes back from an outing at 12:00 is likely to be sleeping on the couch at 13:00. In the 3rd relationship we see that part of the normal routine of the recorded person is starting the day with sleeping until 7 o'clock then getting dressed, eating and cooking breakfast at 8, and ending the day with watching TV for at least two hours around 21:00 and 22:00 o'clock.

` Efficient Apriori finds 114253 rules in a second with the same parameters of support 0.2 and confidence of 0.8. For example, it seems that this person usually goes to the toilet at 7 in the morning and is in bed by 23:00. Occasionally the person leaves for less than an hour or someone visits them in the morning around 10 o'clock, seen by the low support this is not a daily event but still happens often enough to be picked up. We of course pick up on the natural relationship between sensor events such as, if the person is returning from the bathroom at 6 in the morning, they are going back to bed and are asleep by

1. `{Toilet_07:00:00} -> {Sleep_23:00:00} (conf: 0.800, supp: 0.300, lift: 1.143, conv: 1.500)`
2. `{Enter_Home_10:00:00} -> {Leave_Home_10:00:00} (conf: 1.000, supp: 0.225, lift: 1.818, conv: 450000000.000)`
3. `{Enter_Home_12:00:00} -> {Personal_Hygiene_22:00:00} (conf: 0.833, supp: 0.250, lift: 1.852, conv: 3.300)`
4. `{Bed_Toilet_Transition_06:00:00} -> {Sleep_07:00:00} (conf: 1.000, supp: 0.225, lift: 1.111, conv: 100000000.000)`

With that in mind, let's try to see if efficient Apriori finds the same routines and what are the additional rules discovered, that would explain the increase in discovered rules volume between Apriori and efficient Apriori. For 0.3 support in efficient Apriori we can see the same rule has been derived: {Cook_Breakfast_08:00:00} -> {Dress_08:00:00}, however the opposite rule of apyori can be seen: {Cook_Breakfast_07:00:00} -> {Sleep_07:00:00}, considering our hourly aggregation, we can be sure how accurate these particular rules are. As with Apyori we don't see the behaviour from before. Let's look for the same in the 0.2 support efficient Apriori. Despite the larder volume we can immediately see the same pattern of early morning waking up, however there are several more rules that correlate with the sleeping. The ones following our 'backwards' understanding would be filtered, for example {Bed_Toilet_Transition_06:00:00} -> {Sleep_01:00:00} (conf: 0.889, supp: 0.200, lift: 1.368, conv: 3.150).

```
{Bed_Toilet_Transition_03:00:00} -> {Sleep_03:00:00} (conf: 1.000, supp: 0.200, lift: 1.600, conv: 375000000.000)
{Bed_Toilet_Transition_03:00:00} -> {Sleep_06:00:00} (conf: 1.000, supp: 0.200, lift: 1.143, conv: 125000000.000)
{Bed_Toilet_Transition_03:00:00} -> {Sleep_07:00:00} (conf: 1.000, supp: 0.200, lift: 1.111, conv: 100000000.000)
{Bed_Toilet_Transition_05:00:00} -> {Sleep_05:00:00} (conf: 1.000, supp: 0.200, lift: 1.333, conv: 250000000.000)
{Bed_Toilet_Transition_05:00:00} -> {Sleep_07:00:00} (conf: 1.000, supp: 0.200, lift: 1.111, conv: 100000000.000)
{Bed_Toilet_Transition_06:00:00} -> {Sleep_01:00:00} (conf: 0.889, supp: 0.200, lift: 1.368, conv: 3.150)
{Bed_Toilet_Transition_06:00:00} -> {Sleep_06:00:00} (conf: 1.000, supp: 0.225, lift: 1.143, conv: 125000000.000)
{Bed_Toilet_Transition_06:00:00} -> {Sleep_07:00:00} (conf: 1.000, supp: 0.225, lift: 1.111, conv: 100000000.000)
{Bed_Toilet_Transition_06:00:00} -> {Sleep_23:00:00} (conf: 0.889, supp: 0.200, lift: 1.270, conv: 2.700)
{Bed_Toilet_Transition_07:00:00} -> {Sleep_06:00:00} (conf: 1.000, supp: 0.200, lift: 1.143, conv: 125000000.000)
{Bed_Toilet_Transition_07:00:00} -> {Sleep_07:00:00} (conf: 1.000, supp: 0.200, lift: 1.111, conv: 100000000.000)
```

Figure 8. *Efficient Apriori discovered pattern of early morning waking up*

Looking at the rule quality and processing times, the preferred Apriori implementation is the efficient Apriori, with the 0.2 support and 0.7. Even though we have more rules as a result, the processing time gives us also sufficient opportunity to filter the rules that derive insights for the past and still be left with a good ruleset in a reasonable amount of time (Figure 8).

4.2.2 FP Growth

Frequent Pattern Growth (FP Growth) is a Data Mining algorithm, aimed at deriving association rules in large itemsets first introduced in the early 2000 by Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao (Han, J. et. al., 2004). The foundational concept of the algorithm is the generation of a Frequent Pattern Tree (FP Tree), which is a compressed representation of the itemset database, that also records the association between itemsets. This allows FP Growth to overcome the shortcomings of Apriori and handle large datasets as they are stored in a compact tree which we can traverse efficiently. As with Apriori, let's compare the two types of datasets, first the reduced one, where we remove all 'Other_Activity' categories and then the full day dataset with them present. Also, we use the same parameters as for Apriori, meaning support of 0.2 and confidence of 0.8 and the same dataset – hh105.

