Comparing supervised machine learning algorithms for client-specific care plans

Author: Cindy Pistorius
MSc. Thesis – Industrial Engineering and Management – October 29, 2023
Supervisors: Dr. S. Rachuba, Dr. A. Abhishta – Supervisor Ecare: N. Letwory

ABSTRACT

Creating client-specific care plans is a complex task where mistakes are easily made, especially by inexperienced caregivers. Supervised machine learning models can support these caregivers by suggesting relevant actions for client-specific care plans. In this research, four different algorithms (ML-KNN, MLP, BR-SVM and BR-RF) are compared to determine whether supervised machine learning can provide actions based on scalar data of EHRs of clients. This is a multi-label classification problem. These algorithms suggest actions for clients with one of three illnesses: heart failure, dementia and diabetes. The BR-RF has the highest weighted F1-score (0.78 for heart failure, 0.67 for dementia and 0.63 for diabetes), precision (0.89 for heart failure, 0.84 for dementia and 0.84 for diabetes) and recall (0.53 for heart failure, 0.62 for dementia and 0.63 for diabetes) on the test set. Furthermore, the created care plans of the BR-RF are compared to experienced caregivers' care plans. The comparison shows that there is promise in the models, but that they are not performing well enough. Out of eight caregivers, five preferred the model's care plans for heart failure clients, one selected the model's care plan for a client with dementia and none chose the model's care plan for clients with diabetes. The models cannot currently be implemented in a real-life situation based on the values of the performance metrics and the results of the comparison between a model's and an experienced caregiver's care plans.

KEYWORDS

Suggesting care plans; home care; multi-label classification; supervised machine learning.

1 INTRODUCTION

Approximately 585,000 people in the Netherlands use home care services regularly [1]. All these people (from now on called '*clients*') need a personal care plan drawn up by caregivers to ensure they receive the personal care they need. Multiple organisations provide platforms to monitor the Electronic Health Records (EHRs) to support the creation and registration of these care plans. One of these organisations is Ecare, which offers a platform called '*PUUR.*', an environment where caregivers can create and keep track of the individual care plans of every client. These care plans are designed using the 'Omaha Classification System', which looks at different predefined areas related to a specific client's well-being to define actions for the care plan. Unfortunately, Ecare notices that caregivers find it challenging to apply the Omaha system correctly. Before we elaborate on the problem description, we explain PUUR and the Omaha System to provide the needed background information.

1.1 PUUR.

The idea of PUUR is to give caregivers the freedom and responsibility to register the EHR of a client themselves by following a set of easy steps. The advantage of PUUR with respect to other EHR platforms is that caregivers keep track of information during all the steps of a client's EHR, from registering a client to evaluating and keeping track of appointments. Caregivers are also flexible in changing a care plan or adding new information when needed. They keep track of the following items to ensure a complete EHR for each client:

- **Personal information:** A client's personal information is stored (e.g. date of birth, address and general practitioner).
- **History:** The important activities of a client can be saved here on a timeline. These activities do not have to be healthcare-related, but can also cover personal moments. The client and caregiver can both view and adjust this timeline.
- **Care moments:** The caregiver can add and keep track of actions in the form of a planning and notes.
- **Main dossier:** This is the most important item for this research. The main dossier creates assessments and the main care plan for each client. These are created using the Omaha System, which is explained below. How Omaha is translated into PUUR is explained in Section 1.3.

Caregivers must justify each action to ensure quality, which is realised by applying the Omaha System.

1.2 The Omaha System

The Omaha System [3, 4] is a classification system to record clients' health status, actions, and (medical) measurements. It consists of three main components: the problem classification scheme, the intervention scheme and the problem rating scale for outcomes. These three components are briefly explained below. Figure 1 shows the flow of the schemes in relation to each other.

- **The problem classification scheme:** This scheme consists of 42 health concepts that are divided into four domains: Environmental, Psychological, Physiological and Health-related behaviours. Every concept is described with a unique set of signs and symptoms that are either present or not present.
- **The intervention scheme:** This scheme is used to determine specific actions for each health concept. The scheme comprises four levels: problem, category, target and care description. The problem level consists of all the concepts of the problem classification scheme. The category level is made up of four action types: teaching/guidance/counselling, treatments and procedures, case management and surveillance. The target level consists of 76 terms that specify what a caregiver needs to do to

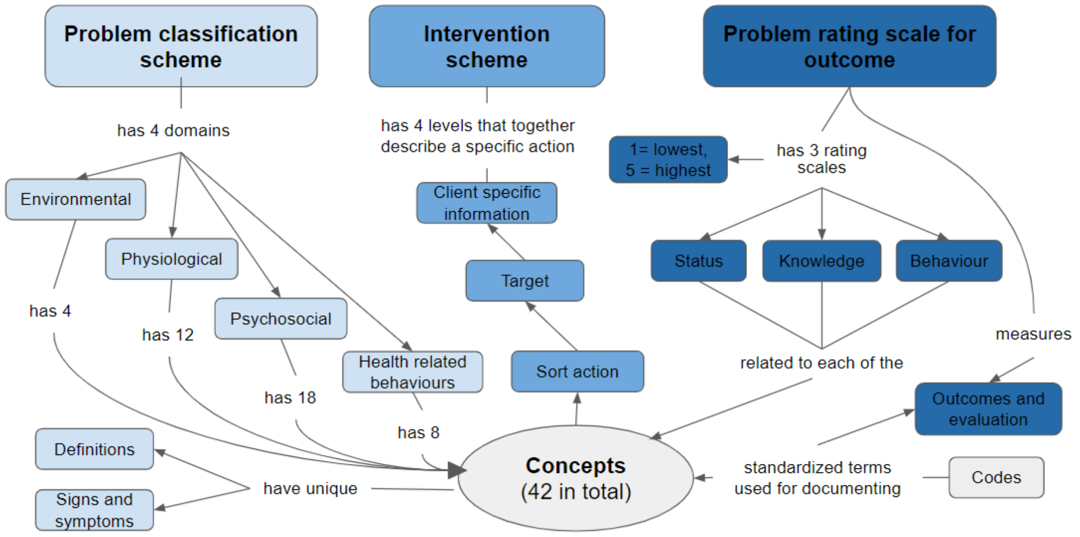


Figure 1: Flowchart of the Omaha system [2]

improve a concept. The care description allows for providing extra-textual information.

- **The problem rating scale for outcomes:** This scheme consists of three scales to rate the client's knowledge, behaviour and status about a specific problem. For every chosen concept, these scales filled out twice: once for the current situation and once for the desired situation.

With Omaha, every care-related problem can be categorised. The fundamental idea behind Omaha is to work structurally and to justify the defined actions.

