



Семантични технологии в електронното здравеопазване

(Приложение на технологии за машинно самообучение за
подпомагане на самостоятелния живот на възрастни хора
и хора в неравностойно положение)

от

Кристин Илиева Александрова

Автореферат

*на дисертация за придобиване на образователна и научна степен
„Доктор“*

в

професионално направление

4.6. „Информатика и компютърни науки“, докторска програма
"Информационни системи" – Системи, основани на знания

Ръководител: Професор д-р Мария Нишева-Павлова

София, 2024

1	Въведение.....	3
1.1	История на проблема.....	3
1.2	Постановка на проблема.....	6
1.3	Изследователски въпроси, хипотеза и цел.....	7
2	Архитектура и имплементация на системата.....	8
2.1	Първа версия на прототипа, базирана на OpenRemote.....	9
2.1.1	Настройки.....	11
2.1.2	Симулация на целевия сценарий.....	12
3	ОРЗД Изисквания и изпълнение.....	15
4	Прогноза на поведение.....	18
4.1	Невронни мрежи.....	18
4.1.1	Автоенкодери (Autoencoders).....	19
4.1.2	Дълга Краткосрочна Памет (LSTM).....	20
4.2	Извличане на асоциативни правила.....	21
4.2.1	Априори.....	22
4.2.2	FPGrowth.....	24
4.3	Сравнение на невронни мрежи и алгоритми за извличане на асоциативни правила при прилагане за анализ на човешкото поведение.....	25
4.3.1	Невронни Мрежи.....	27
4.3.2	Извличане на асоциативни правила.....	29
5	Заключение.....	31
5.1	Дискусия.....	32
5.2	Приноси.....	38
5.3	Бъдещи изследвания.....	39
	Литература.....	42

1 ВЪВЕДЕНИЕ

През последните години забелязваме, че застаряващото население в световен мащаб нараства с безпрецедентна скорост, поставяйки значителни предизвикателства пред системите на здравеопазване и мрежите за социална подкрепа, което в перспектива има икономически последици (Akhter, M. & Kamraju, M., 2023 г.), (Santos, E., 2023). Концепцията за подпомаган живот от обкръжаващата среда - Ambient Assisted Living (AAL) - се очертава като обещаващо решение за справяне с нуждите на това застаряващо население. AAL системите имат за цел да подобрят качеството на живот и независимостта на възрастните хора чрез безпроблемно интегриране на интелигентни технологии в тяхната жизнена среда. Въпреки нарастващия интерес и потенциалните ползи от AAL системите, се констатира силна ограниченост тяхното използване и наличност в световен мащаб, което поставя необходимостта от по-нататъшни изследвания и разработки за откриване и смекчаване на ограничаващите фактори, които биха позволили създаването на ефективни и лесни за употреба прототипи.

1.1 История на проблема

През последната година тема с нарастващ интерес в световен мащаб беше скоростта на демографското застаряване и предизвикателствата, които поставя пред всички страни по отношение на здравеопазването и социалните системи. Същото може да се види и в съобщението на Световната здравна организация (СЗО): „До 2030 г. един на всеки 6 души в света ще бъде на възраст над 60 години. В този момент делът на населението на 60 и повече години ще нарасне от 1 милиард през 2020 г. на 1,4 милиарда. До 2050 г. населението на света от хора на възраст над 60 години ще се удвои (2,1 милиарда). Очаква се броят на хората на възраст над 80 години да се утрои между 2020 г. и 2050 г., за да достигне 426 милиона (Steverson, M., 2022). В момента личните грижи, старческите домове и болниците се оказват едновременно скъпи и неспособни да се справят с прогнозирания брой хора през следващите години. Това поставя повече фокус върху намирането на начини възрастните хора да живеят сами, с минимална помощ от лица, които се грижат за тях, семейство или медицински персонал.

Общото събрание на Организацията на Обединените Нации (ООН) обяви 2021-2030 г. за Десетилетие на остаряването в добро здраве на ООН и поиска от СЗО да ръководи изпълнението. Десетилетието на ООН за остаряване в добро здраве предполага глобално сътрудничество, обединяващо правителства, гражданско общество, международни агенции, професионалисти, академични среди, медии и частния сектор за 10 години съгласувани, каталитични и съвместни действия за насърчаване на по-дълъг и по-здравословен живот (Steverson, M., 2022), подробности за които могат да се видят в придружаващия план за действие (WHO, 2022). Там също виждаме изискването за разработване на „помагащи технологии, като същевременно се гарантира, че

използването на тези услуги не причинява финансови затруднения на потребителя“ и „Насърчаване на използването на безопасна, достъпна, ефективна цифрова технология в интегрираните грижи. Анализ на пазара на труда и провеждане базирано на нуждите планиране, за да се оптимизират настоящите и бъдещи работни сили, за да посрещне нуждите на застаряващото население“. Това е мястото, където системите за подпомагане (AAL) влизат в картината. Тяхната единствена цел е да подобрят независимостта и качеството на живот на хората, нуждаещи се от помощ, независимо дали това е в старчески дом или в собствената им домашна среда. AAL системите са показали огромен потенциал за подобряване на качеството на живот не само на възрастни хора, но и на хора с увреждания.

Понастоящем можем да обединим наличните възможности за грижи за възрастни хора в три основни категории, всяка от които има своя собствена цена. „Разходите за дългосрочни грижи за възрастни хора, включително както разходите за старчески дом, така и за здравна помощ в дома, достигнаха 61 милиарда евро през 2019 година. Половината от тези разходи са за старчески домове, докато само около 22,5% от бенефициентите използват тези институции. Разходите с лични средства се различават значително при различните видове грижи. Разходите с лични средства съставляват едва около 7% от общите разходи за домашни грижи. В старческите домове 41% от разходите са плащания с лични средства.“ (Geiger, J. et al., 2023).

- Институционална грижа. Например старческите домове, в които работят медицински лица, които се грижат едновременно за много хора, и няма акцент върху това средата да прилича на домашна.
- У дома. В този случай лицето продължава да живее в дома си и за него се грижи болногледач, който обикновено е някой от членовете на семейството му като партньор или деца. Понякога е възможно да бъде нает болногледач с медицински опит.
- Обществени грижи, домове, специално построени за възрастни хора, и т.н. Някои страни и региони, като Германия, имат жилища, предназначени за възрастни хора. Тези домове са достъпни и предоставят интегрирани бутони за спешни случаи и лесен достъп до службите за спешна помощ. Те са междинна опция, която позволява на лицето да живее самостоятелно, като същевременно въвежда някаква форма на наблюдение и грижа.

Междувременно наблюдаваме силно предпочитание на възрастните хора да остаряват в къщи. Практика, обикновено наричана в литературата „стареене на място“. „Възрастните хора искат избор къде и как да остаряват на място. „Остаряването на място“ се разглежда като предимство по отношение на чувството за привързаност или

свързаност и чувствата на сигурност и познаване както по отношение на домовете, така и на общностите. Остаряването на място е свързано с чувството за идентичност както чрез независимост и автономност, така и чрез грижовни взаимоотношения и роли в местата, където хората живеят.“ (Janine L. W. et al., 2012)

Проучване в Китай показва, че предпочитанията на 91,9% от респондентите са домашни грижи за възрастни хора, следвани от грижи в общността и медицински сестрински грижи, като институционалните грижи са най-малко предпочитаната алтернатива (Du, J. et al., 2023 г.). Подобно заключение е направено и от проучване в Япония, където „Сред 10 119 отговорили, 61% избраха дома си като най-желаното място, където да прекарат последните си дни“. (Saito, T. & Konta, T. & Kudo, S. & Ueno, Y., 2023). Един фактор, допринасящ за това, е възприеманото качество на грижите, които хората получават в институциите, което се наблюдава в други изследвания (Fahley T, et al. 2003), (Mukamel, D. et al., 2023), (Borsa, A. и др., 2023). Въпросът какви биха били приемливите възможности за грижа също е неясен, както е показано в „Спорове при справянето с грижите за възрастни хора: междусекторен анализ, насочен към болногледачите в Германия“ (Auth, D. & Leiber, S. & Leitner, S., 2023 г.).

В друго изследване можем да видим, че „Малко повече от половината от изследваната извадка (57%) определят текущото си качество на живот с положителни оценки, докато 18% представят отрицателна оценка за него. Група от 25% определя настоящия си живот като неутрален или имащ и двете стойности (положителна и отрицателна) ... Основният източник на отчетено ежедневно благополучие е участието в селски или домашни дейности. Сред интервюираните липсата на здраве е основният източник за липса на благосъстояние, въпреки че има междуличностни различия по отношение на това, което всеки субект смята за загуба на здраве.“ (Xavier, F. et al., 2003).

Деменцията е често срещано заболяване, което засяга в различна степен много хора. До 2050 г. се предвижда да има 154,8 милиона случая в сравнение с 57,4 милиона през 2019 г. (Ghith, N., 2022). „Деменцията променя хранителното поведение, сигналите за глад и жажда, функцията за преглъщане, способността за самостоятелно хранене и разпознаването и интереса към храната. Съществуват значителни различия в докладваното разпространение на недोхранването сред възрастните хора, които живеят в дългосрочни грижи. ... Разпространението на недохранването варира от 6,8 до 75,6%, а рискът от недохранване е 36,5–90,4%. Общата честота на недохранване при тези с деменция при дългосрочни грижи е 26,98% (95% CI 22,0–32,26, $p < 0,0001$, $I^2 = 94,12\%$). Обобщеното разпространение на риска от недохранване при тези с деменция е 57,43% (95% CI 49,39–65,28, $p < 0,0001$, $I^2 = 97,38\%$). Недохранването е широко разпространено при хората с деменция, живеещи в дългосрочни грижи.“ (Perry, E. & Walton, K. & Lambert, K., 2023).

Освен това можем да наблюдаваме стереотип, присъстваща в живота на възрастните хора, и това е често срещано при намаляване на когнитивните и психологическите способности, както е заключено от Bergua, V. et al., 2006:

„Въпреки че стереотипните дейности са важни за нормалното функциониране във всички фази на живота, тяхното изразяване при възрастните хора може да бъде свързано с когнитивна и психологическа уязвимост. Връзката между тези променливи е изследвана при 235 възрастни френски участници от кохортното проучване RAQUID. Намерени са междусекторни положителни асоциации между предпочитанията за стереотипи, нивата на тревожност и депресия и когнитивните оплаквания. Общият когнитивен спад за период от три години също се свързва с по-голямо желание за спазване на стереотипно поведение в края на този период. Прогресивното стереотипизиране на поведението и дейностите при по-възрастните хора се обсъжда като маркер за афективна и когнитивна уязвимост и неговото разбиране има потенциал за подобряване на ранното откриване на трудностите при адаптиране и цялостната грижа за тази популация (Bergua, V. et al., 2006).

При възрастните с деменция, поддържането на ежедневието и познаването на обкръжаващата среда е от решаващо значение за умствената яснота и стабилност на човека и качеството на грижите, които получават. „Ние демонстрираме как метастабилността осигурява разбиране на непрекъснато променящите се ритми на всеки ден и ни позволява да преминем отвъд непосредствеността на аритмичните прекъсвания и да изследваме фините промени, които се случват в (поли)ритмите. По този начин евритмията като метастабилно равновесие ни позволява да изследваме постепенното и фино развитие и промени в грижите за деменция и други стереотипни практики в здравната география (Osborne, T. & Lowe, T. & Meijering, L., 2023).

1.2 Постановка на проблема

Настоящото състояние на проблемното пространство поставя повече фокус върху намирането на начини възрастните хора да живеят сами, с минимизирана помощ от лица, които се грижат за тях, семейство или лекари. От гледна точка на компютърните науки, приносът към решението се съдържа в системите за подпомагане на обкръжението (AAL), които капсулират всички технологични постижения, които намират приложение за подобряване на автономността и качеството на живот на възрастните хора. Това обяснява нарастващото количество изследвания в областта на AAL, но въпреки нарастващото търсене, много малко от инициативите и прототипите намират продуктивно и широко разпространено приложение, което поставя въпроса защо това е така. Можем да предположим, че в предишни прототипи на AAL системи фокусът е бил върху разрешаването на техническата функционалност за сметка на сигурността, поверителността на данните и т.н. (Schomakers, E.M. & Ziefle, M., 2022), тъй като тези

системи могат да повишат качеството на живота на възрастните хора и може да осигури жизненоважна подкрепа в ежедневните им дейности.