For the reduced dataset, the total running time is 59.2s, and we derive a total of 88918 rules. Some examples include the same pattern we saw also with Apriori, that there seems to be an occasional leaving or visiting of the home, that happens within the same hour in the morning at 10:00, but also coupled with several variations of the sleeping and morning routine, see examples 1-3. Also, we can see more rules relating to the persons routine, such as 11 o'clock is a time they commonly go out, and also on the evenings where they have dinner around 18:00 they are also likely to be in bed sleeping at 23:00.

1. `[{'Enter_Home_10:00:00', 'Morning_Meds_07:00:00', 'Sleep_07:00:00'}, {'Leave_Home_10:00:00'}, 1.0]`
2. `[{'Enter_Home_10:00:00', 'Sleep_07:00:00'}, {'Leave_Home_10:00:00'}, 1.0]`
3. `[{'Enter_Home_10:00:00', 'Sleep_07:00:00'}, {'Leave_Home_10:00:00', 'Morning_Meds_07:00:00'}, 1.0]`
4. `[{'Relax_11:00:00', 'Sleep_06:00:00', 'Sleep_07:00:00'}, {'Leave_Home_11:00:00'}, 0.8888888888888888]`
5. `[{'Cook_Dinner_18:00:00', 'Eat_Dinner_18:00:00', 'Sleep_03:00:00'}, {'Sleep_23:00:00'}, 1.0]`

4.3 Comparison of Neural Network and Association Rule Mining algorithms for human behaviour analysis

Until now for the training and evaluation of the different approaches we were using a small subset of 1-3 datasets from the CASAS dataset. With this data format we have around 31 distinct datasets and houses. Each dataset contains measurements for 3-6 months in households with a single person or 2 people. With the description of each dataset, we can't immediately identify the number of people that were living in the home. We can assume a second residents in larger homes, like the hh121 dataset, see sensor map bellow. Here we see not only the size of the apartment but also two distinct zones of activity from the two sides of the bed. However, even that is purely an assumption and it could be the case that a single resident lives here. With that in mind it is possible that we select a dataset, where no routine can be found not because of the algorithms themselves, but because we are recording more than one person. Of course, we can't assume a routine is present even in single-resident households.

Let us compare some of the datasets. In Figure 9 bellow we can see several sensor maps for the datasets, namely the first 15 datasets. We can see that some homes are very similar in their layout and others differ greatly. In principle when we categorize activities, we don't record the sensors that provided the input, meaning that if there are two bathrooms and only one resident, showering in either of them would be recoded as Personal_Hygiene. On the other hand, when using feature vectors, this would be considered as two separate events. Such distinction would only make sense in select few cases and is more harmful to the general application case. From that perspective we can anticipate poorer results from those homes.

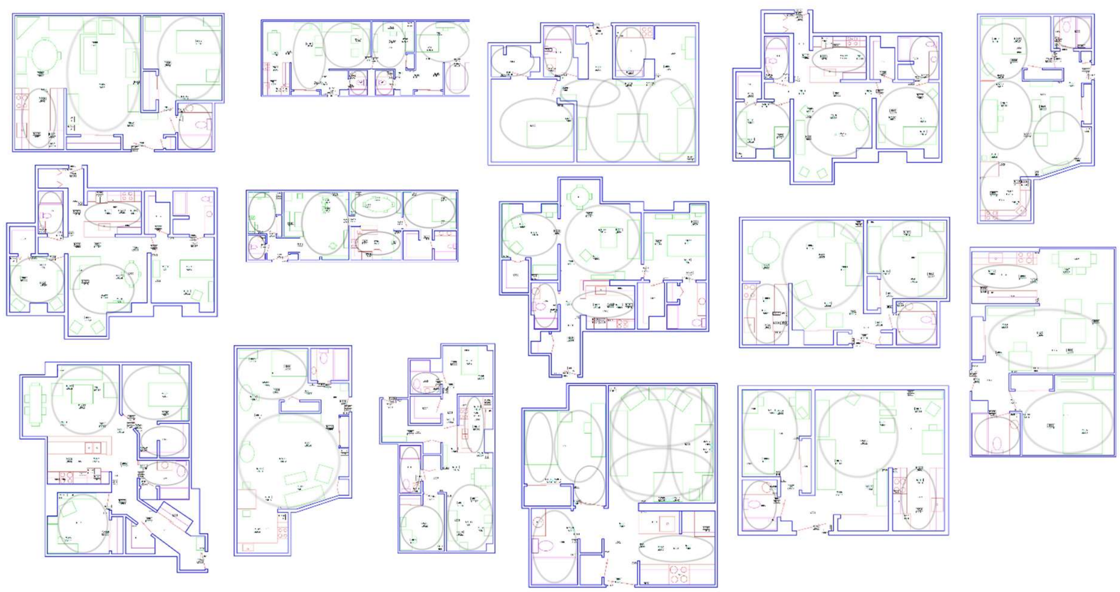

Figure 9. *Sensor maps for datasets hh101-hh115*

In order to confirm that expectation, let's briefly look at the size of the different apartments, the total number of rooms they have and any duplications, such as two and more bedrooms or bathrooms. In the table below we have summarized that information, we count the number of rooms, where we have sensors measuring for ADLs, for example kitchen, living room, bathroom, bedroom, walk-in closets, office, etc. Not all homes have all types of rooms. What is also of interest is the volume of the data which we are using for training, for this purpose in the table below we can see also the total number of raw records in each dataset as well as the size of each feature vector.