1.3 The Omaha System in PUUR.

Omaha is used to systematically define assessments and care plans, which are specified in the main dossier of PUUR. Every caregiver follows a set of required steps, ensuring that Omaha is appropriately applied. Figure 2 shows this workflow.
The dossier consists of five main parts. First, a caregiver fills out a client's personal information in free text fields that can be linked to one (or more) of the Omaha concepts. This allows caregivers to really record client-specific information, which can be used to understand a client's situation and provide personal care. This is the only step that is not compulsory.
Next, the assessment is filled out in five different steps. This assessment follows the Omaha System and will form the basis of the chosen actions in the care plan. First, the caregiver fills out any illnesses and symptoms. After that, the caregiver chooses all relevant concepts and specifies the signs that make the concept visible (e.g. for the concept circulation, irregular heartbeat could be a sign). Each chosen concept is also ranked on status, knowledge, and behaviour. After selecting the concepts, the caregiver has to estimate the time needed for the care. With this last step, the assessment is completed.
Once the assessment is completed, the care plan is defined. For each previously chosen concept, the caregiver has to create actions that will cover the client's needs. These actions are chosen using the intervention scheme. Some actions are already predefined based on the previously selected concepts and symptoms. A caregiver can add a predefined action, but this is not mandatory. Furthermore, there are 38 example care plans available: one for each illness/symptom. These example care plans combine the Omaha System, national guidelines, and expertise from the field and are updated frequently. A caregiver can use these as a starting point and see which actions can be relevant for a certain combination of illness and concept. An action in an example care plan is a combination of a concept, a sort action and a target with some extra explanation of why this action can be relevant. These plans ensure that a caregiver can gain knowledge of the current healthcare standards easily and quickly. After choosing the actions, the caregiver has to fill out why and how the chosen actions will contribute to the well-being of the client to ensure that every action contributes to the overall personal care.
The time planning is defined after the care plan is created. The caregiver plans every action on the timeline to ensure that the desired care is provided. Lastly, the rating scales are evaluated whenever a caregiver deems this necessary. The caregiver keeps track of these ratings in the evaluation section.

1.4 Problem description

The process of creating a care plan is quite complex. The complexity leads to mistakes, especially when the Omaha system is not applied properly. This can lead to incomplete or insufficient care plans. Ecare notices that inexperienced caregivers make (a combination of) the following mistakes:

- They choose concepts in which a client cannot improve (e.g. the concept hearing, while a client is deaf and will never hear again);
- They choose too few actions in a client-specific care plan and, therefore, miss actions that contribute to the recovery of a client;

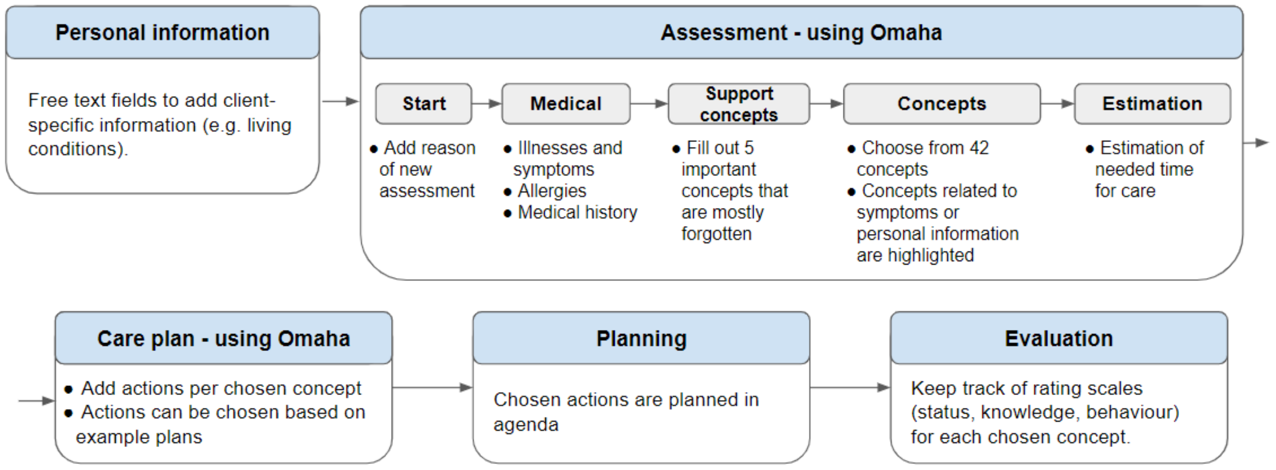

Personal information
Free text fields to add client-specific information (e.g. living conditions).
Assessment - using Omaha
Start
Add reason of new assessment
Medical
Illnesses and symptoms
Allergies
Medical history
Support concepts
Fill out 5 important concepts that are mostly forgotten
Concepts
Choose from 42 concepts
Concepts related to symptoms or personal information are highlighted
Estimation
Estimation of needed time for care
Care plan - using Omaha
Add actions per chosen concept
Actions can be chosen based on example plans
Planning
Chosen actions are planned in agenda
Evaluation
Keep track of rating scales (status, knowledge, behaviour) for each chosen concept.

Figure 2: Workflow of the steps taken in the main dossier based on the Omaha System

- They choose the 'sort action' that does not match the situation of the client (e.g. they choose the sort action 'guidance' for a client that needs help with putting on clothes, while they should choose 'treatments and procedures, because they are helping the client physically);
- They forget to add actions to a care plan that are less common, but would make the care plan client-specific;
- They barely use the example care plans, even though these plans follow national guidelines and cover all the possible actions that correspond to a certain illness.

It is especially unfortunate that the example care plans are hardly utilised, since these example plans contain many actions that are often forgotten or wrongly implemented by caregivers. This reduces the quality of the client-specific care plan, while plans of high quality that tackle the client's problems are essential in healthcare. Lower quality of care plans can lead to insufficient care, which leads to longer recovery times and higher costs. Furthermore, Ecare notices a big difference in the quality of the client-specific care plans between healthcare organisations and caregivers using their platform. Even though every healthcare organisation receives training when they start using the platform, Ecare sees that over time, some caregivers seem to forget how to apply Omaha correctly within PUUR.
Nonetheless, Ecare aims to support caregivers in creating care plans of the highest possible quality, where all suitable actions are implemented in the client-specific care plan. All relevant actions from the example care plans should be implemented to obtain this. One way to achieve this is to use the knowledge of experienced caregivers who implement the suitable actions correctly and make this knowledge available for every caregiver and healthcare organisation that uses PUUR. This can be accomplished by offering suggestions for relevant actions that are often forgotten or wrongly implemented by inexperienced caregivers, based on the knowledge of these experienced caregivers and the example care plans. This will create a moment for the inexperienced caregivers to reflect on the actions they have already chosen and consider other suitable actions that they may have forgotten or wrongly defined. Experienced caregivers, however, do not have time to always assist inexperienced caregivers. Therefore, models can be used to automate this process of providing suggestions. To propose suggestions that are suitable for a client-specific care plan, the EHR of that specific client needs to be used, since the EHR contains the client-specific information. Therefore, in this research, a model will be built to suggest suitable actions based on the client's characteristics. To define which actions are appropriate, previously defined care plans of experienced caregivers from five healthcare organisations are combined with the example care plans. This model uses two types of input, namely knowledge of experienced caregivers to establish connections between the client's characteristics and suitable actions, and EHRs of new clients to propose suitable actions for this client. Since the model needs predetermined data to find connections between a client and appropriate actions, supervised machine learning is applied [5]. This leads to the following research question:

'How accurately can a supervised machine learning model, trained on care plans defined by experienced caregivers, suggest suitable actions (a combination of action type, target and concept) based on example care plans to create a client-specific care plan?'