Една потенциална причина за ограничената наличност може да бъде цената на поддръжката на човек в системата. Това е комбинация от множество фактори. Ще назовем няколко: цената на инфраструктурата за работа и поддръжка на системата; поддържане и осигуряване на актуална лична информация; времето и ангажираността, изисквани от човек, който се грижи за него. Един от начините за намаляване на разходите за управление на широкомащабна система е внедряването на алгоритми за машинно самообучение. Едно потенциално пространство е откриването на отклонение в поведението на стандартния човек. Например, разпознаване дали приемат редовно лекарствата си, дали са пропуснали доза или са взели двойна доза, защото са забравили, моделите им на сън влияят ли върху когнитивните им способности и т.н. Притежаването на тези знания би позволило на даден болногледач да наблюдава устойчиво повече хора, от колкото в момента, където съотношението в домашна среда е почти 1:1, един болногледач на всеки възрастен. С предложената система е възможно един човек да се грижи и наблюдава няколко свързани към системата възрастни едновременно, тъй като могат лесно да реагират на извънредни ситуации, като същевременно получават ежедневен доклад за нормалното поведение на всеки човек.

Човешкото поведение е сложен феномен. В дисертационния труд ние се фокусираме върху възрастните хора и особено хората с деменция, което е пространство, в което обикновено виждаме наличие на стереотип, с отклонения, сочещи към по-големи проблеми и потенциално опасни епизоди за човека, като изгубване или объркване. В тази работа ние формулираме хипотезата, че решение на този проблем е да се разгледат модели за машинно самообучение за откриване на извънредни стойности, в този случай можем да стесним проблема до установяване на модел на нормалното поведение на човека и разглеждане на всички извънредни стойности на този модел като прекъсване на стереотипа, което трябва да бъде оценено от човек и при нужда да предизвика подходящата реакция.

1.3 Изследователски въпроси, хипотеза и цел

Имайки предвид постановката на проблем, ще формулираме няколко изследователски въпроса, на които ще се стремим да отговорим.

1. Можем ли да създадем рентабилна AAL система?

Когато говорим за рентабилна системи, има няколко параметъра, по които може да се прецени това, и които съответно се стремим да оптимизираме. Това включва изчислителните ресурси за управление на AAL системата и за машинното самообучение на поведенческите модели; разходите за разработка и поддръжка на системата и моделите; цената на потребител на системата за първоначално

включване в системата; цената на човек за осигуряване на функционална AAL система и актуални модели за машинно самообучение.

2. Можем ли да създадем AAL система, осигуряваща поверителност на данните, точно съобразена с изискванията на Общият регламент относно защитата на данните (ОРЗД)? Също така какво би било влиянието на ОРЗД върху обучените алгоритми?
3. Би ли имала полза AAL система от подобряването ѝ с персонализиран алгоритъм за машинно обучение, обучен върху събраните данни, който има за цел да идентифицира извънредните стойности в поведението на лицето и да отправи подходящ сигнал към техния гледач? Ако е така, какъв би бил най-добрият подход за справяне с проблема, без да се прави компромис с цената на системата и установените изисквания за поверителност на данните.

Въз основа на горните въпроси можем да формулираме следната хипотеза:

„Можем да преизползваме междинен софтуер за интелигентен дом с отворен код, за да създадем AAL система, съобразена с разходите и поверителността на данните, да я разширим с алгоритми за машинно самообучение по полезен начин и да докажем, че алгоритмите за извличане на асоциативни правила (ARM) могат да се използват за разпознаване на човешкото поведение и те биха били по-добрият избор в сравнение със стандартните подходи за откриване на отклонения, тъй като те като цяло са по-евтини, по-лесно изпълняват изискванията за поверителност на данните и имат обясними резултати“

Целта на тази дисертация е да докаже хипотезата като цяло и в нейните производни части, като отговори на зададените изследователски въпроси.

2 АРХИТЕКТУРА И ИМПЛЕМЕНТАЦИЯ НА СИСТЕМАТА

В тази разработка ние правим уговорката, че готовите мидълуерни платформи с отворен код са по-подходящи за създаването на Системи за подпомагане на обитаването от обкръжаващата среда, отколкото техните специфични за домейна партньори, поради по-широкото им приемане в множество области и отвореността към нова употреба естество на случаите. Освен това, ние предполагаме широкото приложение на алгоритми за машинно обучение при разпознаването на поведенчески модели при хора в напреднала възраст с леки случаи на деменция. За да проверим и илюстрираме тези точки наистина, ние създаваме прототип на AAL система, съзнавайки необходимите усилия за създаване и използване на системата и с цел да запазим общите разходи за работа на полученото решение до минимум. След това симулираме работата на споменатия прототип и оценяваме неговата полезност в предложените случаи на употреба. В следващите раздели разглеждаме различните анализи и решения за разработка, които бяха направени като част от създаването на споменатия прототип.

Преди това разгледахме внимателно детайлите на четири от най-известните кандидати за мидълуер платформа, на която да се базира прототип. Имаше много други опции, които не успяха да покрият поне половината от зададените критерии и като такива бяха изключени от това сравнение. Дефинирахме девет критерия, по които бяха оценявани платформите, и обобщихме заключенията за всеки един от най-изявените кандидати – openHAB, ThingsBoard, universAAL, OpenRemote. OpenRemote напълно удовлетворява желаните сценарий и предоставя възможности за по-нататъшно развитие и разширяване при разумен компромис между ползите и разходите. Затова в тази работа изграждаме прототип, базиран на OpenRemote, който би илюстрирал функционалността и потенциала на този тип система. Не трябва да изключваме напълно universAAL, вместо това допълнителна цел на прототипа би била да покаже силните и слабите страни на базирана на OpenRemote AAL система, на свой ред оправдавайки или обезсърчавайки по-нататъшни изследвания за създаване на подобен прототип, базиран на universAAL.

2.1 Първа версия на прототипа, базирана на OpenRemote

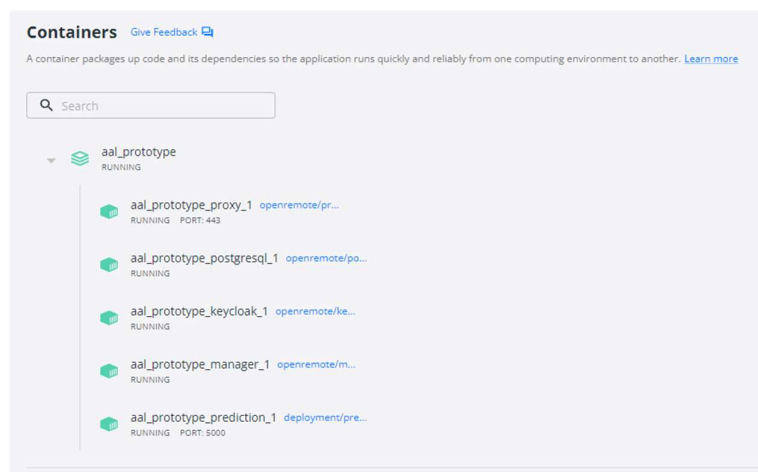
След като избрахме нашата междинна платформа, можем да започнем да разработваме нашия прототип на AAL система. Има няколко въпроса, на които трябва да отговорим и да ги потвърдим в първия прототип, който тестваме, който ще послужи като основа за бъдещи версии: Как е пакетирана дистрибуцията на OpenRemote и как е конфигурирана средата за разработка; Как се изпълняват персонализирани модификации и как се пакетират в готов продукт; Как се моделират сензорите и какви са възможностите за създаване на правила въз основа на тях; Можем ли да създадем виртуална симулация на един от целевите сценарии, които сме дефинирали по-рано, без да свързваме системата с реални сензори и без да компрометиране функционалността на системата? Това е предпоставката на този раздел.

Тъй като OpenRemote е решение с отворен код, има два начина за получаване и разработка, базирано на оригиналния код. Единият е чрез клониране на GitHub хранилището и изграждане на системата локално. Другата опция би била да се използват създадените Docker контейнерни изображения за OpenRemote, публикувани в Docker Hub. Изображение на Docker контейнер е лек, самостоятелен, изпълним пакет от софтуер, който включва всичко необходимо за стартиране на приложение: код, време за изпълнение, системни инструменти, системни библиотеки и настройки. Изображенията на контейнери стават контейнери по време на изпълнение, а в случая на Docker контейнери – изображенията стават контейнери, когато се изпълняват на Docker Engine. Наличен както за Linux, така и за Windows-базирани приложения, контейнеризираният софтуер винаги работи по един и същи начин, независимо от инфраструктурата. Контейнерите изолират софтуера от неговата среда и гарантират, че работи еднакво въпреки разликите, например между разработката и етапа. Множество контейнери могат да работят на една и съща машина и да споделят ядрото на ОС с други контейнери, всеки

от които работи като изолиран процес в потребителското пространство. Контейнерите заемат по-малко място от виртуалните машини, те обикновено са с размер десетки МВ, могат да обработват повече приложения и изискват по-малко виртуални машини и операционни системи. (Use containers to Build, Share and Run your applications, n.d.). Това гарантира лекотата на разпространение, която първоначално дефинирахме. Полученият прототип на системата AAL също е пакетирани и разпространен като контейнеризирано приложение. Ако успеем да го изградим като абстракционен слой към текущите контейнери на OpenRemote, можем да използваме повторно всички актуализации и подобрения на платформата.

За да създадем контейнерите, има две неща, от които се нуждаем. Docker Engine и файл за съставяне на докер. Docker Engine използва дълго работещ демон процес, наречен `dockerd`, за създаване и управление на Docker обекти, включително контейнери. Той действа като приложение клиент-сървър с процеса на демон, работещ от страна на сървъра и интерфейс на командния ред (CLI), който взаимодейства с демона. Има няколко начина да получите Docker Engine, но един от по-лесните методи е приложението Docker Desktop. Docker Desktop включва Docker Engine, от който се нуждаем, Docker CLI клиент, Docker Compose, Docker Content Trust, Kubernetes и Credential Helper.

Нека се съсредоточим върху Docker Compose. Това е инструмент, който е разработен, за да помогне за дефинирането и споделянето на многоконтейнерни приложения. Това се реализира със създаването на специфични YAML файлове, наричани също файлове за композиране на Docker, там се дефинират всички услуги, мрежи и обеми за приложение на Docker и контейнер се завърта или разрушава въз основа на тази информация. Споменатият файл се намира в основата на проекта на приложението и обикновено се нарича „`docker-compose.yml`“. В случая на OpenRemote, файлът е стандартен и ни позволява много модификации, базирани на описаните параметри.



Фигура 1. Персонализирана инсталация на OpenRemote в Docker Desktop

Новият контейнер наричаме „prediction“. За създаването му се нуждаем от няколко файла. За начало създаваме `train.py fail`, който съдържа цялата логика за обучението на модела, включително получаване на данните от контейнера `postgres`, почистването и разделянето им. Ще финализираме съдържанието на този файл в секцията с резултати, след като сравним точността на различните подходи. Създаваме файл `requirements.txt`, който описва всички необходими `python` пакети, които трябва да бъдат инсталирани, за да е възможно обучението. Споменахме също, че този контейнер трябва да може да обработва входящи заявки и да прави запитвания към модела за тях, това е ролята на файла `api.py`. За да създадем контейнер за „предсказания“, трябва да създадем `Dockerfile`, който описва типа контейнер и операциите, които трябва да бъдат извършени. Може да се раздели на четири основни секции, като първо създаваме контейнер от изображението на контейнер на `Jupyter`. След това копираме файла с изискванията и използваме `pip` на `python`, за да инсталираме всички необходими модули. Създаваме файлова структура и променливи на околната среда, които позволяват на `API` кода да намира моделните файлове и изпълнимите файлове и да извлича необходимите прогнози. Накрая копираме двата файла на `Python` и съответно обучаваме модела и стартираме сървъра. Получената контейнерна структура може да се види на фигура 1.