dataset	Number of rooms	Duplicate rooms	Number of records	Feature Vector Size	Collection Period
hh101	5	no	1875256	321428	2012.07.18 - 2013.07.25
hh102	9	yes	6472396	407481	2011.06.15 - 2014.04.16
hh103	6	no	8481133	164809	2011.06.15 - 2017.05.10
hh104	9	yes	6534302	477988	2011.06.15 - 2013.09.04
hh105	5	no	11167384	222481	2011.06.15 -2017.03.22
hh106	9	yes	6364167	259908	2011.06.15 - 2015.01.05
hh107	8	yes	3340529	291133	2012.07.20 - 2013.07.25
hh108	9	yes	12550848	357685	2011.06.15 - 2017.05.11
hh109	5	no	15985956	564452	2011.06.15 - 2016.01.05
hh110	5	no	156689	136716	2011.06.15 - 2011.07.19
hh111	7	no	14040229	351324	2011.06.15 - 2017.05.11
hh112	5	no	826174	660403	2011.06.15 - 2011.10.31
hh113	8	yes	3213217	3190818	2011.06.15 - 2012.11.05
hh114	5	no	11688878	192514	2011.06.15 - 2017.05.06
hh115	5	no	2240010	2139155	2011.06.15 - 2012.05.25

Table 1. *Dataset composition and size for hh101 – hh115*

4.3.1 Neural Networks

Let's first look at autoencoders, in the Results of the training section we looked at several approaches to autoencoders, while the best accuracy was achieved using the feature vectors, this came at a cost of a very high loss. Since in that case we always see a 100% accuracy, for the purpose of the comparison it makes sense to look at the training results of an autoencoder, using the raw sensor data and training for 50 epochs with batch size of 256. Overall, no model managed to break above the 50% accuracy mark, with very high loss on the training and validation.

dataset	loss	accuracy	val_loss	val_accuracy
hh101	1089.67	0.424634	1230.009	0.372507
hh102	1246.745	0.394556	1007.881	0.50369
hh103	1118.527	0.543028	1333.84	0.396355
hh104	1385.82	0.403193	1490.415	0.295929
hh105	2115.238	0.255698	2086.841	0.217718
hh106	2105.554	0.225117	3574.163	0.194145
hh107	1612.309	0.286165	1739.301	0.256039
hh108	2666.369	0.162554	3236.166	0.09343
hh109	1383.816	0.309845	2259.949	0.260966
hh110	833.9977	0.392737	872.1605	0.301784
hh111	3524.788	0.1733	3848.066	0.091387
hh112	1163.672	0.416863	981.7435	0.418019
hh113	1444.335	0.346073	1411.587	0.330933
hh114	1228.272	0.425059	1654.324	0.285553
hh115	973.3888	0.489532	1072.799	0.453623
hh116	725.4113	0.499146	1663.438	0.246305
hh117	917.3568	0.477592	1199.039	0.329247

Table 2. *Training results of Autoencoder with mean square error*

If we compare the training result table with the dataset composition table, we can see some expected trends. For example, our model for the hh111 dataset is almost the worst in accuracy and loss, but this is also our largest dataset and large datasets are not as easily handled by autoencoders. Naturally the smallest dataset hh15 shows the smallest loss, and the highest accuracy, this is not a trend in the other datasets as the second smallest hh110 falls below the average on accuracy and the lowest accuracy is observed in hh108, which is the 3rd largest dataset. To summarize while the poor performance of the autoencoder model is not unexpected due to the nature of the dataset and the problem area we are addressing, we can use the results below as a baseline to analyse more complex RNNs, like LSTM.

With that in mind, let us do a similar comparison for LSTM, here we use the feature vector approach as it had a clear advantage. We set the batch size at 32, learning rate at 0.0001 and we train for 50 epochs using the mean squared error loss function. It performs significantly better than the autoencoder with accuracy usually averaging around 88% and loss around 0.03, as you can see in the table below.

dataset	loss	accuracy	val_loss	val_accuracy	time
hh101	0.0284	0.9179	0.0284	0.9181	78m 38.2s
hh102	0.0297	0.8836	0.0302	0.886	95m 43.1s
hh103	0.033	0.8811	0.0341	0.8765	n/a
hh104	0.0283	0.8933	0.0294	0.8945	n/a
hh105	0.0267	0.9139	0.0285	0.9162	n/a
hh106	0.0371	0.8796	0.0383	0.8834	36m 10.4s
hh107	0.0465	0.8677	0.0478	0.8704	38m 4.9s
hh108	0.0304	0.8644	0.0316	0.8647	77m 28.0s
hh109	0.0341	0.8807	0.0356	0.8766	104m 6.4s
hh110	0.0453	0.8585	0.0493	0.8611	28m 13.5s
hh111	0.0298	0.8769	0.0294	0.8817	76m 3.0s
hh112	0.039	0.8581	0.039	0.8599	138m 17.8s
hh113	0.0275	0.9181	0.0243	0.918	206m 1.1s
hh114	0.035	0.9004	0.0356	0.9015	39m 46.3s
hh115	0.0251	0.9027	0.0241	0.9007	511m 14.6s

Table 3. *Training results of LSTM on hh101 – hh115*

LSTM doesn't have the same limitations for large datasets, that the autoencoder has. We can see that when we compare the training results for hh111. This is the largest dataset and had the highest loss and lowest accuracy in the autoencoder model. For LSTM it has one of the lowest losses and performs comparable well to the other datasets. Unlike the autoencoder we can see a more consistent result in the accuracy and loss of the models on new datasets, which would make this approach more consistent when applied to our AAL use case. The hh110 dataset had decent results with the autoencoder model, due to its small size. In the LSTM case we can see that it has the lowest accuracy and fasted processing time, in line with our expectation. As we see larger datasets support better LSTM behavioural model. Which would mean that the longer our data collecting AAL system is in use, the better the behaviour predictions would be.