These suggested actions are selected from the previously mentioned example care plans based on the knowledge of these experienced caregivers. This will guarantee that the proposed actions are valuable, fit a client's symptoms and follow national guidelines. The suggestions should on the one hand increase the quality of the plans and, thus, the quality of healthcare. On the other hand, it should decrease the time to create the care plans, since the suggestions help with drawing up care plans.
The remainder of this article is structured as follows: In Section 2, we review literature on imbalanced data and different supervised machine learning models. In Section 3, we describe our applied approach and explain the data processing and the implementation of the different chosen models. Section 4 shows the results of our approach, which are discussed in Section 5 together with the

drawbacks of this research and possible future research. Finally, we conclude this research in Section 6.

2 LITERATURE

Since our work focuses on creating a model that can suggest multiple possible combinations of actions (i.e. labels) based on client characteristics (i.e. independent features), we restrict this section to multi-label supervised machine learning. Furthermore, we focus on only scalar input data, since this research will be limited to using the numerical and categorical data of the EHRs. First, this section provides literature on imbalanced data, which is used as input, because the number of occurrences of a specific action differs greatly within the data. We do not restrict this literature research to the healthcare domain, because processing data and implementing machine learning models is a generic process that can be applied to different fields of study. Multi-label classification has already been successfully applied in other areas, such as recommendation systems [6, 7, 8], genomics [9] and natural language processing [10, 11].

2.1 Multi-label imbalance

Imbalanced data is a common problem that affects the learning process of any classification model when using a multi-labelled dataset. In our research, certain actions are not as common as other actions, which results in minority and majority classes (i.e. actions that rarely occur versus actions that are frequently added to a care plan). Since minority instances occur less frequently, prediction of a minority class is scarce, and samples belonging to a minority class are misclassified more often than samples belonging to a majority class [12]. With multi-label classification, there are three possible imbalance types [13]:

- **Imbalance within labels:** In this case, a label experiences imbalance on its own. This manifests in many negative samples (= 0) and little positive samples (= 1) within one label.
- **Imbalance between labels:** The frequency of positive instances between the labels is considered for this type. The number of positive instances may be higher in one label than in another label, which causes imbalance between labels.
- **Imbalance among the label-sets:** A combination of labels is often more frequent in a dataset than others. This implies that some label-sets can be considered a majority while others are considered a minority case.

There are four approaches to tackle the problems that arise with multi-label imbalanced data: resampling data, classifier adaptation, ensemble methods and cost-sensitive methods [13].

- **Resampling data:** Resampling data is the most common approach to handle an imbalanced dataset. With this approach, the data is pre-processed to create a more balanced dataset before it is used as input for a model. Resampling consists of either undersampling [14], which removes samples from the majority class, oversampling [15], which creates new samples based on the minority class, or a combination of both. There are two subcategories within resampling: random methods and heuristic methods [13]. Random methods randomly choose samples to delete or produce using undersampling or oversampling, respectively. Because of the randomness of random resampling, there is no consideration of the distribution of classes w.r.t. each other. Random sampling is also more prone to replicating noise, leading to a less robust model. With heuristic methods, heuristic techniques are applied to find the proper samples to delete or produce. These techniques will apply some rule to choose which instance is altered. There are both downsides to heuristic under and oversampling. With undersampling, one is at risk of removing valuable data, especially if there is not much data to begin with. With oversampling, the model is at risk of overfitting, which results in a model that performs very well on the train set, but poor on unseen data. It also does not introduce new data, so it does not tackle the problem of lack of data [16].
  Research of Zeng [17] and Batista [18] show that a combination of over- and undersampling could be the solution. They both use a combination of Synthetic Minority Oversampling Technique (SMOTE) and Tomek-Links for over- and undersampling, respectively. Their research shows that combining these sampling techniques improves the performance of their models. They both first apply SMOTE and then Tomek-Links to resample their data.
- **Classifier adaptation:** With classifier adaptation, the machine learning algorithms are adapted to directly learn the distributions of the imbalanced data and incorporate that in the predictions. One of the proposed classifier adaptations in literature is the use of a min-max modular [19]. They break the classification down into two class sub-problems. For each sub-problem, a prediction can be made using a classification algorithm, after which all classifiers are combined using the minimisation and maximisation principles to generate predictions. Another approach is cross-coupling aggregation (COCOA) [20]. COCOA combines binary and multi-class machine learning algorithms to create predictions for each label.
- **Ensemble methods:** Ensemble methods [21] combine several individual base models to generate a model that improves in terms of generalisation and reduction of overfitting. Common ensemble methods are based on binary relevance. Binary relevance fits a model for every label separately and combines those into one model. It does not consider the combination of labels, but looks at each label on its own. On the one hand, this type of modelling is less prone to overfit, but on the other hand, it does not look at any relationship between labels [21]. Ensemble methods try to overcome the latter. One of those binary relevance ensemble methods is Multi-Label Stacking (MLS) [22] or 2BR, since it applies binary relevance twice. MLS first fits separate models for all labels. After that, it fits a second level of models, taking the outputs of all first-level models as input for the second level. This ensures that the relationship among the labels is taken into account.
  Another ensemble method is the random forest of predictive clustering trees (RF-PCT) [21]. It creates an ensemble with clustering trees as classifier. Due to the random forest, every classifier uses a different set of instances. To make a prediction, the RF-PCT averages the output of all classifiers for each label and applies a threshold to determine whether a label is present.
- **Cost-sensitive methods:** Cost-sensitive methods [23] use a cost metric to link a certain cost to a misclassified sample to

minimise the total cost. To tackle the imbalanced learning problem, higher costs are associated with the minority classes. This approach is not typical for multi-label classification, due to the fact that defining a cost matrix is very difficult and, most of the time, the appropriate misclassification cost is unknown [24].

2.2 Multi-label classification models

When tackling a multi-label classification problem, there are two different types of strategies, namely methods that process the labels one by one (problem transformation methods) and methods that handle multi-label data directly (algorithm adaptation methods) [25]. One of the most popular and intuitive problem transformation methods is binary relevance [26], where the multi-label problem is divided into binary learning tasks, one for each label. Binary relevance can be applied with different modelling techniques, such as decision trees/random forest or support vector machines, and using binary relevance is computationally cheap. However, the downside of this approach is that it cannot take label dependency into account.