2.1.1 Настройки

Както вече видяхме в предишния раздел, `OpenRemote` инсталацията е базирана на `Docker` и могат да бъдат постигнати много модификации, когато се определят параметрите за създаване на необходимите контейнери. Например, първоначално главната парола на инсталацията е `„secret“`, това се основава на два параметъра, които трябва да съвпадат по стойност, единият е в дефиницията на контейнера на `keycloak`, наречен `KEYCLOAK_PASSWORD`, а другият е в дефиницията на мениджърска услуга, наречена `SETUP_ADMIN_PASSWORD`.

В допълнение към корекциите, направени чрез файла `Docker Compose`, можем допълнително да персонализираме визуалните елементи на `OpenRemote` и потребителското изживяване. Всичко това се прави по структуриран начин, който е правилно описан в официалната документация на `OpenRemote`, която се хоства на `GitHub`. Нека започнем, като разгледаме файла `manager_config.json`. Това е `JSON`, който позволява модификации на външния вид и усещането на приложението за управление, което също е местоположението на потребителския интерфейс. Както вече обсъдихме в първоначалната оценка на междинния софтуер, `OpenRemote` поддържа `multi-tenancy` под формата на различни „сфери“ за различните потребители. В съответствие с тези концепции компонентите на потребителския интерфейс могат да бъдат специфични за сфера. Името на раздела на приложението се определя от параметъра `„appTitle“`, в параметъра `styles` виждаме основните цветове на приложението. В този пример сме дефинирали предварително зададените настройки по подразбиране за сфери, ако нищо

друго не е посочено, и допълнителни конфигурации за област с техническо име „ivan“, което представлява дома на нашата примерна личност Иван.

По същия начин можем да обменим картата, за да покажем регион по наш избор. В този прототип вземаме карта на България и позиционираме симулирания пример вкъщи в близост до сградата на Факултета по математика и информатика на Софийския университет „Св. Климент Охридски“. Изискването към картите е те да идват като данни за векторни плочки, по-точно във формат Mapbox GL. Други формати изискват допълнителни усилия, но не са невъзможни за прилагане. Освен това има поддръжка за растерни карти (Mapbox JS), но те изискват да работи допълнителен контейнер за карти, който след това обслужва картографските данни към приложението за управление. Въпреки това, за нашия сценарий е достатъчна векторна карта на град София, тъй като се стремим да представяме няколко домакинства едновременно и да имаме възможност за допълнително разширяване на функционалностите на системата извън дома на човек.

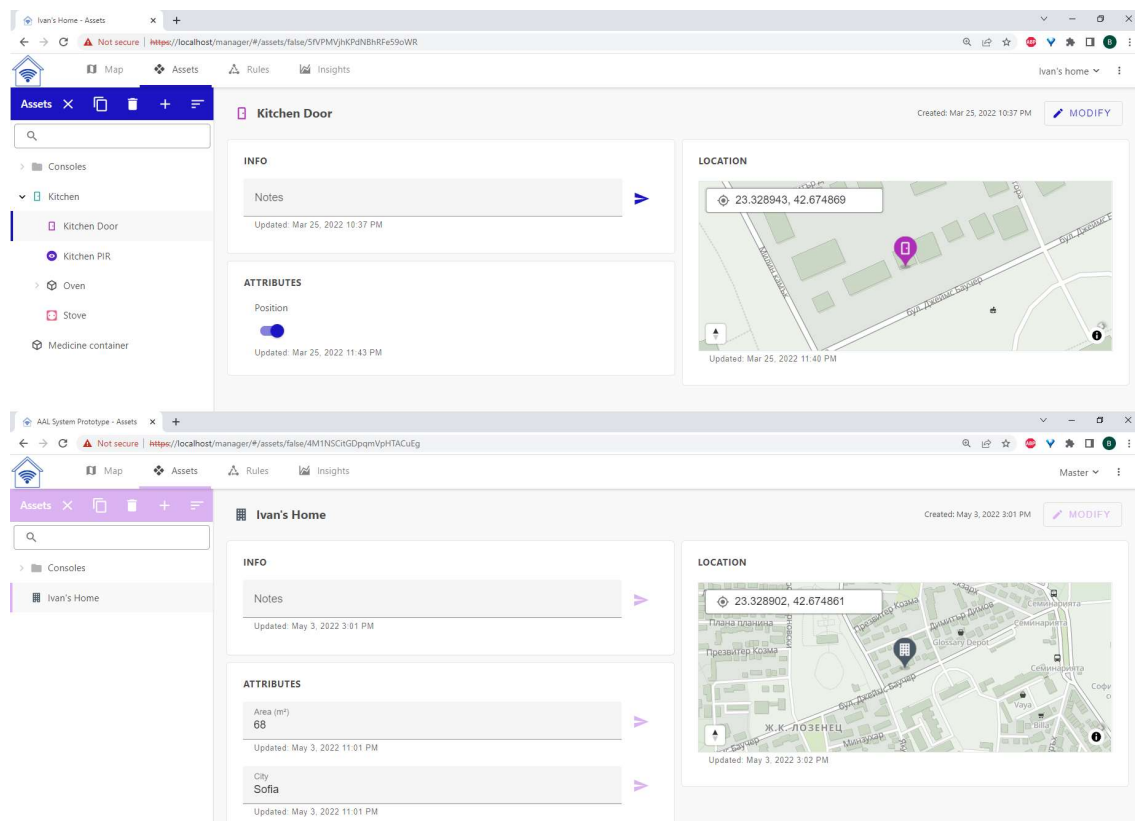
Има два файла, от които се нуждаем, за да дефинираме новата карта към приложението: `mapsettings.json` и `mapdata.mbtiles`. `Mapdata.mbtiles` съдържа данните за векторна карта, докато файлът `mapsettings` съдържа настройки, свързани с изходните данни на плочките на картата, както и настройки за изобразяване на потребителския интерфейс, като централната точка на картата, както и нива на мащабиране. Освен това в този JSON файл е възможно да се дефинират отделни настройки за различни сфери. Това позволява на едно и също приложение да се фокусира върху различни домакинства, в зависимост от потребителя, който има достъп до него. След това и двата файла се намират в папка с име `map`, която е на същото ниво като папката на мениджъра.

2.1.2 Симулация на целевия сценарий

Когато определяхме целевата личност за нашия случай на употреба, ние подчертахме важността на наблюдението на електрическите уреди, тъй като възрастен човек може да се разсеи и да забрави, че е готвил, което на свой ред създава риск от пожар в къщата. Затова първата задача при създаването на прототипа беше създаването на симулация на въпросния сценарий. Ако подходът на междинния софтуер за обработка на тези случаи не е подходящ за личността, която имаме предвид, разумно е да потърсим алтернативи за основата на прототипа. Имайки предвид това, нека моделираме най-лесния сценарий за наблюдение – а именно печката оставена ли е без надзор и включена? Същият въпрос за електрически уред като фурната е различен, тъй като много рецепти за готвене на фурна предвиждат по-дълго време за готвене и без обратна връзка от човека, който готви, е трудно да се прецени дали фурната трябва да е включена 10 минути или 2 часа, а да не споменаваме, че повечето печива изискват предварително загрята фурна, което би било в противоречие с всяко правило за наблюдение на празна включена фурна. За котлони въпроса е по-лесен, тъй като няма концепция за предварително загряване, така

че всеки път, когато котлона е включен и няма съд директно върху него, е причина за безпокойство. Също така не се препоръчва възрастният човек да оставя котлон без надзор за дълго време. С други думи, този проблем може да се извлече от въпроса включен ли е котлона и има ли присъствие в кухнята?

Следователно, първата стъпка би била да се създадат активи, представляващи кухнята, включително сензор за откриване на движение и сензор за печка, показващи дали печката е включена или изключена. Активът на стаята няма допълнителни атрибути и служи само като организационна единица за всички устройства, разположени в кухнята. Активът на вратата има един задължителен атрибут, който е позицията на вратата в булев формат, на фигура 2 стойността е вярна, което означава, че вратата е отворена. Ако имахме такива сензори, оборудвани към вратата, това щеше да е мястото за свързване на стойностите в системата. За целите на симулацията активът на вратата не се използва.



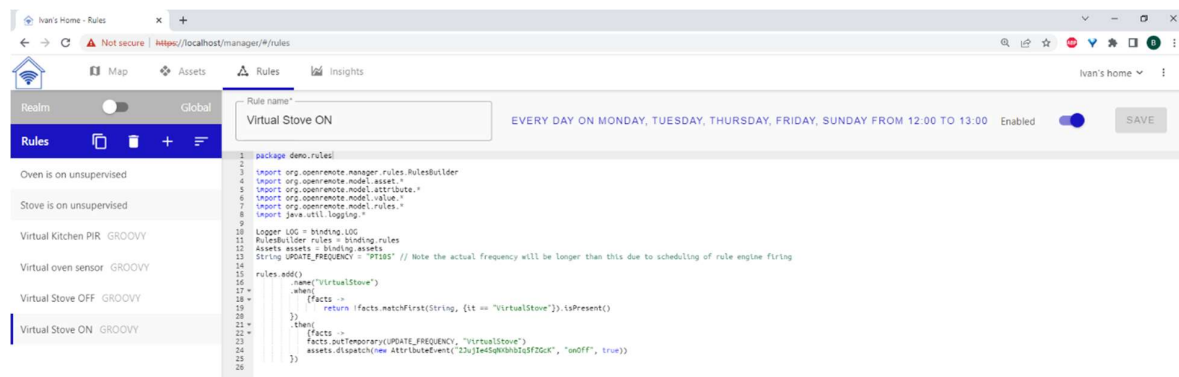
Фигура 2. Активи, моделирани в сферите „Master“ и „Ivan's home“ с цел симулация на сценарии

Виждаме също същата страница с активи, но от сферата „Master“. Тъй като сме влезли като администраторски потребител, имаме права за редактиране на всички сфери и сме свободни да превключваме, както решим. Както очаквахме, активите са специфични за сферата, в която са създадени. Следователно в сферата „Master“ има само един актив и това е сградата на апартамент на Иван, като индикация за поддържаните

домакинства, системата в момента поддържа. Всички устройства, създадени от кухнята, са ограничени до сферата „Ivan’s home“. Това доказва първоначалната концепция за multi-tnancy и ни позволява допълнително да подобрим системата в тази посока.

На настоящия етап от разработването на прототип, ние се опитваме да визуализираме концепция за функционалностите на системата и подхода за разрешаване на проблеми. В съответствие с това нямаме референтен дом или устройства, от които да получаваме данни в реално време, така че трябва да създадем виртуални устройства, които да се държат по начин, подобен на реалните им аналози. За дефинирания сценарий се нуждаем от PIR устройство за виртуален детектор на движение, разположено в кухнята на човек и печка, което може да уведоми системата, когато е включена или изключена.

Един подход за създаване на този тип, ако дадено устройство е чрез Groovy правилата на OpenRemote. След като бъде създаден активът за печка, един от неговите атрибути печка се нарича „onOff“. Това е булева стойност, която е вярна, когато печката е включена. Вместо да свързваме този актив с действително устройство, можем да използваме правило Groovy, за да симулираме стойностите. OpenRemote има специфична класова структура, която позволява да намерите необходимия актив по техния ID и да зададете желаната от нас стойност. На фигура 3 можем да видим правилото, което симулира, че печката е включена. Може също да забележите, че има повторение на това правило, а именно понеделник, вторник, четвъртък, петък и неделя на обяд от 12:00 до 13:00 часа. Това, комбинирано със зададената честота от 10 секунди, гарантира, че всеки ден в продължение на един час нашата виртуална фурна ще докладва, че е включена. Правилото за виртуално изключване на печката е същият код, но с различна стойност на атрибута false и повторение веднъж на всеки споменат ден точно в 13:00 часа.