Looking at the training times, autoencoders training on the full sensor dataset have a running time of 30 minutes to an hour, depending on the dataset. LSTM shows similar times for a batch size of 256, with reducing the batch size there is no significant improvement in the accuracy of the model, but we observe that the processing time is multiplied with a factor of the batch size decrease. For this selection of data, the experiments found an optimal trade-off between the results and training time in the parameters defined above with batch size 256 and 50 epochs. In the next section we similarly analyse the association rule mining approaches.

4.3.2 Association Rule Mining

We looked at the neural network algorithm performance across the datasets hh101 to hh115. In this section we make a similar comparison for the Association Rule Mining approaches. In the Results of Training section for Apriori, we already identified that efficient Apriori has just as good of a quality of the derived rules but at a fraction of the time for large datasets. Therefore, in this section we are using the efficient Apriori implementation, when evaluating the Apriori performance and results.

dataset	Number of Rules	time
hh101	15561908	2m 17.3s
hh102	1793282	15.0s
hh103	15171293	2m 25.0s
hh104	566482	4.7s
hh105	154048	1.6s
hh106	538255	4.7s
hh107	n/a	n/a
hh108	28263929	4m 44.3s
hh109	n/a	n/a
hh110	19419920	4m 1.6s
hh111	6738776	1m 19.1s
hh112	2692530	33.7s
hh113	3774789	1m 9.1s
hh114	42276	1.0s
hh115	20985389	6m 23.8s

Table 4. *Training results of Apriori on hh101 – hh115*

Let us look at the results of FPGrowth, in table 8 you can see the volume of rules per dataset that was derived. Training is slower than Apriori, comparable in time only for hh103 and hh105, some datasets take a few minutes, other hours and for hh107 and hh108 the resources on the training machine were not sufficient leading to a timeout and memory crash. We can observe the same pitfalls as with Apriori, because of the data processing, which aggregates events on an hourly basis. For each dataset we see the same routines identified by both algorithms, but also rules that are unique to each approach.

dataset	FPGrowth	time
hh101	554151	9m 42.5s
hh102	1035194	22m 33.0s
hh103	929094	9m 34.4s
hh104	387020	6m 22.1s
hh105	88918	40.2s
hh106	372456	3m 2.5s
hh107	n/a	n/a
hh108	n/a	n/a
hh109	n/a	n/a
hh110	36416334	158m 26.5s
hh111	5686350	71m 34.4s
hh112	2873711	71 m 50.4s
hh113	2942661	593m 29.3s
hh114	24560	9.8s
hh115	n/a	n/a

Table 5. *Training results of FPGrowth on hh101 – hh115*

In conclusion, autoencoders cannot handle the multi-dimensional complexity of the problem space. Nevertheless, it is a very fast and reliable approach to detect sensor malfunctions and outliers as it takes less than a minute to train and has goof results. Sensor malfunctions should be removed from the data, as we are interested only in behavioural anomalies in a person's daily life and not outlier device recordings. So, while not sufficient on its own it would be a good pre-processing companion to either of the alternative algorithms.

LSTM is capable of understanding a routine and detecting outliers. However, due to our premise of creating a general approach to training with pre-processing that is consistent for every dataset and not additionally cleaned up per person, LSTM cannot achieve accuracy above 90% on average, which leads to many false positives and false negatives. We are working with a sliding window of 30 events, therefore we are unable to identify which of the recorded events raised the outlier exception. Therefore, since there is no way to backtrace the cause of the outlier alert or a way to explain why it was raised, we are unable to judge each raised event and mitigate the errors. Additionally, the 30-event sliding window restricts the model's ability to find correlation between behaviors throughout the day.

In the case of Apriori and FPGrowth, both algorithms manage to produce reasonable rulesets, that have a foundational understanding of the existing routines. Naturally, they are not perfect and miss certain behaviors when constructing the ruleset, which raises many more outlier events. Unlike the Neural Networks, here we have traceability and understanding of the raised event, based on the event that triggered it and it's fairly easy to correct and add or remove a rule that is not reflecting correctly the reality of that person's routine. This approach would be also easier to integrate with our OpenRemote prototype as we already have a concept of rules and a rule database there, which would also. In general, the efficient Apriori implementation manages to find more and better rules compared to FPGrowth, but further testing and research is needed to confirm that there are no cases where the routine defined by FPGrowth is equal to the one in Apriori.

While neither approach is perfect, using ARM-based algorithms for outlier detection in the person's behavior is the most cost-aware option, that also has understandable results and easily mitigates technical outliers without the need of pre-processing. It is also fitting and easy to integrate with the OpenRemote prototype. Between the two algorithms Apriori is definitely quicker and has just as good result generation, nevertheless we need to further evaluate if in some cases the slower FPGrowth finds routines that Apriori misses.