An algorithm that considers label-dependency is the widely used algorithm adaptation method Multi-label KNN (ML-KNN) [25, 27]. ML-KNN is the famous KNN algorithm adapted for multi-label problems. ML-KNN finds the nearest neighbours of the instance and takes the instances into account that at least have a positive value for a specific label. Furthermore, ML-KNN can create a ranking of the labels as output. According to Zhang [27], ML-KNN outperforms a set of well-known multi-label algorithms, such as BoosTexter and rank-SVM. ML-KNN considers label dependency by operating in a multi-label space, and the neighbours it uses for a given instance with unknown labels will inherently capture some aspects of correlation between labels (i.e. if two instances are similar in their feature space, they will probably also have similarities in their labels).

Another flexible but easy model that can be used for multi-label classification is a multilayer perceptron (MLP) [28]. An MLP is a neural network that consists of an input layer, one or multiple hidden layers, and an output layer. The output layer consists of the same amount of neurons as there are labels to predict, to be able to predict each label simultaneously. MLPs can learn complex non-linear relationships within data, which makes them suited to handle high-dimensional input data and capture patterns in a multi-label dataset. According to the work of Zhang [29] and Rajput [30], an MLP and an ML-KNN have very similar performance. However, due to the structure of both MLPs and ML-KNNs, they do not have an explicit mechanism to capture label dependencies, but rather an implicit one. This can result in models that do not entirely capture the relationship between labels, unlike approaches that do take label dependency explicitly into account. One of these approaches to create explicit label dependency is to apply Label-Power set [31]. With this approach, each set of present labels is considered a single label. This approach also has one major drawback, namely that the newly created dataset experiences sparseness; the data is likely to have a large number of optional labels, but only has a few instances per label. Other types of models that also exploit label dependency are label correlation models [32] or multi-dimensional Bayesian network classifiers (MBCs) [33]. Unfortunately, the main drawback of the latter is the high computational cost while finding the most optimal network structure.

Computational cost is a challenge in general when working with machine learning models. Law et al. [34] states that more complex models are computationally expensive, whereas extremely simple models may not be able to classify as desired. Thus, while creating a classifier, efficiency and simplicity should be considered and handled simultaneously. Al-Jarrah et al. [35] states that while ensemble methods (i.e. methods that combine multiple models into one) improve performance, model complexity and computational cost will grow exponentially if large-scale data is used. Since we are working with a lot of client information, we have to consider the balance of performance and computational cost when creating a model that generates suggestions for care plans.

2.3 Contribution

Based on the literature, we will use resampling methods to resample our data. With this method, we are flexible in choosing and comparing different models in comparison with classifier adaptation and ensemble methods where the resampling is incorporated into the model itself. Furthermore, we choose resampling over cost-sensitive methods, since we cannot define the cost matrix appropriately with our data. Because a combination of SMOTE and Tomek-Links seems successful [18, 17], we combine these two strategies to resample our dataset.

To see the effect of label dependency and computational cost on the performance of the models, we compare different models. We apply Binary relevance in combination with SVM and random forest, ML-KNN, and we build an MLP model. We will not look into label powerset, due to the fact that our data is already sparse, and this approach will only create an even sparser dataset. All the chosen models are compared on their performance in terms of different metrics (Section 3.5 shows which performance metrics are used).

Addressing the research question and the related work, the contributions of this research will be as follows:

- We will demonstrate how to process EHR data that consists of scalar (structured) data;
- We will show and provide insights if relatively computationally inexpensive supervised machine learning algorithms can be applied to give suggestions for actions;
- Our study is based on unique data, namely EHRs of five Dutch home care organisations;
- We will demonstrate how care plans created using supervised machine learning compare to care plans created by experienced caregivers.

3 METHOD

In this section, we explain the method applied to design the supervised machine learning models. Figure 3 provides a short workflow of the steps taken to create the models which is based on the workflow presented in [36]. Creating a multi-label supervised machine learning model starts with gathering labelled data, in our case the EHRs of clients. After that, the data needs to be processed to create datasets that can be used as input for the models. Next, the data is split into a train, validation and test group, after which the train and validation data is resampled to create more even distributions over

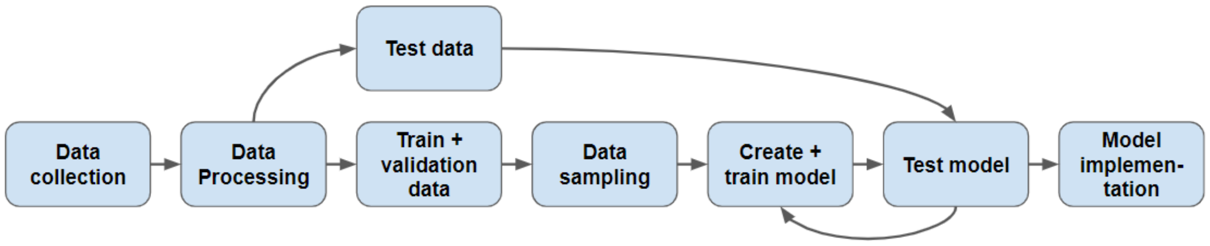


Figure 3: General workflow of the applied methodology based on [36]

the labels. This data is used as input to create and train the model. Subsequently, the test data is used to determine the performance of the models. Lastly, the care plans of the model with the highest performance will be compared to care plans of an experienced caregiver.

3.1 Available data

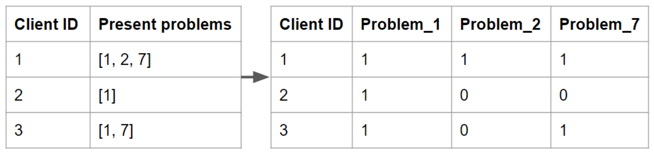
Ecare has provided data from clients of five different healthcare organisations that all use PUUR and the Omaha classification system. These healthcare organisations apply the system correctly in general, but to filter out any exceptions, only clients with at least five actions in their care plan are considered. This number is based on expert knowledge provided by Ecare. Furthermore, Omaha has been used since 2017; thus, only patients registered in 2017 or later are included. Since the example care plans are used as a basis and cover only one illness at a time, three models are created for the most common illnesses among the clients, namely dementia (10,702 clients), heart failure (6,286 clients), and diabetes (5,511 clients). This will ensure that only relevant actions are suggested for a specific illness. We have chosen these three, since we want to see if a model is generalisable for the most common illnesses and does not only work for one specific disease. Furthermore, these three illnesses have the most available data and are the most relevant for the clients. Based on the example care plans, heart failure has 44 possible actions, dementia has 65, and diabetes has 49 possible actions. These actions present the number of labels, where each action is either present or not (binary variable).
Since we work with three different diseases and create three separate models, three different data sets have been made, one for each disease. In these datasets, only the clients with said disease are included. All three datasets consist of the independent features presented in Table 1. We have not taken the symptoms of each problem into account, because it creates highly sparse datasets, which is undesirable due to the fact that sparse data is more likely to overfit [37].