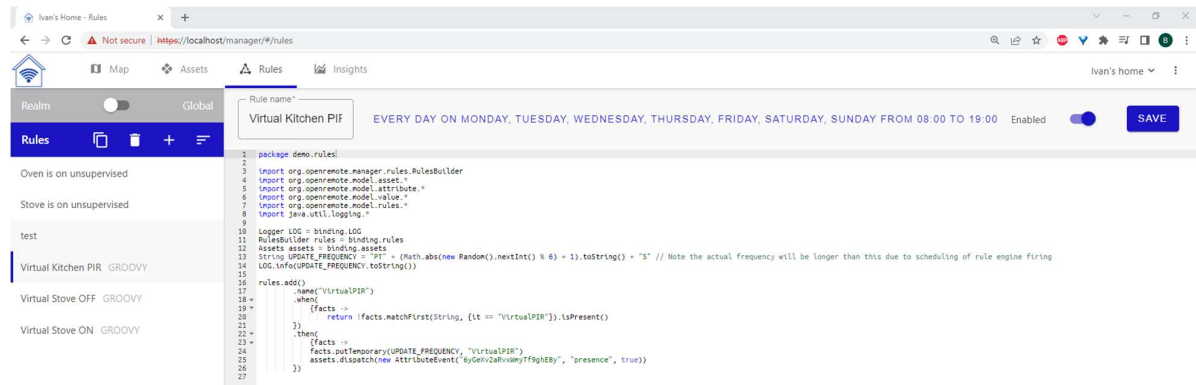
The screenshot shows the OpenRemote web interface for a rule named "Virtual Stove ON". The rule is configured to run "EVERY DAY ON MONDAY, TUESDAY, THURSDAY, FRIDAY, SUNDAY FROM 12:00 TO 13:00" and is currently "Enabled". The Groovy code for the rule is as follows:

```
1 package demo.rules;
2
3 import org.openremote.manager.rules.RulesBuilder
4 import org.openremote.model.asset.*
5 import org.openremote.model.attribute.*
6 import org.openremote.model.value.*
7 import org.openremote.model.rules.*
8 import java.util.logging.*
9
10 Logger LOG = binding.LOG
11 RulesBuilder rules = binding.rules
12 assets assets = binding.assets
13 String UPDATE_FREQUENCY = "10s" // Note the actual frequency will be longer than this due to scheduling of rule engine firing
14
15 rules.add()
16     .name("VirtualStove")
17     .facts()
18     .return ifacts.matchFirst(String, (it == "VirtualStove")).isPresent()
19     .then()
20     .facts()
21     .facts().setTemporary(UPDATE_FREQUENCY, "VirtualStove")
22     .assets().dispatch(new AttributeEvent("23u1e5qV0hb1a5f756k", "onoff", true))
23
24
25
26
```

Фигура 3. "Virtual stove ON" Groovy правило

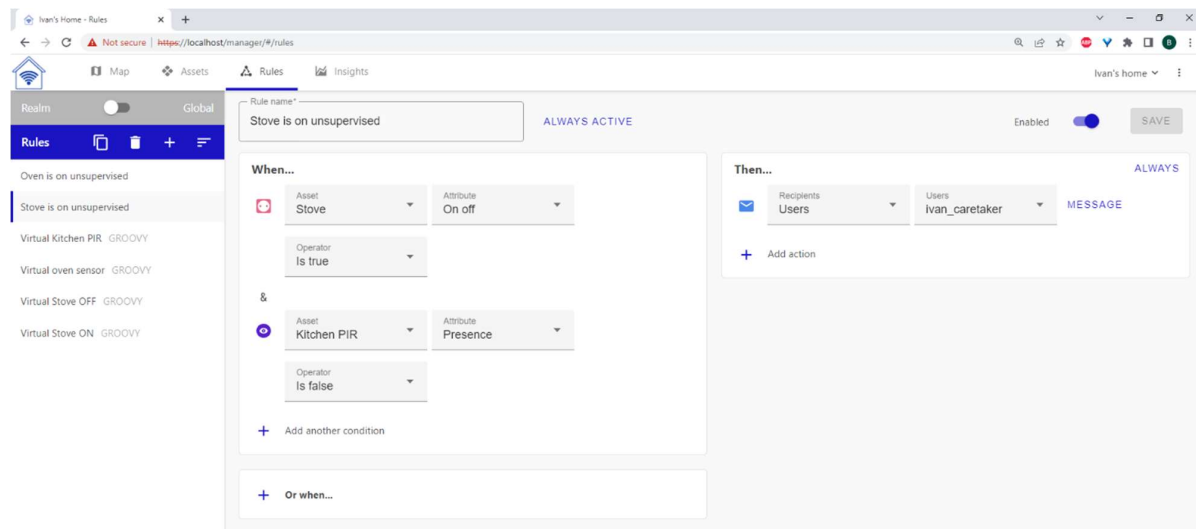
Същото важи и за виртуалния PIR сензор. Работи всеки ден от 8:00 до 19:00 часа и отчита присъствие в Кухнята на произволни интервали. Първоначално това беше

реализирано чрез рандомизиране на интервалите, на които се отчита присъствие. (Фигура 4)



Фигура 4. “Virtual Kitchen PIR” Groovy правило

На Фигура 5 виртуалните сензори са комбинирани от праволинейно правило „when-then“, което основно гласи, че ако печката е включена и няма открито присъствие в кухнята, гледачът на лицето трябва да бъде уведомен по имейл. По-късно това беше заменено от по-усъвършенствано Groovy правило, което взема предвид изминалото време от откриването на присъствие. Освен това се изпраща известие до телефона на пазача чрез приложението Pushsafer, което е препращане на имейла или директна POST заявка от Groovy скрипта.



Фигура 5. “Stove is on unsupervised” When-Then правило

3 ОРЗД ИЗИСКВАНИЯ И ИЗПЪЛНЕНИЕ

Един основен недостатък на AAL системите, който възпрепятства тяхното масово приложение, е осигуряването на защита на личните данни. Тъй като AAL системите се възползват от статуса си на системи с въздействие върху общественото здраве и често

са част от изследователски проекти, те не са задължени да отговарят на каквото и да е законодателство или стандарт за сигурност. От своя страна много малко го правят, което възпрепятства по-широкото им приложение и развитие. В тази работа искаме да потвърдим, че осигуряването на поверителност на данните може лесно да бъде постигнато, когато споменатото съответствие се разглежда в началото на разработването на продукта. Като справка в тази работа ние използваме Общия регламент за защита на данните на ЕС 2016/679 (ОРЗД). Един от основополагащите принципи на това законодателство е защита на данните на етапа на проектирането и защита на данните по подразбиране. В нашия случай това означава разбиране на ОРЗД и дефиниране на системните функционалности. Защита на данните на етапа на проектирането се доказва като водещ подход при въвеждане на сигурност в една система, тъй като с времето и разрастването на проекта разходите за реконструкция в името на защитата на данните растат експоненциално. Ето защо в отделен анализ разгледахме подробно последиците от ОРЗД върху AAL система, която има за цел да комбинира традиционно ненаатрапчиво събиране на данни с алгоритми за машинно обучение, за да извлече представа за физическото и психическото състояние на лицето, за което се грижи системата. В резултат на тази работа изведохме 15 изисквания към прототипа на AAL системата и нейното поведение:

1. Спазването на ОРЗД трябва да бъде потвърдено от правен екип. Те са тези, които определят дали обработването е справедливо, законно и прозрачно; какви са законовите основания за обработване; за кои данни и до каква степен е приложимо всяко искане на субект на защитата на данни спрямо ОРЗД; има ли резултати от автоматизирана обработка на личните данни, които са обект на ОРЗД и др.
2. При проектирането на системата и работата на нейните процеси трябва да гарантираме, че по всяко време, когато данните се обработват, те са сведени до минимум, включително когато данните се споделят с друг обработващ лични данни.
3. Преди изграждането на архитектурата на AAL системата, въз основа на рисковете за сигурността, трябва да се направи „Оценка на въздействието върху защитата на данните“, както е и изискването на член 35 на ОРЗД.
4. Данните трябва да бъдат отделени по такъв начин, че ако даден “субект на данни” иска да види собствените си данни, ние можем да ги предоставим, без да компрометираме поверителността на всички други “субекти на данни” в системата. По същия начин ако се наложи изтриване на данни, то трябва да е възможно без да бъдат засегнати останалите потребители на системата и цялостната ѝ производителност.

5. Трябва да осигурим механизъм на системата, така че да бъде проверен заявителя на “искания” към ОРЗД. Има много средства за удостоверяване и можем да изберем подходящ вариант за системата AAL.
6. Необходимо е да се анализира очакваното количество заявки, свързани с “искания на субект на данни”, за да можем да гарантираме, че обработката им е извършена в рамките на 30 дни. Това включва и случаи на изтриване или ограничаване на данни.
7. Трябва да се изготви и публикува съобщение за поверителност въз основа на законосъобразността на използването на данните и функционалността на системата.
8. Трябва да се създадат средства за уведомяване за субектите на данни в случай на нарушения на данните и като потенциален канал за комуникация относно законосъобразността и персоналното съгласие.
9. Въз основа на правната оценка трябва да е възможно да се избират и изтриват само лични данни, които подлежат на правото на изтриване, както е определено от правния екип.
10. Трябва да е наличен начин за маркиране на данни, така че да не се използват при обработка в случаите, когато обработката е ограничена съгласно “искане на субекта на данни”.
11. Трябва да е обособена стратегия за архивиране и възстановяване на данни, съгласно ОРЗД не е необходими при изтриване или ограничаване на данни същото да бъде направено и в съществуващите архиви. Въпреки това, трябва да сме наясно с възможността да възстановим лични данни, които са ограничени или изтрити.
12. В зависимост от използваните алгоритми трябва да оценим анонимизацията и псевдонимизацията на данните. Ако първото е осъществимо за всички лични данни, ние сме съвместими с ОРЗД, второто се препоръчва от ОРЗД като мярка за сигурност. И двете са валидни не само за обработваните данни, но и за резултатите от изпълнението на алгоритмите.
13. Криптирането от край до край също се споменава в GDPR като необходима мярка за сигурност, и като такава е очаквано изискване за нашата система.
14. В зависимост от това колко критична са функционалностите на системата, може да се наложи да се инвестира в архитектура позволяваща висока наличност или възстановяване след бедствие.
15. Управление на самоличността - всяка AAL система трябва да установи правилни дефиниции на роли и нива на достъп, така че личните данни да бъдат достъпни проследяемо и само с ясна подходяща цел.

4 ПРОГНОЗА НА ПОВЕДЕНИЕ

В предишните раздели разгледахме създаването на прототип на AAL система, която подпомага изпълнението на ежедневни дейности на възрастните хора с помощта на персонализирани сензори и комбинация от правила, такива че системата да е съобразена с лицето, за което е предназначена. Разгледахме приложението на тази система при възрастни хора с леки форми на деменция. Предпоставката е, че можем да създадем общ набор от правила, които могат да определят дали дадено лице се отклонява от обичайното си поведение, например възможно е човекът да забрави дали се е хранил днес и с каква честота, което може да доведе до преяждане или по-честото отказване от храна за целия ден. Показахме как тези прости сценарии могат да бъдат моделирани чрез няколко прости правила за вече свързани сензори, но когато говорим за отклонения от стандартното поведение, очевиден подход би бил прилагането на различни алгоритми за машинно обучение, които биха създали поведенчески модели. Идеята зад тези модели би била да се обучат на стандартен набор от данни, съдържащ различни дейности от ежедневието, ефективно създавайки модел който разбира какво е нормалното стереотипно поведение на определен човек. Съответно е възможно да анализираме всеки нов сензорен запис и да оценим дали е пример за стандартно поведение или обратното е отклонение от очакваното поведение.