5 CONCLUSION

In this work we challenged the common gaps research on general purpose Ambient Assisted Living (AAL) Systems identifies. Namely ensuring data privacy and security in a scalable system, factors that have hindered the widespread availability and adoption of AAL Systems. By reusing the state-of-the-art middleware, that is available and supported in open-source middleware platforms we have created a cost-aware prototype. Identifying the functional requirements of GDPR we were able to propose an implementation that abides by GDPR and derivatively enables data privacy for its users.

Similarly, we have seen that machine learning algorithms have an application in understanding human behaviour in the context of daily activities. In the scope of a cost aware AAL system we rely on the already available sensors and smart devices to create a data stream of events that point to the daily activities of the person using the system. This means we have no control over the available data sources and sensors and cannot tailor the data as much as we would need to get the optimal training results. Neural networks are very affected by this constraint, as they can't achieve a good enough accuracy to be applied in a productive manner in an AAL system, on the other hand associated-rule-based models are flexible on this point, as their human-readable predictions and output allow for an end user in the role of a caretaker to understand them and in turn ignore the false positives.

Combining the prototype and the models we propose a future proof AAL System, that could find a widespread application, due to its powerful customizations, low cost of maintenance and user onboarding and good foundation for real use applications, due to its compliance with foundational principles such as data privacy and enablement of data security.

5.1 Discussion

The first goal we set in the form of a research question was the creation of a cost aware Ambient Assisted Living (AAL) System and in this thesis, we have created a cost-conscious prototype, when compared to its predecessors:

- The system is based on an open-source middleware, which outsources most of the needed support for keeping the system up-to date with emerging technology and best practices, as required by GDPR especially for security and data protection. This also reduces the cost of developing the system, the technical supportability of more and diverse device connectivity mechanisms and protocols and reduces the cost of support of the system, as OpenRemote has a highly active user and developer community which allows the quick identification and resolution of bugs and vulnerabilities in a productive large-scale usage of the system.
- The system prototype is running on as a set of docker containers and because of that the productive system can be easily installed in a matter of minutes, as the personalization of the realm is achieved in the docker compose file, containing the container configuration. The same allows the quick and easy move or regular backup of the system. In addition, we can easily introduce additional resilience of the systems, by using the Kubernetes best practices and using a reconciliation loop to ensure the containers are always running healthy and available.
- For the Machine Learning training, there is a threshold of resources that we can't optimise beyond. In the case of the system prototype, we have a good trade-off between the usefulness of the model and the compute resources, by using Association Rule Mining algorithms, that are easily integrated with OpenRemote's architecture and a very optimized implementation of the Apriori algorithm. In contrast the evaluated neural networks consumed more than double the resources in some cases in pre-processing and training.
- The same cost optimisation logic is present also when re-training the model. The cost of doing so for the Neural Networks is much higher than the cost for retraining the ARM-based algorithms, not only because of the consumed resources, but also because the OpenRemote prototype we designed allows the caretaker to manually correct or add new rules, which allows the model to be retrained less often and maintain a higher usefulness and accuracy of the outlier alerts for longer times.
- The cost of the initial onboarding of the system is reduced by the use of already existing smart home devices in the person's home. Since OpenRemote has an architecture that supports the most common connectivity protocols it is possible in principle to connect existing devices and the initial investment of purchasing, setting up and connecting sensors and devices is lowered.
- OpenRemote has a realm concept which allows one instance of the middleware to be shared among several users in nearby proximity, depending on the type of sensors. For wireless sensors we can use a local hub to collect and transport data to the central instance. This in turn allows the cost of the system and the computational resources to be split among several users.

Secondly, among the many existing AAL prototypes, we saw that there are two main questions left open: security and data privacy, which we theorised could be one of the reasons why there isn't such a wide adoption and easily available to the general public AAL system. In this work we did not look at security beyond the requirements of GDPR and rather consumed the results from the middleware platform. However, we put special focus on data privacy and created a system that can confidently claim is GDPR-compliant. From our GDPR research, we derived 15 main functional requirements, that we abided to.

- We highlighted the importance of the proper involvement of a legal team in the data privacy compliance. They are the ones, that determine whether the processing is fair, legal, and transparent; what are the lawful reasons for processing; for which data and to what extend is each data subject request (DSR) applicable; are there results of automated processing personal data and as such subject to the GDPR, etc. And this is highly dependent on the conditions in which a system is released and productized. In this thesis we are striving to prove the technical feasibility and compliance of such a system, hence we have not engaged with a legal team, which would be a prerequisite to make the system productively available
- Data processing in GDPR is any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means. To ensure the safeguard of that, data needs to be minimized, meaning processing is adequate, relevant, and limited to what is necessary. In the prototype we ensure data minimization on all levels of data transportation. At collection time we only record the minimally necessary information collected by the device. For the purpose of the alerts only a human readable statement is sent, with the necessary information and decoupled from the actual sensors and data. The machine learning model has direct access only to the attribute data table in an isolated network. Additionally, data is stored in a decoupled method, as the attribute details are stored separately form the values and both are disconnected from the rule database.
- Since the data is decoupled in case the person is correctly evoking their right to erasure, as defined by the legal team, the actual clean-up is a matter of a joint select statement and an update of the system backups, to ensure erased data can't be restored. Similarly, we have several ways to enable restricted processing, as described in GDPR, the easiest of which is to let the data pass through the system without recording it.
- For the topic of security, authorization, and authentication GDPR is deliberately vague, as the concepts evolve continuously. Instead, it requires that systems use 'state of the art' technology. From that perspective we benefit from relying on an open-source middleware. As long as we have an active community and widespread productive applications, there will be a stream of innovation and technology adoption. We went over the 'state of the art' at the time of implementation in the chapter dedicated to

GDPR compliance. Nevertheless, to claim a truly secure system we would need to go deeper and analyse each device and connectivity type, all methods of communication between systems and devices, methods of authorization and authentication, analyse vulnerabilities, invest into threat modelling, etc.