3.2 Data processing

To be able to design a model with the given data, the data needs to be processed. The steps are discussed in the following subsections.

3.2.1 *Change old codes to new codes.*

The first step of the data processing consists of creating consistency in the numerical codes used for the features. All diseases, problems, targets, and sort actions have certain numerical encodings. Over time, these encodings have changed, and it is essential to ensure that the same code has the same meaning for every client. Therefore, all the old codes are adjusted to the new encodings.



Client ID	Present problems
1	[1, 2, 7]
2	[1]
3	[1, 7]

Client ID	Problem_1	Problem_2	Problem_7
1	1	1	1
2	1	0	0
3	1	0	1

Figure 4: Example of transformation from vector feature to binary features

3.2.2 *Filter actions on actions example care plans.*

In this step, we have only kept the actions that are part of the example care plans for the three diseases. So for heart failure, we have filtered out the 44 relevant labels, for dementia 65 labels and for diabetes 49 labels. This ensures that only relevant actions are suggested for one of the specific diseases. These labels are the dependent features, i.e. the features that will be predicted.

3.2.3 *Delete duplicates.*

The next step of the data processing is to remove any duplicate rows, if present. If there is an exact duplicate, an evaluation of a client of the exact same moment ended up twice in the database. Duplicates will bias the model, which results in the model learning extra patterns for the duplicate instances. No duplicate rows are present in our three datasets, so we have not removed any entries.

3.2.4 *Feature vectors to separate features.*

To ensure that the features currently displayed as a vector can be used as input for the models, we have split all vectors into separate binary features. Figure 4 displays how these vectors are split into binary features. This transformation is applied to the features 'Present illnesses', 'Present problems', 'Current status', and 'Target status'. We have also transformed the dependent features to separate binary labels to create multi-label datasets.

3.2.5 *Handling missing/unknown values.*

Features with more than 10% missing data will be removed from the dataset. If a specific client misses more than 50% of the features, this client will also be removed. The remaining missing values will be replaced based on the distribution of the present data for categorical data and the mean of the feature for numerical data. In all three datasets, no client misses more than 50% of the features, so no clients are removed. Furthermore, the following adjustments are made in the three datasets:

Table 1: Description of independent features present in heart failure (n= 6,286), dementia (n= 10,702) and diabetes (n= 5,511)

Feature	Description	Datatype	Values
Age	Age of client in years	Numerical	Range heart failure: [-23;141] Range dementia: [-76; 141] Range diabetes: [1; 123]
Sex	Gender of client	Categorical	1: man 2: woman 3: unknown
Civil status	Client's relationship status	Categorical	0: unknown 1: not married 2: married 3: divorced 4: widower 5: registered partner 6: divorced 7: separated from registered partner 9: not married and no registered partner NaN: not filled out
Living condition	Type of living arrangement of client	Categorical	0: unknown 1: lives alone and independently 2: lives as child together with parent 3: lives with partner 4: lives with partner and children 5: lives with adult and children 6: other multi-person household 7: lives in care institution with residence 9: unknown NaN: not filled out
Present illnesses	37 possible diseases that a client can have next to the main disease	Binary vector	Length of vector: 39 possible illnesses that are either present or not (binary variable)
Present problems	41 possible problems that a client can have	Binary vector	Length of vector: 41 possible problems that are either present or not (binary variable)
Current status	Current status that accompanies present problem	Vector	Length of vector: 41 values for current status that are filled out for each present problem Range: [1;5] NaN: not present
Target status	Desired status that accompanies present problem	Vector	Length of vector: 41 values for target status that are filled out for each present problem Range: [1;5] 0: not filled out NaN: not present

- **Sex:** For the heart failure dataset, there are five people with unknown sex. These are replaced based on the existing distribution in this dataset, namely a probability of 0.42 for men and 0.58 for women. The dementia dataset has seven people with unknown sex, which are replaced based on the following probabilities: 0.39 for men and 0.61 for women. The diabetes dataset has seven people with unknown sex, which are replaced based on the following distribution: a probability of 0.46 for men and 0.54 for women.
- **Civil status:** The set threshold of 10% is not met for any dataset, so this feature is removed from all three sets.
- **Living condition:** There are 5404 clients with NaN, two clients with category 0 and 178 clients with category 9 in the heart failure dataset. These are replaced based on the known probabilities: 0.57 for category 1, 0.38 for category 3, 0.01 for category 4, 0.02 for category 5, and 0.02 for category 6.
  In the dementia dataset, there are 7136 clients with NaN, 37 with category 0, and 148 with category 9. These are replaced with the following probabilities: 0.51 for category 1, 0.44 for category 3, 0.01 for category 4, 0.01 for category 5, 0.02 for category 6, and 0.01 for category 7.
  For diabetes, there are 3825 clients with NaN, five with category 0, and 163 with category 9. These are replaced with the following probabilities: 0.52 for category 1, 0.01 for category 2, 0.4 for category 3, 0.02 for category 4, 0.02 for category 5, 0.02 for category 6, and 0.01 for category 7.

- **Present illnesses:** For each illness it is determined if it reaches the set threshold of 10%. This results in 19 illnesses that are removed from the heart failure dataset, 26 diseases removed from the dementia dataset, and 19 diseases removed from the diabetes dataset.
- **Present problems:** For every problem it is determined if the threshold of 10% is exceeded. This results in 17 problems that are removed from the heart failure dataset, 20 problems removed from the dementia dataset, and 19 problems removed from the diabetes dataset.
- **Current status:** The accompanying current status is removed for every present problem that is removed.
- **Target status:** For each present problem that is removed, the accompanying target status is also removed. Furthermore, for all clients with a particular problem, a current status must be filled out (see Section 1.3 for the explanation). The target status, however, is optional. If the target status is not filled out for a client, it is assumed that it is the same as the current status. The target status and current status are further processed during feature engineering.

3.2.6 *Range of numerical features.*

The only numerical feature is the age. Based on the type of clients that need care and the oldest registered age in the Netherlands (115 years), we have changed the age of the clients younger than 50 and older than 110. The age of these clients is replaced with the mean value, which is 90 years for clients with heart failure, 89 years for clients with dementia, and 87 years for clients with diabetes.

3.2.7 *Feature engineering.*

Four new features are created based on the available data:

- **Number problems illness:** This feature is a numerical feature that shows the number of problems a client has that matches the main illness;
- **Number problems not illness:** This feature is a numerical feature that shows the number of problems a client has that do not match the main illness;
- **Number other illnesses:** This feature is a numerical feature that shows the total number of other illnesses a client has;
- **Difference status:** This feature replaces the current status and target status to make the datasets less sparse. It shows the difference between the current status and the target status, i.e. the desired status progress of a client. Next, this feature is binned into three categories. This is needed because zero is a possible value, but does not represent the value 'zero', but the value 'not filled out'. To avoid a correlation between zero and the other values, the values are categorised into three categories: negative (values below -1), stable (values between and including -1 and 1), and positive (values higher than 1).