4.1 Невронни мрежи

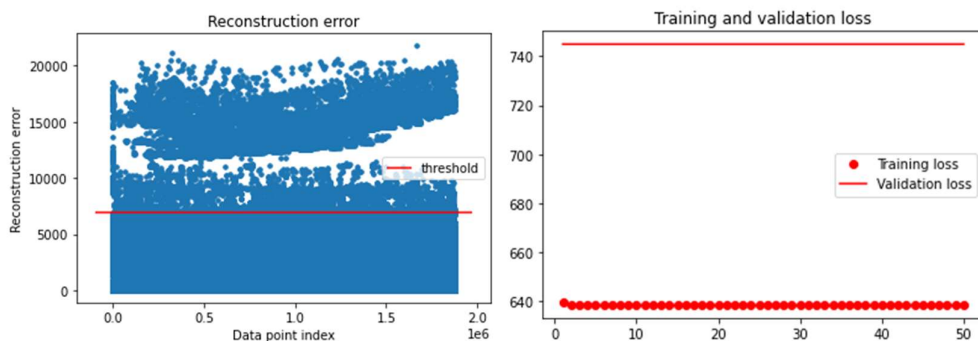
Невронните мрежи имат широко приложение при откриване на аномалии (J. E. D. Albuquerque Filho, 2022 г.), тъй като те са способни да научават сложни модели в многомерни, променящи се във времето данни. Това ги прави подходящи за приложение върху необработени данни от реалния свят, където обемът и сложността на данните могат да нараснат много бързо извън обхвата на обработка за други алгоритми. Невронните мрежи също са подходящи за използване в нашия случай на откриване на аномалии в човешкото поведение, тъй като те не изискват обучение върху набор от данни с идентифицирани аномалии, вместо това те са способни да извличат съответните характеристики от необработени данни. По този начин се идентифицират аномалиите въз основа на разликата между предказанието на модела и действителните наблюдавани стойности. Този механизъм позволява на невронните мрежи да работят по-добре от обичайните статистически методи, тъй като за разлика от тях невронните мрежи не предполагат специфично разпределение на данните. Те могат да се научават и откриват аномалии, без да изискват предварителна информация, а също така да извличат контекстуална информация за местоположението и тежестта на аномалията. Надграждайки това, една от силните страни на невронните мрежи е способността им да научават сложни и нелинейни модели от данни, което ги прави ефективни при улавянето на сложни връзки между променливи. Невронните мрежи могат да адаптират своите вътрешни параметри и структура по време на учебния процес. Тази гъвкавост им

позволява да обобщават добре различни видове аномалии, което ги прави подходящи за различни задачи за откриване на аномалии.

4.1.1 Автоенкодери (Autoencoders)

Автоенкодерите са невронни мрежи с предварителна връзка, където входната форма е същата като изходната. Идеята е, че входът се компресира в код с по-ниско измерение и след това изходът се реконструира от това представяне. Следователно всеки запис ще бъде кодиран и след това декодиран от невронната мрежа до предишното му състояние. Ако декодирането е успешно, можем да заключим, че това е запис, с който невронната мрежа е запозната, и да го категоризираме като нормално поведение. Ако невронната мрежа не може да реконструира записа, това се счита за аномалия. В този раздел разглеждаме по-подробно автоенкодерите (Jordan, J., 2018) и тяхното приложение при разпознаване на аномалии в човешкото поведение.

Най-лесният подход би бил да се предостави набора от данни на токенизатора без допълнителна намеса. Въпреки това, колоните „Стая“, „Обект“ и „Сензор“ служат само за изясняване на данните от сензора на крайния потребител, автоматичният енкодер няма да има полза от тях и те просто биха добавили допълнителни нива на сложност, следователно можем да ги изпуснем правилно далеч. Освен това не очакваме хората да действат в същата секунда или милисекунда, ние се интересуваме повече от частта от деня, в която се извършва всяка дейност, следователно има смисъл да закръгляме времената до най-близкия час и да разглеждаме регион около минути, ако е необходима допълнителна точност в реалния случай на употреба. Полученият модел се обучава за 16m 40.8s с по-малка загуба от предишния, но с неподвижна точност от 51.54% през всичките 50 епохи (Фигура 6).



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

Независимо от това, може да има приложение на автокодери в нашия сценарий. Въпреки че не могат да се справят с броя на функциите и зависимостите, които имаме в нашия набор от данни, автоенкодерите могат да се окажат доста полезни, когато разглеждаме данни от един сензор. Например, ако вземем светлинния сензор в Motion Sensor Package. От описанието на данните знаем, че докато сензорът за светлина може

да отчита стойности от 0 до 100, обикновено наблюдаваме стойности между 0 и 10 поради капака и стикерите върху опаковката на сензора. Единствените пикове в стойността възникват, когато капакът се сваля, за да се смени батерията. Въпреки че сме запознати с тази аномалия, може да има много извънредни стойности като тази, които нямат значима информация и няма нужда да ги използваме, когато обучавате който и да е модел, тъй като те просто влошават качеството на данните и резултатите. За тази цел можем да обучим автоенкодер на сензор, който бързо да прецени дали записаните данни са аномалия от стандартните записи. За самия автоенкодер променяме функцията на загуба от средна квадратна грешка на двоична кросентропия, скоростта на обучение на 0,0000001 и обучаваме 50 епохи в партии с размер 256. Полученият автоенкодер се обучава за 17,3 s и има -0,07 загуба със 100% точност.

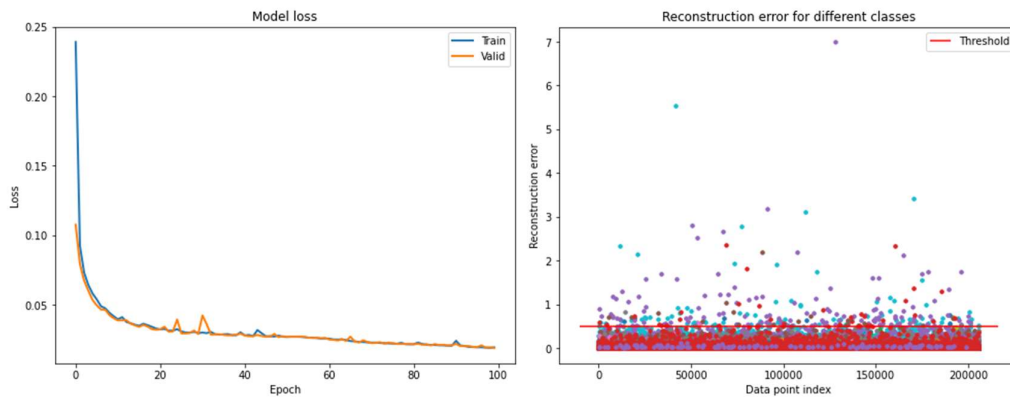
Този подход за откриване на аномалия не се очаква да доведе до значими резултати в контекста на разпознаване на връзката между различни ADL, тъй като всеки запис се обработва независимо, което ясно се вижда от предварителните резултати от обучението. Ние обаче показахме, че може да разпознава аномалии на техническо ниво за сензорите, които използваме и може да се използва като предварителна обработка към други алгоритми за почистване на всеки шум в данните, като хардуерни неизправности. За да подобрим този подход към невронните мрежи, в следващия раздел разглеждаме LSTM, който е по-добре оборудван да обработва многовариантни данни.

4.1.2 Дълга Краткосрочна Памет (LSTM)

Дългата краткосрочна памет (LSTM) е вид повтаряща се невронна мрежа (RNN), която има за цел да преодолее ограниченията на традиционните невронни мрежи при улавяне на дългосрочни зависимости в последователни данни. По дефиниция това го прави подходящ подход от автоенкодерите за целите на тази разработка. LSTM модулите се състоят от клетка с памет, входна врата, изходна врата и врата за забравяне. Клетката с памет запомня стойности за произволни интервали от време, забравящите гейтове филтрират и изхвърлят част от входната информация за всяка итерация, а изходният гейт решава коя информация от текущото състояние ще включва крайния изход. Тази селективност в изхода позволява на LSTM да поддържа полезни, дългосрочни зависимости и да прави прогнози в текущите и бъдещите времеви стъпки. Ето защо те намират приложения при откриване на аномалии (Lindemann, B. et al., 2021).

За архитектурата на модела имаме 2 кодиращи слоя, повтарящ се вектор, който репликира кодираните характеристики и връща данните в 3-измерен масив, който служи като вход за декодиращите слоеве и като такъв свързва кодирането и декодирането. Следват 2 декодиращи слоя и накрая имаме TimeDistributed слой. Параметърът `n_features` е 36 в нашия случай, а `timesteps` са погледа назад, който е 5. Използваме `adam optimizer` със скорост на обучение от 0,001, обучаваме 100 епохи с размер 256. Времето

за обучение за набора от данни hh101 е 24 минути 47,9 s с точност 0,9162, загуба 0,0194 и загуба при валидиране 0,0189. Можем също така да начертаем промяната в загубата през епохите (Фигура 7):



Фигура 7. Загуба и реконструкция на LSTM за hh101

При размер на партида от 32 и скорост на обучение от 0,0001 обучението отнема 117m 18,1s, което е 5 пъти времето за обучение на партида с размер 256. Това представлява малко подобрение на резултатите, с точност 0,9171, което е с 0,0050 точки по-добро, а загуба от 0,0229, което е с 0,0017 по-малко и подобно отклонение с точност на валидиране от 0,9149 в сравнение с 0,9144 и загуба на валидиране от 0,0229 в сравнение с 0,0252. Това не е значително подобрение, което би оправдало толкова много компютърно време и ресурси.

4.2 Извличане на асоциативни правила

Извличането на асоциативни правила (ARM) извежда смислени наблюдения на често срещани модели, корелации или асоциации от големи масиви от данни, поради което намира широко приложение в извличането на данни (Saxena, A. & Rajpoot, V., 2021). Можем да използваме същата идея за прогнозиране на поведението на човек чрез предсказване на връзките и зависимостите между различни ежедневни дейности, част от стереотип. Това може да ни позволи да разберем вариациите в рамките на деня, например стандартният стереотип за неделя може да е много различна от останалата част от седмицата. В допълнение, извличането на правила за асоцииране също е приложено в областта на откриването на аномалии (г-жа Гарги Джоши, 2014 г.), което означава, че можем да стесним проблема по същия начин, както при невронните мрежи. Най-често срещаните ARM алгоритми в тези пространства са Apriori и FPGrowth и това е, върху което се фокусираме.

Алгоритмите за извеждане на асоциативни правила очакват списък от транзакции, въз основа на който трябва да се изведе корелация на общите елементи и зависимостите във всяка транзакция. Очевидно нашите сензорни данни не отговарят на