- Additionally, the selected service to send and receive instant push notifications and messages Pushsafer has been developed in the European Union, ensuring that we can also abide by the limitation that data in the EU is not accessible or processed from the outside of the EU. Pushsafer is also focused on ensuring data protection, which means the integration does not introduce new limitations.
- On the setup of backups and high availability setups, we can rely on the already established Docker and container management best practices. We can go for a full Kubernetes setup, use docker swarm, create custom solutions, etc. The approach would depend on the volume of the data and users in the instance of the system and the prognosed reliability of each instance of the system.
- In GDPR data is accessible and distributed on a 'need-to-know'-basis. In our AAL system there can also be medical data present, which needs to be further protected. Based on the existing role and permission-based paradigms of OpenRemote we have created an identity management system, which categorizes users based on their relationship with the main user of the system and restricts their access to different subsets of the data.
- GDPR recitals state that a data subject has the right to "an explanation of the decision reached after [algorithmic] assessment", which we can achieve by using rule-based models as opposed to neural networks, and this has also impacted the preferred ML approach.
- When discussing the implication of GDPR on the machine learning models a main concern is the right to erasure which poses the question should the data used for training the model be removed from the model as well and is it possible for the machine learning model to expose personal or medical data to unauthorized people. For our prototype we have simplified both answers by training a model per person, as opposed to a generic model that uses every user's data jointly. This is not only because of GDPR but also because of the nature of the problem space, we are looking for highly specific and personalized routine behavioural patterns and their deviations. On the topic of unauthorized data exposure, the model itself only interacts with users by raising alerts and events, which it has considered an anomaly. The details behind the reason of the raised alert are only available per request and for the Apriori implementation.

Lastly, in the beginning we asked if the application of machine learning algorithms, trained on the collected data would have a positive impact on the system functionality, without compromising the cost-awareness of the system. This translated into calculating the trade-off between the model correctness and usefulness and the cost of training, maintenance and resources for it.

- We reconfirmed the application of autoencoder neural network algorithms for the recognition of outlier technical behaviour in a timeseries stream of numeric sensor data. We proposed that an autoencoder is used for the pre-processing of all data to ensure that the technical outliers in the sensor data are separated from the person's behavioural outliers we aim to identify.
- We showed that it is possible to apply neural networks like LSTM for the recognition of patterns in human behaviour and as a result detect outliers in it. Our experiments showed that a usable LSTM model requires pre-processing, customization of the parameters, clean-up of the data and general time investment beyond what we are willing to invest per person in a cost aware AAL system. Nevertheless, even without that investment, the LSTM model showed decent understanding of events that naturally follow one another. One limitation of that is the recognition of patterns that span across the full day, week, month etc. This would be possible to achieve via LSTM with additional modification of the pre-processing and the algorithm itself.
- Similarly, we identified and exemplified the application of association rule mining algorithms for the same problem: the identification of outliers in a human's behaviour based on pre-recognized daily routines. In this case stored as rules. We compared the Apriori and FPGrowth algorithm on a variety of datasets to judge the quality of results and applicability and showed that we are able to derive a daily routine connecting events throughout the day in the form of association rules.
- To achieve the resulting routine, we proposed a data preparation for ARM algorithms, that transforms all daily events in a transaction, that can be used by Apriori or FPGrowth in this case to recognise the commonalities between days.
- In the course of the research and implementation we found and validated an optimised Apriori implementation, that had better results than the standard FPGrowth implementation.
- In our case where we are looking for usable results in a lower cost of training and model lifecycle management, we compared the neural network algorithms to the association rule mining ones to evaluate the cost/usefulness trade-off and derived that the better option with our prototype and use case in mind is the usage of association rule mining and the efficient Apriori implementation of Apriori. It allows a human to mitigate the errors in the raised alerts, due to its explainability and also the present confidence with which each alert is raised.
- One limitation to recognize in this section is the boundary condition that we look for behaviour patterns occurring in a day, we are ignoring holidays, weekly, monthly and yearly routines. This would need further enhancement of the data pre-processing and

training of both models until an optimum is discovered. Notably, here we also need to consider the cost factor: what is the cost of developing a ML model that recognizes that once a year on his birthday a person has a special routine compared to simply adding it as a rule in a rule database? How do you differentiate between those behaviours and how many years/months of data would you need for that?

Based on the research questions we formulated the following hypothesis: "We can reuse open-source smart home middleware software to create a cost- and data privacy-aware AAL system, extend it with machine learning algorithms in a useful manner and prove that association rule mining (ARM) algorithms can be used for human behavioural recognition and they would be the better choice compared to standard outlier detection approaches for an AAL system as they are overall cheaper, easier to conform to data privacy regulations and they have explainable results.". From the argumentation above we can clearly see that our hypothesis was confirmed as part of this researched with several limitations and corner cases identified. Based on those we can expand the research and further improve on the results; we expand on that in the Future Research section.