3.2.8 *Remove outliers.*

We have removed outliers, since it increases model performance. Outliers can only exist in numerical features and are defined as an instance bigger or smaller than the mean ± 3 times the standard deviation [36].

3.2.9 *Feature scaling.*

To make the optimisation process faster while training the model, we have scaled the numerical features. The scaled features are 'Age', 'Number problems illness', 'Number problems not illness' and 'Number other illnesses'. We have normalised the data using the following standard equation, where x displays a feature:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

3.2.10 *One-hot encoding.*

We have applied one-hot encoding to transform the categorical features with more than two categories into binary variables. These features are 'Living condition', 'Present problems' and 'Present illnesses'. This ensures that all categories within a feature have equal importance and that the machine learning algorithms do not misinterpret the categories for integers with a meaning.

3.2.11 *Remove multicollinear parameters.*

The variance inflation factor (VIF) is an indicator of multicollinearity [37]. The lowest possible value for the VIF is 1, and a value of 10 or higher indicates a lot of collinearity between features. When there is high correlation between independent variables, multicollinearity occurs. This phenomenon is undesirable, since it makes the statistical inferences less reliable. If the VIF for one or multiple features is higher than 10, we have removed the feature with the highest VIF. We have repeated this process until all features have a VIF lower than 10.
In general, the present problems have a very high VIF with their matching difference in status. This makes sense, since a difference in status is only present if the problem itself is present as well. Therefore, we have decided to split each dataset into two new datasets: one dataset with the binary features that represent whether a problem is present and one dataset with the binary features that represent the desired difference between the current and target situation. This leads to six new datasets in total.
Furthermore, we have removed one of the 'Sex' features that was created during one-hot encoding, one of the 'Living condition' features created after one-hot encoding, 'Number of problems illness' and 'Number of diseases' for the heart failure datasets. We have removed one of the 'Sex' features created during one-hot encoding, one of the 'Living condition' features created after one-hot encoding and 'Number of problems illness' for the dementia and diabetes datasets.

After processing the data, we have obtained the following datasets:

- **Heart failure problem:** 6,286 clients, 51 independent features (Age, Sex, 5 features for Living condition, nr problems not illness, 18 features for present illnesses, 25 features for present problems) and 44 binary labels;
- **Heart failure difference:** 6,286 clients, 99 independent features (Age, Sex, 6 features for Living condition, nr problems not illness, 18 features for present illnesses, 72 features for difference present problems) and 44 binary labels;
- **Dementia problem:** 10,702 clients, 43 independent features (Age, Sex, 6 features for Living condition, nr problems not illness, nr other diseases, 11 features for present illnesses, 22 features for present problems) and 65 binary labels;
- **Dementia difference:** 10,702 clients, 87 independent features (Age, Sex, 6 features for Living condition, nr problems not

illness, nr other diseases, 11 features for present illnesses, 66 features for difference present problems) and 65 binary labels;

- **Diabetes problem:** 5,511 clients, 50 independent features (Age, Sex, 6 features for Living condition, nr problems not illness, 18 features for present illnesses, 23 features for present problems) and 49 binary labels;
- **Diabetes difference:** 5,511 clients, 93 independent features (Age, Sex, 6 features for Living condition, nr problems not illness, 18 features for present illnesses, 66 features for difference present problems) and 49 binary labels.

3.3 Sampling data

As stated in Section 2.1, imbalanced data is a common problem that affects the learning process of any classification model when using a multi-labelled dataset. To tackle this problem, we have split the datasets into a train and test group and only apply the sampling strategy on the train dataset so that the models can be tested with the actual data and not the sampled data. The datasets are split with a ratio of 75/25 (train, test resp.). We have counted each disease's number of appearances per label in the train group, as shown in Figure 5. The figure shows that there is a strong imbalance between the labels regarding appearance (the label counts are the same for the datasets with the problem and the difference).

We have applied SMOTE and Tomek-Links to resample the data as explained in Section 2. Since we are working with a multi-label dataset, SMOTE and Tomek-Links are extended to their multi-label variant, namely Multi-label SMOTE (MLSMOTE) and Multi-Label Tomek Links (MLTL). Below, the two strategies are elaborated on.

3.3.1 *Multi-Label Synthetic Minority Over-sampling Technique.*

For heuristic oversampling, we have applied MLSMOTE [38]. This method uses interpolation of instances that belong to the nearest neighbours to create new instances. The corresponding labels are also based on the same nearest neighbours. By interpolating, duplicates are mostly avoided. MLSMOTE uses two metrics to determine the imbalanced labels: the imbalance ratio per label (IRLbl) and the mean imbalance ratio (MeanIR). The IRLbl is calculated as follows:

$$\text{IRLbl}(y) = \frac{(\underset{y' \epsilon L}{max} \sum_{i=1}^{|D|} h(y', Y_i))}{\sum_{i=1}^{|D|} h(y, Y_i)} \quad (2)$$

With D = multi-label dataset, Y = set of labels, Y_i = ith label, and y = the specific label. The larger the IRLbl, the higher the imbalance for the examined label. The MeanIR is calculated as follows:

$$\text{MeanIR} = \frac{1}{|L|} \sum_{y=Y_1}^{Y_{|Y|}} \text{IRLbl}(y) \quad (3)$$

L shows the disjoint set of the labels. The MeanIR shows the average imbalance in a multi-label dataset. These metrics are both used to identify imbalanced labels, identify majority and minority labels, and determine the average level of imbalance. This is done with the following steps:

(1) Determine the minorities using the IRLbl and the MeanIR. An instance is a minority if the IRLbl(l) < MeanIR. This can be interpreted as follows: the frequency of the label is below the average, and is thus a minority.
(2) Choose a minority and find the nearest neighbours.
(3) Based on the neighbours, generate new features. For numerical features, interpolation is used. For categorical features, the category that appears most in the neighbours is chosen. If multiple categories appear the most, a category is randomly selected from these categories.
(4) Generate a new label set. For this, nearest neighbours are used. If a label is present in half or more of the neighbours, it is selected for the new label set.

3.3.2 *Multi-Label Tomek Links.*

For heuristic undersampling, we have applied the MLTL approach [39]. MLTL preserves the relationship between samples across multiple labels, which is important for multi-label classification. MLTL removes data points in four steps:

(1) Identify the nearest neighbours of each instance. The distance or similarity is determined for each instance w.r.t. the other instances.
(2) Determine majority-minority pairs where one instance belongs to the majority and another to the minority across all labels using the IRLbl and the MeanIR.
(3) For each majority-minority pair, check if the instances are neighbours by setting a threshold for the distance.
(4) Remove the majority instance to improve the data balance for each majority-minority pair.