модела. Нека разгледаме отново проблема, който се опитваме да разрешим. Ние търсим модели на поведение при възрастни хора, които са склонни да спазват установен стереотип и всякакви отклонения могат да се третират като причина за безпокойство, тъй като те могат да бъдат показателни или вредни на личността, какъвто може да бъде случаят с деменцията. От поведенческа гледна точка в този случай можем да разгледаме два аспекта на разпознаването на поведението, единият е на подробно ниво на това как се изпълняват стереотипните процедури, например можем да разбием набора от данни въз основа на подгрупата от сензори, които се задействат, когато <activity> се извършва. В дългосрочен план това ни предоставя малко полезна информация, въпреки че сме в състояние да разпознаем, че лицето е взело правилно лекарството си, не можем да разпознаем, че това се случва за втори път през последните 30 минути. Имайки това предвид, по-разумно разделяне на данните се основава на дни. В този случай разглеждаме ежедневието на човек, което се вписва повече в естествения ритъм на стереотипа. Също така можем лесно да разпознаем модели на поведение, които се случват само веднъж или два пъти седмично, и да ги отбележим като правило с достатъчна подкрепа. С ежедневието ние, разбира се, не сме в състояние да сравним седмичното разпределение на дейностите, например ако наблюдаваното лице редовно ходи на седмичен преглед за рехабилитация по едно и също време, ние не сме в състояние да разпознаем дали го е пропуснал напълно в седмица или се опита да отиде няколко пъти в седмицата. Можем да обмислим допълнителни подобрения в тази посока в бъдещи работи. В тази работа ние се опитваме да изградим ежедневен списък с времеви дейности, изглеждащ така:

```
[[ '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 Априори

Apriori е алгоритъм, въведен за първи път от Agrawal и Srikant през 1994 г., използван в извличането на данни и машинното обучение за обучение на асоциативни правила, базирано на релационни бази данни. Той е проектиран да работи с транзакционни данни

и всяка транзакция се разглежда като набор от елементи. Той идентифицира често срещани набори от елементи в набор от данни и генерира правила за асоцииране въз основа на тяхното появяване. Генерирането на кандидати в Apriori е подход „отдолу нагоре“ за разширяване на чести подмножества един по един елемент, последвано от тестване на групи от кандидати спрямо данните. Алгоритъмът се прекратява, когато не бъдат намерени други успешни разширения. За този тип генериране на кандидати е ясно, че алгоритъмът има много сканирания на базата данни и броят на генерираните подмножества е голям. В допълнение, времевата сложност на алгоритъма е експоненциална.

Въпреки това Apriori се използва широко в анализа на пазарната кошница, където помага да се идентифицират модели и асоциации в поведението на клиентите при покупка. Като се има предвид, че разглеждаме поведенчески модели в ежедневието, ако представим събраните дейности като транзакции, можем да се опитаме да използваме Apriori за генериране на поведенчески правила.

С пълния набор от данни, съдържащ също действията категоризирани като други: Other_Activity, Apriori не успява да приключи поради липса на памет и не може да произведе смислен резултат. От друга страна, с намалената версия, където пропускаме категорията Other_Activity, извличаме точно 600 правила за набора от данни hh105 с поддръжка 0.2, увереност 0.8 и повдигане 3. Например, можем да наблюдаваме ясни правила. Виждаме, че ако човека работи на масата в 16:00 часа, е вероятно също да работи на масата и в 17:00 часа. По същия начин можем да видим още няколко примера за правилата, като например човек, който се връща от излизане в 12:00 часа, вероятно ще спи на дивана в 13:00 часа. В 3-тото правило виждаме, че част от нормални стереотип на човека започва със сън до 7 часа, след което се облича, яде и готви закуска в 8 и завършва деня с гледане на телевизия поне два часа около 21:00 и 22:00 часа.

Efficient Apriori имплементацията намира 114253 правила за секунда със същите параметри на поддръжка 0,2 и увереност 0,8. Например, изглежда, че този човек обикновено отива до тоалетната в 7 сутринта и е в леглото до 23:00. Понякога човекът си тръгва за по-малко от час или някой го посещава сутрин около 10 часа, като се вижда от ниската поддръжка, това не е ежедневно събитие, но все пак се случва достатъчно често, за да бъде взет. Разбира се, ние улавяме естествената връзка между сензорните събития, като например, ако човекът се връща от банята в 6 сутринта, той се връща в леглото и заспива.

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: 45000000.000)
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)

Имайки това предвид, нека се опитаме да видим дали ефективният Apriori намира същите стереотипни процедури и какви са допълнителните открити правила, които биха обяснили увеличаването на обема на откритите правила между Apriori и ефективния Apriori. За поддръжка на 0.3 в ефективен Apriori можем да видим, че е получено същото правило: {Cook_Breakfast_08:00:00} -> {Dress_08:00:00}, но може да се види обратното правило на аруогі: {Cook_Breakfast_07:00:00} -> {Sleep_07:00:00}, като се има предвид нашето почасово обобщаване, можем да сме сигурни колко точни са тези конкретни правила. Както при Аруогі, ние не виждаме поведението от преди. Нека потърсим същото в Apriori с ефективна поддръжка 0.2. Въпреки по-големия обем на килера, веднага можем да видим същия модел на ранно сутрешно събуждане, но има още няколко правила, които корелират със съня. Тези, които следват нашето „обратно“ разбиране, ще бъдат филтрирани, например {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)
```

Фигура 8. Разпознат от Efficient Apriori стереотип за ранно ставане

Разглеждайки качеството на правилото и времената за обработка, предпочитаната реализация на Apriori е ефективната Apriori, с поддръжка 0.2 и 0.7. Въпреки че имаме повече правила като резултат, времето за обработка ни дава също достатъчна възможност да филтрираме правилата, които извличат прозрения за миналото и все пак да останем с добър набор от правила за разумен период от време (Фигура 8).

4.2.2 FPGrowth

Frequent Pattern Growth (FPGrowth) е алгоритъм за извличане на данни, насочен към извличане на правила за асоцииране в големи набори от елементи, въведен за първи път в началото на 2000 г. от Jiawei Han, Jian Pei, Yiwen Yin и Runying Mao (Han, J. et. al., 2004). Основополагащата концепция на алгоритъма е генерирането на дърво с често срещани модели (FPTree), което е компресирано представяне на базата данни с набор от елементи, което също така записва връзката между наборите от елементи. Това позволява на FP Growth да преодолее недостатъците на Apriori и да обработва големи масиви от данни, тъй като те се съхраняват в компактно дърво, което можем да

обхождаме ефективно. Както при Arriori, нека сравним двата типа набори от данни, първо редуцирания, където премахваме всички категории „Other_Activity“ и след това набора от данни за целия ден с тях. Освен това използваме същите параметри като за Arriori, което означава поддръжка от 0,2 и увереност от 0,8 и същия набор от данни – hh105.

За редуцирания набор от данни, общото време за работа е 59,2s и ние извличаме общо 88918 правила. Някои примери включват същите поведения, които видяхме и с Arriori, че изглежда има напускане или посещение на дома, което се случва в рамките на един и същи час сутрин в 10:00, но също така съчетано с няколко варианта на заспиване и сутрешният стереотип, примери 1-3. Освен това можем да видим повече правила, свързани със стереотипа на хората, като например 11 часа е времето, когато те обикновено излизат, а също и вечерите, когато вечерят около 18:00, те също е вероятно да са в леглото и да спят 23:00 часа.

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 Сравнение на невронни мрежи и алгоритми за извличане на асоциативни правила при прилагане за анализ на човешкото поведение

Досега за обучението и оценката на различните подходи използвахме малък поднабор от 1-3 набора от данни от набора от данни CASAS. С този формат на данни имаме около 31 различни набора от данни и къщи. Всеки набор от данни съдържа измервания за 3-6 месеца в домакинства с един човек или 2 души. С описанието на всеки набор от данни не можем веднага да идентифицираме броя на хората, които са живели в дома. Можем да приемем втори обитател в по-големи домове, като набора от данни hh121, вижте сензорната карта по-долу. Тук виждаме не само размера на апартамента, но и две отделни зони на активност от двете страни на леглото. Но дори и това е чисто предположение и може да се окаже, че тук живее един жител. Имайки предвид това, възможно е да изберем набор от данни, където не може да бъде намерена стереотип не поради самите алгоритми, а защото записваме повече от един човек. Разбира се, не можем да приемем, че стереотип е налице дори в домакинствата с един жител.

Нека сравним някои от наборите от данни. На фигура 9 по-долу можем да видим няколко сензорни карти за наборите от данни, а именно първите 15 набора от данни.

Можем да видим, че някои домове са много сходни в оформлението си, а други се различават значително. По принцип, когато категоризираме дейностите, ние не записваме сензорите, предоставили входните данни, което означава, че ако има две бани и само един обитател, вземането на душ във всяка от тях ще бъде прекодирано като Лична_хигиена. От друга страна, когато се използват вектори на характеристики, това ще се счита за две отделни събития. Такова разграничение би имало смисъл само в избрани няколко случая и е по-вредно за общия случай на приложение. От тази гледна точка можем да очакваме по-лоши резултати от тези домове



Фигура 9. Сензорни карти за набори от данни hh101-hh115

За да потвърдим това очакване, нека разгледаме накратко размера на различните апартаменти, общия брой стаи, които имат и евентуални дублирания, като две и повече спални или бани. В таблицата по-долу сме обобщили тази информация, преброяваме броя на стаите, в които имаме сензори, измерващи ADL, например кухня, всекидневна, баня, спалня, дрешници, офис и т.н. Не всички домове имат всички видове стаи. Това, което също представлява интерес, е обемът на данните, които използваме за обучение, за тази цел в таблицата по-долу можем да видим и общия брой необработени записи във всеки набор от данни, както и размера на всеки вектор на характеристиките.

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

Таблица 1. Обем и композиция на набори данни hh101 – hh115

4.3.1 Невронни Мрежи

Нека първо да разгледаме автоенкодерите, в предишните глави разгледахме няколко подхода към автоенкодерите, докато най-добрата точност беше постигната с помощта на векторите на характеристиките, това дойде на цената на много висока загуба. Тъй като в този случай винаги виждаме 100% точност, за целите на сравнението има смисъл да разгледаме резултатите от обучението на автокодера, използвайки необработените данни от сензора и обучение за 50 епохи с размер на партидата 256. Като цяло, не моделът успя да надмине марката за точност от 50%, с много големи загуби при обучението и валидирането.

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

Таблица 2. Резултати от обучението на автоенкодер с MSE

Ако сравним таблицата с резултатите от обучението с таблицата за състава на набора от данни, можем да видим някои очаквани тенденции. Например, нашият модел

за набора от данни hh111 е почти най-лошият по отношение на точността и загубата, но това също е най-големият ни набор от данни и големите набори от данни не се обработват толкова лесно от автокодери. Естествено най-малкият набор от данни hh15 показва най-малката загуба и най-високата точност, това не е тенденция в другите набори от данни, тъй като вторият най-малък набор от данни hh110 пада под средното ниво на точност, а най-ниската точност се наблюдава в hh108, който е 3-ият по големина набор от данни. За да обобщим, въпреки че лошото представяне на модела на автокодера не е неочаквано поради естеството на набора от данни и проблемната област, която разглеждаме, можем да използваме резултатите по-долу като базова линия за анализиране на по-сложни RNN, като LSTM.

Имайки това предвид, нека направим подобно сравнение за LSTM, тук използваме подхода на вектора на характеристиките, тъй като имаше ясно предимство. Задаваме размера на партидата на 32, скоростта на обучение на 0,0001 и обучаваме за 50 епохи, използвайки функцията за загуба на средна квадратна грешка. Той се представя значително по-добре от автокодера с точност, обикновено средно около 88% и загуба около 0,03, както можете да видите в таблицата по-долу.

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

Таблица 3. Резултати от обучението на LSTM при hh101 – hh115

LSTM няма същите ограничения за големи набори от данни, каквито има автокодерът. Можем да видим това, когато сравним резултатите от обучението за hh111. Това е най-големият набор от данни и има най-висока загуба и най-ниска точност в модела на автокодера. LSTM има една от най-ниските загуби и се представя сравнено добре с другите набори от данни. За разлика от автокодера, можем да видим по-последователен резултат в точността и загубата на моделите на нови набори от данни, което би направило този подход по-последователен, когато се прилага към нашия случай на използване на AAL. Наборът от данни hh110 имаше прилични резултати с модела на

автокодера, поради малкия си размер. В случая на LSTM можем да видим, че той има най-ниската точност и бързо време за обработка, в съответствие с нашите очаквания. Както виждаме, по-големите набори от данни поддържат по-добър модел на поведение LSTM. Което би означавало, че колкото по-дълго се използва нашата AAL система за събиране на данни, толкова по-добри ще бъдат прогнозите за поведението.

Разглеждайки времето за обучение, автоенкодерите, обучаващи се на пълния набор от данни на сензора, имат време за работа от 30 минути до един час, в зависимост от набора от данни. LSTM показва подобни времена за размер на партида от 256, с намаляване на размера на партидата няма значително подобрене в точността на модела, но наблюдаваме, че времето за обработка се умножава с коефициент на намаляване на размера на партидата. За тази селекция от данни експериментите откриха оптимален компромис между резултатите и времето за обучение в параметрите, дефинирани по-горе, с размер на партидата 256 и 50 епохи. В следващия раздел по подобен начин анализираме подходите за извличане на правила за асоцииране.

4.3.2 Извличане на асоциативни правила

Разгледахме производителността на алгоритъма на невронната мрежа в наборите от данни hh101 до hh115. В този раздел правим подобно сравнение за подходите за извличане на асоциативни правила. В раздела Резултати от обучението за Apriori вече установихме, че Efficient Apriori има също толкова добро качество на извлечените правила, но за част от времето за големи набори от данни. Следователно в този раздел използваме ефективната реализация на Apriori, когато оценяваме производителността и резултатите на Apriori.

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

Таблица 4. Резултати от обучението на Apriori на hh101 – hh115

Нека да разгледаме резултатите от FPGrowth, в таблица 4 можете да видите обема на правилата за набор от данни, който е извлечен. Обучението е по-бавно от Apriori,

сравнимо по време само за hh103 и hh105, някои набори от данни отнемат няколко минути, други часове, а за hh107 и hh108 ресурсите на машината за обучение не са достатъчни, което води до изчакване и срив на паметта. Можем да наблюдаваме същите клопки като при Arriori, поради обработката на данни, която агрегира събития на почасова база. За всеки набор от данни виждаме едни и същи процедури, идентифицирани от двата алгоритъма, но също и правила, които са уникални за всеки подход.