5.2 Contributions

Based on the above-described research and results we can formulate this works contributions, separated in three fundamentally overlapping categories:

Scientific Contributions

1. Analytical review of the state of art of the field of AAL Systems, the currently existing types of systems, commonly solved problems, and methodology
2. Analytical review of GDPR and a summarization of the functional requirements an AAL system needs to implement in order to ensure data privacy. The same can easily be translated and applied to different space of systems.
3. Modifications and improvements of machine learning algorithms for the purpose of recognizing human behavior patterns. Comparison of performance, accuracy and applicability in the proposed AAL system of neural network and association rule mining algorithms.
4. Development of a methodology to general AAL system creation.
5. Discovery and validated recognition of an optimized Apriori implementation, that provides faster, and as accurate results compared to FPGrowth for behavioural pattern recognition.
6. Proposed method for the transformation of recoded as timeseries daily activities into labelled daily transactions for the purpose of rule generation.

Scientific-Applied Contributions

1. Proposal for a machine learning enhanced AAL system, that uses already existing sensors and devices in a person's home to create a model that recognizes human behavioral routines and can alert to outliers to them.
2. Proposed and implemented architecture of an AAL system, enhanced with machine learning algorithms that ensures data privacy compliance by running as a containerized solution in an isolated network.
3. Experiments on approximately 15 datasets from the CASAS collection of datasets, resulting in a conclusion that ARM algorithms are more cost-conscious and easier to maintain when applied to and AAL system for behavioral pattern recondition.
4. Concept and requirements for the creation of AAL systems focused on the overall cost-effectiveness of implementation, long-term support and cost per person.

Applied Contributions

1. Prototype implementation of an AAL system, enhanced with machine learning models.

Prototype implementing the functional requirements of a GDPR-compliant system.

5.3 Future Research

There are several directions in which we can enhance and develop the current research: The prototype of the system itself, the real-world validation of the results and the improvement of the models:

- In this work we laid the architecture of the system and implementation independent form the machine learning approach which we would then integrate in the system, as at the time we had no clarity which one it would be. Now that we know that this would be the Apriori algorithm it makes sense that we further integrate it into the system. Namely by enhancing Apriori's database with the confidence parameter and integrating the resulting from Apriori database into the OpenRemote one. This should be done in a way that does not override the manually provided rules upon re-running of the algorithm but also does not preserve the outdated rules from previous Apriori iterations.
- The prototype relies on the smart devices already available in the person's home. Nowadays we see a decline in the availability of the data stream from those devices. Smart wearables data is usually only consumed via and application and as of recent that also requires a monthly subscription for the most popular devices. This is counterproductive for the cost-awareness of the system. That is not the case for every type of device and in our prototype, we had no data from a wearable device. Therefore, a logical next step would be to research the most common smart devices in a person's home, classify them and analyse the availability of their input. From this research we could apply the Apriori algorithm and derive what is the minimal number and type of devices that is needed to correctly recognise a routine and train a model on.

- The prototype used pre-recorded data of people of whom we know nothing of. The main target group of our AAL system is elderly people and especially dementia patients, where we assume a routine is present. With that we need to apply the system and Apriori algorithm on data collected for a dementia patient. Significant part of that research would be to also identify which outliers in the behaviour lead to an episode where the person is confused, lost or needs assistance. We can do this with an already available dataset, but in this case the next step would be to enable the devices in a real dementia patient house and validate the initial hypothesis there. A limitation of the final real-person validation would be the fact that at present dementia patients are not independent and are rarely left to take care of themselves, therefore our system would be recording always the behaviours of the patient and their caretaker.
- There are two obvious optimisations in Apriori, that should be explored. One the current Apriori treats all items in the example as equal and in our case, they are sequential. This is an important fact as more than half the time is spent into looking for rules that have results in the past which consumes both training resources and post-processing resources for the clean-up. The second is to create a smarter aggregation window, that can handle events that are recorded close to the top of the hour.
- Similarly, Apriori can be enhanced to cover routines that span over weeks, months and years. Again, we have to be careful of the cost of training an algorithm to recognise rare or exceptional rules, that can be manually added later on. All in all, Apriori shows great potential for optimisation and application in the space of recognizing patterns in human behaviour.

DECLARATION OF AUTHORSHIP

I hereby declare that this dissertation contains original results obtained by me with the support and the assistance of my supervisors. The results, obtained by other scientists, are described in detail, and cited in the bibliography. This dissertation has not been previously submitted for a degree or any other qualification at another University or any other institution.

Signature:

REFERENCES

Akhter Ali, Mohd & Kamraju, M.. (2023). The Economic Consequences of Population Aging Challenges and Policy Implications. 02. 45-50.

Auth, Diana & Leiber, Simone & Leitner, Sigrid. (2023). Contestations in coping with elderly care: an intersectional analysis addressing family caregivers in Germany. European Journal of Politics and Gender. 6. 1-18. 10.1332/251510821X16805286127766.

Bergua, Valérie & Fabrigoule, Colette & Barberger-Gateau, Pascale & Dartigues, Jean-François & Swendsen, Joel & Bouisson, Jean. (2006). Preferences for routines in older people: Associations with cognitive and psychological vulnerability. International journal of geriatric psychiatry. 21. 990-8. 10.1002/gps.1597.