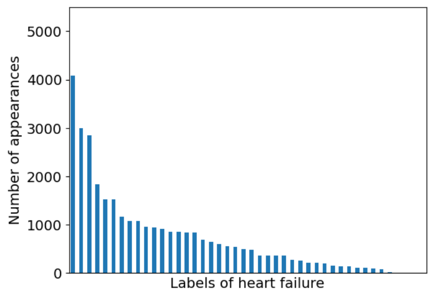
We have first applied MLSMOTE and, afterwards, MLTL, as proposed in Section 2.1. We have created 1000 oversamples for the two heart failure datasets, 1200 oversamples for the dementia datasets and 800 oversamples for the diabetes datasets based on trial and error. After applying MLTL as well, this results in the distributions visible in Figure 6, where the labels are sorted on the frequency. The number of appearances is similar for the datasets with the problems and the differences. The labels do not have the exact same frequency after resampling, due to their dependency/correlation between them. The labels with a higher number of appearances correlate with a higher number of other labels than those with a lower number of appearances. Resampling happens on a combined level and not for each label independently.

3.4 Supervised machine learning algorithms

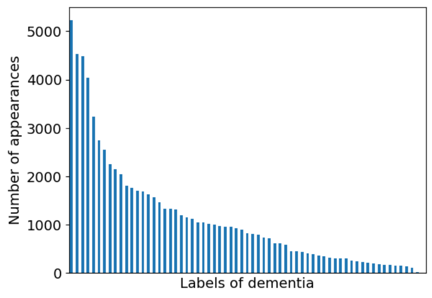
A prediction of relevant actions for a certain client is made using four different models, namely ML-KNN, an MLP, Binary relevance with SVM and Binary relevance with Random Forest, as discussed in Section 2.3. In this case, an action is a combination of a problem, target and sort action, as mentioned earlier. The models have been built using Python. During training, the train set is divided into a train and validation set using k-fold cross-validation with k=5. Cross-validation is applied to mitigate overfitting by averaging the performance over the five folds [37]. The hyper-parameters of each model are all fine-tuned to find the 'optimal' settings for each model, i.e. the settings that result in the highest combination of precision, recall and F1-score.

3.4.1 *Multi-label K-Nearest Neighbours (ML-KNN).*

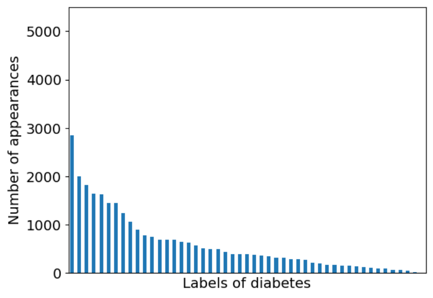
The first model that we have fitted is the ML-KNN. One hyperparameter needs to be tuned, namely the number of neighbours (K) that is considered when a new instance is predicted. We have used



(a) Heart failure

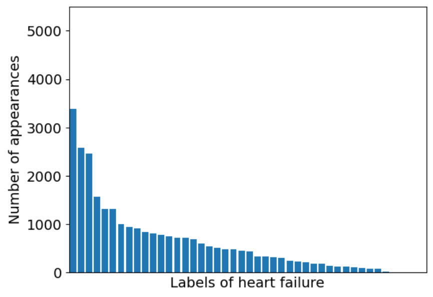


(b) Dementia

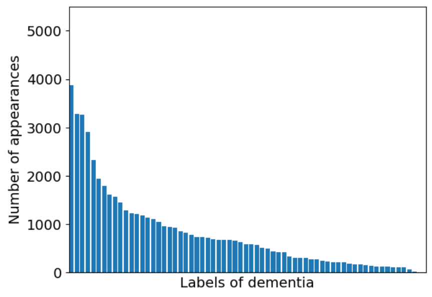


(c) Diabetes

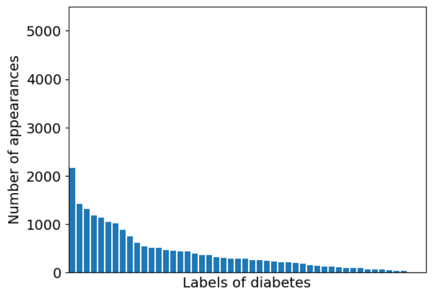
Figure 5: Number of appearances of each label for each illness, sorted on frequency



(a) Heart failure



(b) Dementia



(c) Diabetes

Figure 6: Number of appearances of each label for each dataset after resampling, sorted on frequency

Table 2: Optimal number of neighbours for ML-KNN

Dataset	**Number of neighbours**
Heart failure problem	4
Heart failure difference	5
Dementia problem	4
Dementia difference	6
Diabetes problem	5
Diabetes difference	5

a range of 2 to 30 with an increment of 1 to determine the most suitable number of neighbours. Each dataset's optimal number of neighbours lies between four and six (Table 2).

3.4.2 Multi layer perceptron (MLP).

Since the problem at hand is a multi-label classification function, we have used the ReLU activation function for the hidden layers, the sigmoid activation function for the last layer, and the binary cross entropy function as loss function [40, 41]. The number of input neurons is the number of independent features, and the number of output neurons is the number of dependent features, i.e. the number of labels. S. Karsoliya [42] states that one or two hidden layers are adequate to solve a non-linear complex problem. A third layer can be added to increase the overall accuracy, but it also increases complexity. Therefore, we have fitted 1-3 hidden layers. For the number of neurons in each layer, we have used different settings based on the following rule of thumb: '*The number of hidden neurons should be between the size of the input layer and the size of the output layer*' [43]. This leads to the following ranges: heart failure problem: 40-50 with an interval of 5, heart failure difference: 40-100 with an interval of 10, dementia problem: 40-70 with an interval of 10, dementia difference: 60-90 with an interval of 10, diabetes problem: 40-50 with an interval of 5, and diabetes difference: 40-100 with an interval of 10. Table 3 displays the optimal settings for each dataset.

3.4.3 Binary relevance with Random Forest (BR-RF).

Random forest applies multiple decision trees, which are combined and averaged in the model. The combination of decision trees reduces the variance compared to using an individual decision tree [37]. Therefore, random forest is preferred over decision trees. For each label, we perform a grid search with the following settings:

- **Number of estimators:** The number of trees in the forest (200-2000 with an interval of 200);
- **Max features:** the number of features considered when looking for a split (log2, sqrt);
- **Max depth:** The maximum depth of the tree (10-80 with an interval of 10);
- **Min samples split:** The minimum number of samples required to split a node (2, 5, 10);
- **Min samples leaf:** the minimum number of samples required to be at a leaf node (1, 2, 5, 10);

Table 3: Optimal settings for MLP

Dataset	**# neurons first layer**	**Second hidden layer**	**# neurons second layer**	**Third hidden layer**	**# neurons third layer**
Heart failure problem	50	True	45	True	45
Heart failure difference	100	True	80	False	-
Dementia problem	70	True	70	True	70
Dementia difference	80	True	90	True	80
Diabetes problem	50	True	50	True	45
Diabetes difference	90	True	90	False	-

- **Bootstrap:** If bootstrap samples are used (always True).