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

Таблица 5. Резултати от обученето на FPGrowth върху hh101 – hh115

В заключение, автоенкодерите не могат да се справят с многоизмерната сложност на проблемното пространство. Независимо от това, това е много бърз и надежден подход за откриване на неизправности на сензора и извънредни стойности, тъй като отнема по-малко от минута за обучение и има добри резултати. Неизправностите на сензорите трябва да бъдат премахнати от данните, тъй като се интересуваме само от поведенческите аномалии в ежедневието на човека, а не от записите на извънредни устройства. Така че, макар да не е достатъчен сам по себе си, той би бил добър спътник за предварителна обработка на който и да е от алтернативните алгоритми.

LSTM е в състояние да разбере стереотип и да открие отклонения. Въпреки това, поради нашата предпоставка за създаване на общ подход към обучението с предварителна обработка, която в същността си е еднаква за всеки набор от данни и не подлежи на допълнителна предварителна подготовка. LSTM не може да постигне точност над 90% средно, което води до много фалшиви положителни и фалшиви отрицателни резултати. Работим с плъзгащ се прозорец от 30 събития, поради което не можем да идентифицираме кое от записаните събития е предизвикало извънредното отклонение. Следователно, тъй като няма начин да се проследи обратно причината за предупреждението за отклонение или начин да се обясни защо е било повдигнато, ние не можем да преценим всяко повдигнато събитие и да смекчим грешките. Освен това,

плъзгащият се прозорец с 30 събития ограничава способността на модела да намира корелация между поведенията през целия ден.

В случая на Apriori и FPGrowth и двата алгоритъма успяват да създадат разумни набори от правила, които имат основно разбиране на съществуващите стереотипни процедури. Естествено, те не са перфектни и пропускат определени поведения при конструирането на набора от правила, което поражда много повече извънредни събития. За разлика от невронните мрежи, тук имаме възможност за проследяване и разбиране на повдигнатото събитие въз основа на събитието, което го е задействало и е сравнително лесно да се коригира и добави или премахне правило, което не отразява правилно реалността на стереотипа на този човек. Този подход също би бил по-лесен за интегриране с нашия прототип OpenRemote, тъй като вече имаме концепция за правила и база данни от правила. Като цяло ефективното внедряване на Apriori успява да намери повече и по-добри правила в сравнение с FPGrowth, но са необходими допълнителни тестове и изследвания, за да се потвърди, че няма случаи, в които стереотипа, дефинирана от FPGrowth, е равна на тази в Apriori.

Въпреки че нито един от двата подхода не е перфектен, използването на ARM-базирани алгоритми за откриване на извънредни стойности в поведението на човека е най-рентабилната опция, която също има разбираеми резултати и лесно смекчава техническите извънредни стойности без необходимост от предварителна обработка. Освен това е подходящ и лесен за интегриране с прототипа OpenRemote. Между двата алгоритъма Apriori определено е по-бърз и има също толкова добро генериране на резултати. Въпреки това трябва да оценим допълнително дали в някои случаи побавният FPGrowth намира стереотипни процедури, които Apriori пропуска.

5 ЗАКЛЮЧЕНИЕ

В тази работа разгледахме често срещаните пропуски, идентифицирани от предишни научни трудове разглеждащи системите за подпомаган живот от обкръжаващата среда с общо предназначение, по-кратко наричани AAL системи, а именно устойчиво осигуряване на защита и сигурност на данните в разрастваща се система. Те са фактори, които възпрепятстват широкото разпространение и използване на AAL системите. Чрез преизползване на най-съвременния мидълуер, който е наличен и се поддържа в платформи с отворен код, ние създадохме прототип, съобразен с очакванията за съобразена цена на системата и разходите по нейната поддръжка. Идентифицирайки функционалните изисквания на ОРЗД, ние успяхме да предложим имплементация, която удовлетворява ОРЗД и съответно осигурява защита на личните данните за своите потребители.

По подобен начин видяхме, че алгоритмите за машинно самообучение имат приложение при разбирането на човешкото поведение в контекста на ежедневните дейности. В обхвата на съобразена с цената AAL система ние разчитаме на вече наличните сензори и интелигентни устройства, за да създадем поток от данни от събития, които описват ежедневните дейности на лицето, използващо системата. Това означава, че нямаме контрол върху наличните източници на данни и сензори и не можем да персонализираме данните толкова, колкото би трябвало да получим оптимални резултати от обучението. Невронните мрежи са силно засегнати от това ограничение, тъй като не могат да постигнат достатъчно добра точност, за да бъдат приложени по продуктивен начин в AAL система. От друга страна моделите, търсещи асоциативни правила, са гъвкави по този въпрос, тъй като техните четливи прогнози и изводи позволяват на крайния потребител, в случая този който се грижи за човека, който системата наблюдава ги разбере и на свой ред да игнорира фалшивите положителни резултати.

Комбинирайки прототипа и моделите, ние предлагаме надеждна за бъдещето AAL система, която би могла да намери широко приложение, поради своите мощни персонализации, ниска цена на поддръжка и потребителско включване и добра основа за приложения за реална употреба, поради съответствието си с основополагащите принципи като поверителност на данните и активиране на сигурността на данните.

5.1 Дискусия

Първата цел, която си поставихме под формата на изследователски въпрос, беше създаването на съобразена с разходите система за поддържан живот от обкръжаващата среда (AAL) и в тази работа създадохме прототип, съобразен с разходите, за разлика от неговите предшественици:

- Системата е базирана на междинен софтуер с отворен код, който възлага на общността по-голямата част от необходимата поддръжка на системата в крак с нововъзникващите технологии и най-добри практики, както се изисква от GDPR, особено относно сигурността и защитата на данните. Това също така намалява разходите за разработване на системата, възможността за техническа поддръжка на повече и разнообразни механизми и протоколи за свързване на устройства и намалява разходите за системата, тъй като OpenRemote има силно активна общност от потребители и разработчици, което позволява бързо идентифициране и разрешаване на дефекти и уязвимости при продуктивно широкомащабно използване на системата.
- Прототипът на системата работи като набор от docker контейнери и поради това продуктивната система може лесно да се инсталира за няколко минути, тъй като персонализирането на системата и нейните сфери се постига във файла за

съставяне, съдържащ конфигурацията на контейнера. Същото позволява бързо и лесно преместване и редовно архивиране на системата. Освен това можем лесно да въведем допълнителна устойчивост на системите, като използваме най-добрите практики на Kubernetes и използваме цикъл на съгласуване, за да гарантираме, че контейнерите винаги работят здравословно и са налични.

- За обучението на алгоритми за машинно обучение има праг от ресурси, над който не можем да оптимизираме. В случая на прототипа на системата имаме добър компромис между полезността на модела и изчислителните ресурси, като използваме алгоритми за извличане на правила за асоцииране, които лесно се интегрират с архитектурата на OpenRemote и много оптимизирана реализация на алгоритъма Apriori. За разлика от това, оценените невронни мрежи са изразходвали повече от два пъти повече ресурси в някои случаи при предварителна обработка и обучение с получени по-лоши резултати.
- Същата логика за оптимизиране на разходите е налице и при повторно обучение на модела. Цената за извършване на това за невронните мрежи е много по-висока от цената за повторно обучение на базираните на ARM алгоритми, не само поради изразходваните ресурси, но и защото прототипът на OpenRemote, който проектирахме, позволява на човека който се грижи за подпомагания от системата, ръчно да коригира или добавя нови правила, което позволява моделът да бъде обучаван по-рядко и да поддържа по-висока полезност и точност на предупрежденията за извънредни стойности за по-дълго време.
- Цената на първоначалното включване към системата се намалява чрез използването на вече съществуващи интелигентни домашни устройства в дома на лицето. Тъй като OpenRemote има архитектура, която поддържа най-често срещаните протоколи за свързване, е възможно да се свържат съществуващи устройства и първоначалната инвестиция за закупуване, настройка и свързване на сензори и устройства е намалена.
- OpenRemote има концепция за сфери, която позволява едно копие на междинния софтуер да бъде споделяно между няколко потребители в близост, в зависимост от типа на сензорите. За безжични сензори можем да използваме локален хъб за събиране и транспортиране на данни до централната инстанция. Това от своя страна позволява цената на системата и изчислителните ресурси да бъдат разделени между няколко потребители.

Второ, сред многото съществуващи прототипи на AAL видяхме, че има два основни въпроса, останали отворени: сигурността на системата и поверителността на данните, за които теоретизирахме, че може да са една от причините, поради които няма толкова широко приемане и лесно достъпни за широката публика AAL системи. В тази работа не разгледахме сигурността извън изискванията на ОРЗД и по-скоро използвахме

резултатите от платформата на междинния софтуер. Ние обаче поставихме специален фокус върху поверителността на данните и създадохме система, която може уверено да твърди, че е съвместима с ОРЗД. От нашето проучване на ОРЗД изведохме 15 основни функционални изисквания, които спазихме.

- Подчертахме важността на правилното участие на правен екип в спазването на поверителността на данните. Те са тези, които определят дали обработката е честна, законна и прозрачна; какви са законните основания за обработка; за кои данни и до каква степен е приложимо всяко искане на субект на данни (DSR); има ли резултати от автоматизирана обработка на лични данни и като такива подлежат ли на ОРЗД и т.н. В тази работа ние се стремим да докажем техническата осъществимост и съответствието на такава система, поради което не сме ангажирали правен екип, което би било предпоставка, за да направим системата продуктивно достъпна.
- Обработка на данни в ОРЗД е всяка операция или набор от операции, които се извършват върху лични данни или набори от лични данни, независимо дали чрез автоматизирани средства или не. За да се гарантира защитата на това, данните трябва да бъдат сведени до минимум, което означава, че обработката е адекватна, подходяща и ограничена до необходимото. В прототипа гарантираме минимизиране на данните на всички нива на транспортиране на данни. По време на събирането ние записваме само минимално необходимата информация, събрана от устройството. За целите на предупрежденията се изпраща само четимо от човека изявление с необходимата информация и отделено от действителните сензори и данни. Моделът за машинно самообучение има директен достъп само до таблицата с атрибутни данни в изолирана мрежа. Освен това данните се съхраняват в отделен метод, тъй като подробностите за атрибутите се съхраняват отделно от стойностите и двете са изключени от базата данни с правила.
- Тъй като данните са изолирани, в случай, че лицето правилно се позовава на правото си да бъде заличен, както е определено от правния екип, действителното изтриване е въпрос на заявка към базата достъпваща няколко таблици и актуализация на системните резервни копия, за да се гарантира, че изтритите данни не могат да бъдат възстановени. По подобен начин имаме няколко начина да осигурим ограничена обработка, както е описано в ОРЗД, най-лесният от които е да позволим на данните да преминат през системата, без да ги записваме.
- По темата за сигурността, оторизацията и удостоверяването ОРЗД е умишлено неконкретен, тъй като концепциите се развиват непрекъснато. Вместо това изисква системите да използват най-модерната технология. От тази гледна точка ние се възползваме от разчитането на междинен софтуер с отворен код, с активна поддържаща общност. Докато имаме активна общност и широко разпространени

продуктивни приложения, ще има поток от иновации и възприемане на технологии. Независимо от това, за да претендираме за наистина сигурна система, ще трябва да разгледаме по-задълбочено и да анализираме всяко устройство и тип свързаност, всички методи за комуникация между системи и устройства, методи за оторизация и удостоверяване, да анализираме уязвимостите, да инвестираме в моделиране на заплахи и т.н.