Borsa A, Bejarano G, Ellen M, Bruch J D. (2023) Evaluating trends in private equity ownership and impacts on health outcomes, costs, and quality: systematic review BMJ 2023; 382 :e075244 doi:10.1136/bmj-2023-075244

Du, Jiusheng & Wang, Yu & Feng, Yunchao & Liu, Dingming & Yuan, Wenbin. (2023) Research on care preferences for the elderly and influencing factors in Zhejiang Province: A statistical and geographical approach. Medicine 102(29):p e34374, DOI: 10.1097/MD.0000000000034374

EU General Data Protection Regulation (GDPR): Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), OJ 2016 L 119/1.

Fahey T, Montgomery A A, Barnes J, Protheroe J. (2003) Quality of care for elderly residents in nursing homes and elderly people living at home: controlled observational study BMJ 2003; 326 :580 doi:10.1136/bmj.326.7389.580

Geyer, Johannes and Börsch-Supan, Axel H and Haan, Peter and Perdrix, Elsa. (2023) Long-term Care in Germany. National Bureau of Economic Research.

Ghith, Nermin. (2022). Estimation of the global prevalence of dementia in 2019 and forecasted prevalence in 2050: an analysis for the Global Burden of Disease Study 2019. The Lancet. 7. e105-e125.

Han, Jiawei & Pei, Jian & Yin, Yiwen & Mao, Runying. (2004). Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Min. Knowl. Discov.. 8. 53-87. 10.1023/B:DAMI.0000005258.31418.83.

J. E. D. Albuquerque Filho, L. C. P. Brandão, B. J. T. Fernandes and A. M. A. Maciel, "A Review of Neural Networks for Anomaly Detection," in IEEE Access, vol. 10, pp. 112342-112367, 2022, doi: 10.1109/ACCESS.2022.3216007.

Janine L. Wiles, Annette Leibing, Nancy Guberman, Jeanne Reeve, Ruth E. S. Allen, The Meaning of "Aging in Place" to Older People, The Gerontologist, Volume 52, Issue 3, June 2012, Pages 357–366, https://doi.org/10.1093/geront/gnr098

Jordan, Jeremy. (2018). Introduction to autoencoders. https://www.jeremyjordan.me/autoencoders/ [Last accessed 06.2022.]

Joshi, G. Int. Journal of Engineering Research and Applications, ISSN : 2248-9622, Vol. 4, Issue 1(Version 2), January 2014, pp.88-92

Lindemann, Benjamin & Maschler, Benjamin & Sahlab, Nada & Weyrich, Michael. A survey on anomaly detection for technical systems using LSTM networks, Computers in Industry, Volume 131, 2021, 103498, ISSN 0166-3615, https://doi.org/10.1016/j.compind.2021.103498

Mukamel DB, Saliba D, Ladd H, Konetzka RT. (2023) Association of Staffing Instability With Quality of Nursing Home Care. JAMA Netw Open. 2023;6(1):e2250389. doi:10.1001/jamanetworkopen..50389

Osborne, Tess & Lowe, Thomas & Meijering, Louise. (2023). Care and rhythmanalysis: Using metastability to understand the routines of dementia care. Social Science & Medicine. 331. 116099. 10.1016/j.socscimed.2023.116099.

Perry E, Walton K, Lambert K. (2023) Prevalence of Malnutrition in People with Dementia in Long-Term Care: A Systematic Review and Meta-Analysis. Nutrients.; 15(13):2927. https://doi.org/10.3390/nu15132927

Saito, Tomoko & Konta, Tsuneo & Kudo, Sachiko & Ueno, Yoshiyuki. (2023). Factors associated with community residents' preference for living at home at the end of life: The Yamagata Cohort Survey. Global Health & Medicine. 6. 10.35772/ghm.2023.01072.

Santos, Eleonora. (2023). Shifting Demographics and Economic Performance: Unraveling the Influence of Population Aging on GDP Dynamics and Regional Inequalities. Journal of Economic Analysis. 3. 118-128. 10.58567/jea03010006.

Saxena, Akash & Rajpoot, Vikram. (2021). A Comparative Analysis of Association Rule Mining Algorithms. IOP Conference Series: Materials Science and Engineering. 1099. 012032. 10.1088/1757-899X/1099/1/012032.

Schomakers, Eva-Maria & Ziefle, Martina. (2022). Privacy vs. Security: Trade-Offs in the Acceptance of Smart Technologies for Aging-in-Place. International Journal of Human-Computer Interaction. 39. 1-16. 10.1080/10447318.2022.2078463.

Steverson, M. (2022) Ageing and health, World Health Organization. https://www.who.int/news-room/fact-sheets/detail/ageing-and-health [Last accessed 02.2024.]

Use containers to Build, Share and Run your applications. (n.d.). https://www.docker.com/resources/what-container/ Last accessed 05.2022.

WHO (2022) UN Decade of Healthy Ageing – core narrative. Decade of Healthy Ageing. (https://cdn.who.int/media/docs/default-source/decade-of-healthy-ageing/decade-proposal-final-apr2020-en.pdf?sfvrsn=b4b75ebc_28). [Last accessed 10.02.2024]

Xavier, Flávio & Ferraz, Marcos & Marc, Norton & Escosteguy, Norma. (2003). Elderly people´s definition of quality of life. Revista brasileira de psiquiatria (São Paulo, Brazil : 1999). 25. 31-9. 10.1590/S1516-44462003000100000