- Освен това избраната услуга за изпращане и получаване на незабавни насочени известия и съобщения Pushsafer е разработена в Европейския съюз, което гарантира, че можем да спазваме и ограничението, че данните в ЕС не са достъпни или обработвани извън ЕС. Pushsafer също така се фокусира върху осигуряването на защита на данните, което означава, че интеграцията не въвежда нови ограничения.
- При настройката на архивиране и настройките за висока наличност можем да разчитаме на вече установените най-добри практики за управление на Docker и контейнери. Можем да конфигурираме и използваме инсталация на Kubernetes, да използваме docker swarm, да създадем персонализирано решение и т.н. Подходът ще зависи от обема на данните и потребителите в инстанцията на системата и прогнозираната надеждност на всяка инстанция на системата.
- В ОРЗД данните са достъпни и се разпространяват на принципа „необходимост от информацията“. В нашата AAL система е вероятно да има и медицински чувствителни данни, които трябва да бъдат допълнително защитени. Въз основа на съществуващите парадигми на OpenRemote, базирани на роли и разрешения, създадохме система за управление на идентичността, която категоризира потребителите въз основа на връзката им с основния потребител на системата и ограничава достъпа им до различни подмножества от данни.
- Съображенията на ОРЗД гласят, че субектът на данни има право на „обяснение на решението, взето след [алгоритмична] оценка“, което действително можем да постигнем чрез използване на модели, базирани на правила, за разлика от невронните мрежи, където единственото налично обяснение, е че сигнализираното отклонение в поведението е резултат от разминаване с предсказанието на невронната мрежа и това също е повлияло на предпочитания алгоритъм.
- Когато се обсъжда значението на ОРЗД върху моделите за машинно обучение, основно безпокойство е правото да бъдеш забравен, което поставя въпроса дали данните, използвани за обучение на модела, също да бъдат премахнати от модела и възможно ли е моделът за машинно обучение да изложи лични или медицински чувствителни данни на неоторизирани хора. За нашия прототип ние опростихме и двата отговора, като обучаваме персонален модел на човек, за разлика от общ

модел, който използва данните на всеки потребител съвместно. Това е не само поради ОРЗД, но и поради естеството на проблемното пространство, ние търсим силно специфични и персонализирани стереотипни поведенчески модели и техните отклонения.

И накрая, в началото попитахме дали прилагането на модели за машинно самообучение, обучени върху събраните данни, ще има положително въздействие върху функционалността на системата, без да компрометира цената на системата. Последното се свежда до изчисляване на съотношението и компромиса между коректността и полезността на модела и разходите за обучение, поддръжка и ресурси за него, метод познат като “разходи-ползи” в икономиката.

- Потвърдихме отново приложението на алгоритмите за невронни мрежи тип автоенкодер за разпознаване на извънредно техническо поведение в поток от времеви редове от цифрови сензорни данни. Предложихме да се използва автокодер за предварителната обработка на всички данни, за да се гарантира, че техническите отклонения в данните от сензора са отделени от поведенческите отклонения на лицето, които се стремим да идентифицираме.
- Показахме, че е възможно да се прилагат невронни мрежи тип LSTM за разпознаване на модели в човешкото поведение и като резултат да се откриват отклонения в него. Нашите експерименти показаха, че един използваем LSTM модел изисква предварителна обработка, персонализиране на параметрите, почистване на данните и обща инвестиция във време извън това, което сме готови да инвестираме на човек в AAL система, съобразена с разходите. Въпреки това, дори и без тази инвестиция, моделът LSTM показва прилично разбиране на събитията, които естествено следват едно след друго. Едно ограничение за това е разпознаването на модели, които обхващат целия ден, седмица, месец и т.н. Това би било възможно да се постигне чрез LSTM с допълнителна модификация на предварителната обработка и самия алгоритъм.
- По подобен начин идентифицирахме и илюстрирахме приложението на алгоритми за извличане на правила за асоцииране за същия проблем: идентифициране на отклонения в поведението на човека въз основа на предварително разпознати ежедневни стереотипни процедури. В този случай стереотипа се съхранява под формата на набор от правила. Сравнихме алгоритъма Apriori и FPGrowth върху различни набори от данни, за да преценим качеството на резултатите и приложимостта и показахме, че сме в състояние да извлечем ежедневен стереотип, свързваща събития през целия ден под формата на правила за асоцииране.
- За да постигнем полученото описание на стереотип, предложихме подготовка на данни за ARM алгоритми, която трансформира всички ежедневни събития в

транзакция, която може да се използва от Apriori или FPGrowth в този случай за разпознаване на общите черти между дните.

- В хода на проучването и внедряването открихме и валидирахме оптимизирана имплементация на Apriori, която има по-добри резултати от стандартната реализация на FPGrowth.
- В нашия случай, когато търсим използваеми резултати при по-ниски разходи за обучение и управление на жизнения цикъл на модела, сравнихме алгоритмите за обучение на невронни мрежи с тези за извличане на асоциативни правила, за да оценим съотношението цена/ползност и заключихме, че по-добрият вариант за конкретния прототип и случай на приложение е подобрената имплементация Efficient Apriori на Априори. Той позволява на човек да смекчи грешките в повдигнатите сигнали, поради своята обяснимост, а също и наличният коефициент на увереност, с която се повдига всеки сигнал.
- Едно ограничение, което трябва да се разпознае в този раздел, е граничното условие, че търсим модели на поведение, възникващи в рамките на един обичаен ден, игнорираме празници, седмични, месечни и годишни стереотипни процедури. Това ще изисква допълнително подобряване на предварителната обработка на данни при двата подхода, докато бъде открит оптимумът. По-специално, тук трябва да вземем предвид и фактора цена. Каква е цената за разработване на модел за машинно самообучение, който разпознава, че веднъж годишно на рождения си ден човек има специален стереотип в сравнение с простото добавяне на това обстоятелство като правило в набора правила? Как различаваме годишен стереотип от отклонение в поведението и колко години/месеци данни ще са необходими за това?

Въз основа на изследователските въпроси формулирахме следната хипотеза: *Можем да преизползваме междинен софтуер за интелигентен дом с отворен код, за да създадем AAL система, съобразена с разходите и поверителността на данните, да я разширим с алгоритми за машинно самообучение по полезен начин и да докажем, че алгоритмите за извличане на асоциативни правила (ARM) могат да се използват за разпознаване на човешкото поведение и те биха били по-добрият избор в сравнение със стандартните подходи за откриване на отклонения, тъй като те като цяло са по-евтини, по-лесно изпълняват изискванията за поверителност на данните и имат обясними резултати.* От аргументацията по-горе можем ясно да видим, че нашата хипотеза е потвърдена като част от това изследване с идентифицирани няколко ограничения и крайни случаи. Въз основа на тях можем да разширим изследването и допълнително да подобрим резултатите.

5.2 Приноси

Въз основа на гореописаните изследвания и резултати можем да формулираме приносите на тази дисертация, разделени в три отчасти припокриващи се категории:

Научни приноси

1. Аналитичен обзор на състоянието в областта на AAL системите, съществуващите в момента видове системи, често решавани проблеми и методология.
2. Аналитичен обзор на ОРЗД и обобщение на функционалните изисквания, които AAL системата трябва да изпълни, за да гарантира поверителността на данните. Същото може лесно да се преизползва и приложи към други видове системи.
3. Модификации и подобрения на алгоритми за машинно самообучение с цел разпознаване на модели на човешкото поведение. Сравнение на производителността, точността и приложимостта в предложената AAL система на невронни мрежи и алгоритми за извличане на асоциативни правила.
4. Разработване на методология за създаване на AAL система с общо приложение.
5. Откриване и валидирано разпознаване на оптимизирана реализация на Argioi, която предоставя по-бързи и еднакво точни резултати в сравнение с FPGrowth за разпознаване на поведенчески модели.
6. Предложен е метод за трансформиране на прекодирани като времеви редове ежедневни дейности в етикетирани ежедневни транзакции с цел генериране на правила.

Научно-приложни приноси

1. Предложение за подобрена, в сравнение с предшествениците си, по отношение на цена и сигурност на данните, AAL система, която използва вече съществуващи сензори и устройства в дома на човек, за да създаде модел, който разпознава човешките поведенчески стереотипи и може да предупреждава за отклонения от тях.
2. Предложена и внедрена архитектура на AAL система, подобрена с алгоритми за машинно самообучение, която гарантира съответствие с поверителността на данните, като работи като контейнерно решение в изолирана мрежа.
3. Експерименти върху приблизително 15 набора от данни от колекцията от набори от данни CASAS, които водят до заключението, че алгоритмите на ARM са по-рентабилни и по-лесни за поддръжка, когато се прилагат към AAL система за предсказване на поведенчески модели.
4. Концепция и изисквания за създаване на AAL системи, фокусирани върху общата цена на внедряването, дългосрочната поддръжка и разходите на човек.

Приложни приноси

1. Реализация на прототип на AAL система, разширена с модели за машинно самообучение, разпознаващи стереотипно поведение, базирано на събраните от системата данни.
2. Прототип, изпълняващ функционалните изисквания на система, съвместима с GDPR.

5.3 Бъдещи изследвания

Има няколко посоки, в които можем да подобрим и развием текущите изследвания: прототипът на самата система, валидирането на резултатите в реалния свят и подобряването на моделите:

- В тази работа представихме архитектура на системата и внедряване, независими от подхода за машинно самообучение, който след това ще бъде интегриран, тъй като няхме ясна яснота кой ще бъде този подход. Сега, когато знаем, че по-подходящ е Argiogi, има смисъл да подобрим интеграцията му в системата. А именно чрез разширяване на базата от данни на Argiogi с параметър за доверие и интегриране на получения набор от правила от Argiogi в базата от данни на OpenRemote. Това трябва да се направи по начин, който не отменя ръчно предоставените правила при повторно изпълнение на алгоритъма, но също така не запазва остарелите правила от предишни Argiogi итерации.
- Прототипът разчита на умните устройства, които вече са налични в дома на лицето. В днешно време виждаме спад в наличността на потока от данни от тези устройства. Данните за интелигентните преносими устройства обикновено се консумират само чрез приложения и отскоро често изискват месечен абонамент, което е контрапродуктивно за цената на системата. Това не е така за всеки тип устройство и в нашия прототип няхме данни от преносимо устройство. Следователно, логична следваща стъпка би била да се проучат най-често срещаните “умни” устройства в дома на човек, да се класифицират и да се анализира наличността на техния суров поток от данни. От това изследване бихме могли да приложим алгоритъма Argiogi и да изведем какъв е минималният брой и тип устройства, които са необходими за правилното разпознаване на стереотип и обучение на модел.
- Прототипът използва предварително записани данни на хора, за които не знаем нищо. Основната целева група на нашата AAL система са възрастни хора и особено хора с деменция, при които предполагаем, че е налице стереотип. Съответно като следваща стъпка е редно да приложим системата и алгоритъма Argiogi върху данни, събрани за човек с деменция. Значителна част от това изследване би било също така да се идентифицира кои отклонения в поведението

водят до момент, при който лицето е объркано, изгубено или се нуждае от помощ. Можем да направим това с вече наличен набор от данни, но в този случай следващата стъпка би била да активираме устройствата в реална къща на възрастен с деменция и да потвърдим първоначалната хипотеза там. Ограничение на окончателното валидиране от реален човек би бил фактът, че понастоящем хората с деменция не са независими и рядко са оставени да се грижат сами за себе си, следователно нашата система ще записва винаги поведението на възрастния с деменция и неговия помощник.

- Има две очевидни оптимизации в Argioi, които трябва да бъдат проучени. Текущият Argioi третира всички действия от деня като случили се едновременно, поради подготовката на данните, а в нашия случай те са последователни. Това е важен факт, тъй като повече от половината време се изразходва в търсене на правила, които са имали резултати в миналото, което изразходва както ресурси за обучение, така и ресурси за последваща обработка за почистване на набора от правила. Второто е да се създаде по-интелигентен прозорец за агрегиране, който може да обработва събития, които се записват близо до началото на часа.
- По подобен начин Argioi може да бъде подобрен, за да обхване стереотипни процедури, които обхващат седмици, месеци и години. Отново трябва да внимаваме за разходите за обучение на алгоритъм за разпознаване на редки или изключителни правила, които могат да бъдат добавени ръчно по-късно. Като цяло Argioi показва голям потенциал за оптимизация и приложение в областта на разпознаването на модели в човешкото поведение.

ДЕКЛАРАЦИЯ ЗА ОРИГИНАЛНОСТ НА РЕЗУЛТАТИТЕ

Декларирам, че настоящата дисертация съдържа оригинални резултати, получени при проведени от мен научни изследвания. Резултатите, които са получени, описани и/или публикувани от други учени, са надлежно и подробно цитирани в библиографията.

Настоящата дисертация не е прилагана за придобиване на научна степен в друго висше училище, университет или научен институт.

Подпис:

ЛИТЕРАТУРА

- 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.00000000000034374
- 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 rhythm analysis: 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/S1516-4446200300010000