One of the possible combinations is chosen for each label, depending on the label and the dataset. This results in 44 separate models for the heart failure datasets, 65 models for the dementia datasets, and 49 models for the diabetes datasets.

3.4.4 Binary relevance with SVM (BR-SVM).
Support vector machines apply kernels to try and make the data linearly separable. It is one of the more robust models for supervised learning, but data is rarely completely linearly separable, so it is not used often [37]. Therefore, BR-SVM will have a different performance for each label, which will influence the overall performance of the model. For each label, the following parameters are optimised using grid search:

- **Estimator C:** This is the regularisation parameter, which is a hyperparameter that controls the trade-off between maximising the margin and minimising the classification error. For a high C, the margin is 'hard', and fewer data points are allowed to lay in the 'wrong' area (0.1, 1, 10);
- **Estimator gamma:** This presents the kernel coefficient (0.001, 0.01, 0.1, 1);
- **Estimator kernel:** Specifies the used kernel type (RBF, linear, sigmoid).

One of the possible combinations is chosen for each label, depending on the label and the dataset. This results in 44 separate models for the heart failure datasets, 65 models for the dementia datasets, and 49 models for the diabetes datasets.

3.5 Performance measurements

To compare the performance of the different models and to see if the models capture the knowledge of experienced caregivers, the weighted precision, recall and F1-score are compared. We also look at the hamming loss [44], which is a performance measurement specifically for multi-label categorisation problems and looks at the number of wrongly predicted labels over the complete prediction set as follows:

$$\frac{1}{\text{nrSamples} \cdot \text{nrLabels}} \cdot \sum (\text{predicted} \neq \text{true}) \tag{4}$$

We want to achieve a high F1-score, precision and recall, but a hamming loss that is as small as possible.

3.6 Testing in real-life environment

Based on the performance of the models, the model with the best performance is applied in a real-life environment. Best performance is here defined as the model with the highest F1-score, since this metric is a balance between precision and recall. A comparison is made between a care plan created using the best-performing model and a care plan created by an experienced caregiver to see the performance of the models with respect to experienced caregivers. This is done as follows:

(1) eight caregivers of two healthcare organisations (five caregivers from the first organisation and three caregivers from the second) receive profiles of three clients, one with heart failure, one with dementia and one with diabetes;
(2) For each client, the caregivers are presented with two care plans: one drawn up by an experienced caregiver and one created by the model. The caregivers do not know which care plan is which;
(3) The caregivers decide which care plan is most suitable for the client at hand. They also indicate if they would add/remove any actions from the most suitable care plan;
(4) They can add any additional notes if needed.

To ensure that the two care plans can be compared, we have removed any additional notes of the experienced caregiver concerning the chosen actions, since this would indicate which care plan was drawn up by the experienced caregiver.
Due to privacy, each organisation only receives clients from their own database. So, in total, six clients are used (three clients of the first organisation and three of the second organisation).

4 EXPERIMENTAL RESULTS

This section is divided into two parts: the performance of the models and the results of the best-performing model applied in a real-life environment.

4.1 Performance models

The four performance measurements have been determined after training the four different models on all six datasets.
Figure 7 shows the F1-score, precision and recall for both heart failure datasets, where the problem dataset is visualised in blue and the difference dataset in red. The figure shows that the BR-RF model with the problem dataset has the highest F1-score (0.94 train, 0.78 test), precision (0.95 train, 0.89 test) and recall (0.56 train, 0.53 test). The figure also shows that the ML-KNN with the difference dataset has the lowest performance with an F1-score of 0.59 and 0.43 (for train and test, resp.), a precision of 0.72 and 0.48 (for train and test, resp.), and a recall of 0.56 and 0.43 (for train and test,

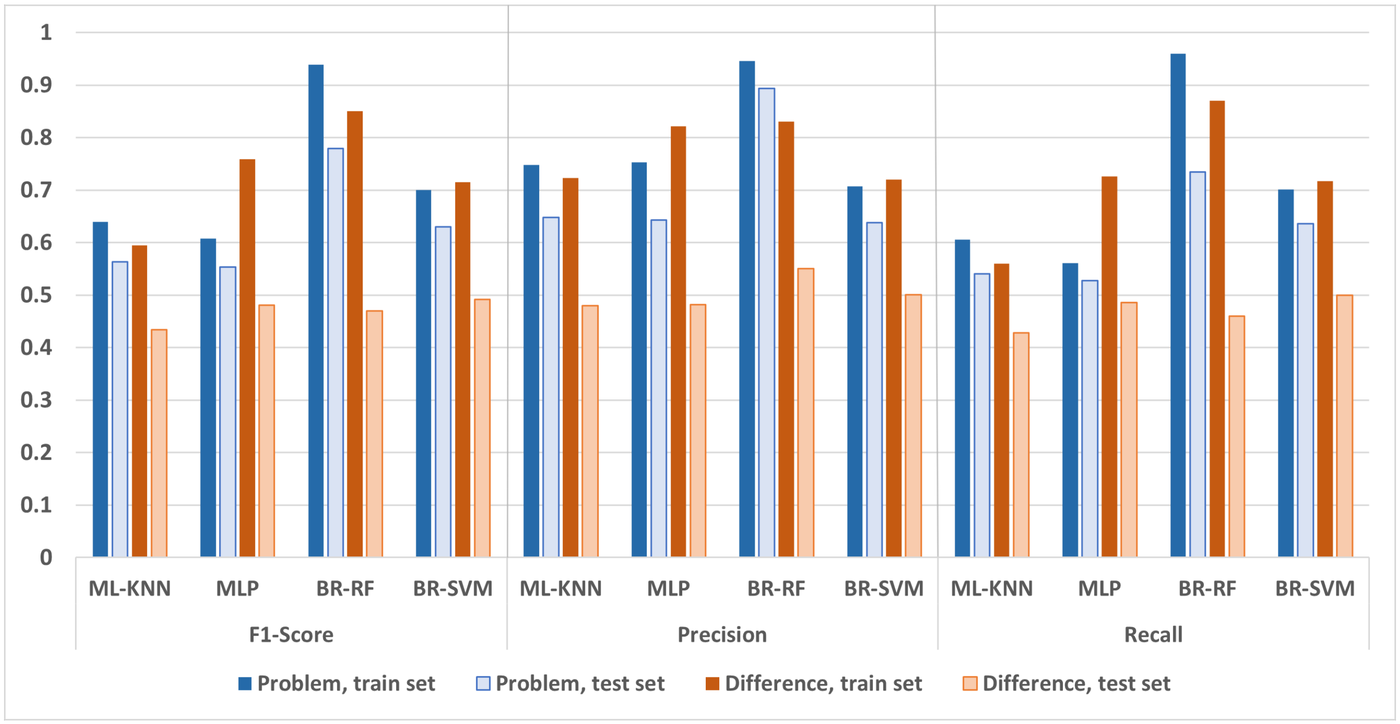


Figure 7: Heart failure - Performance measurements of the train/validation and test split for both datasets

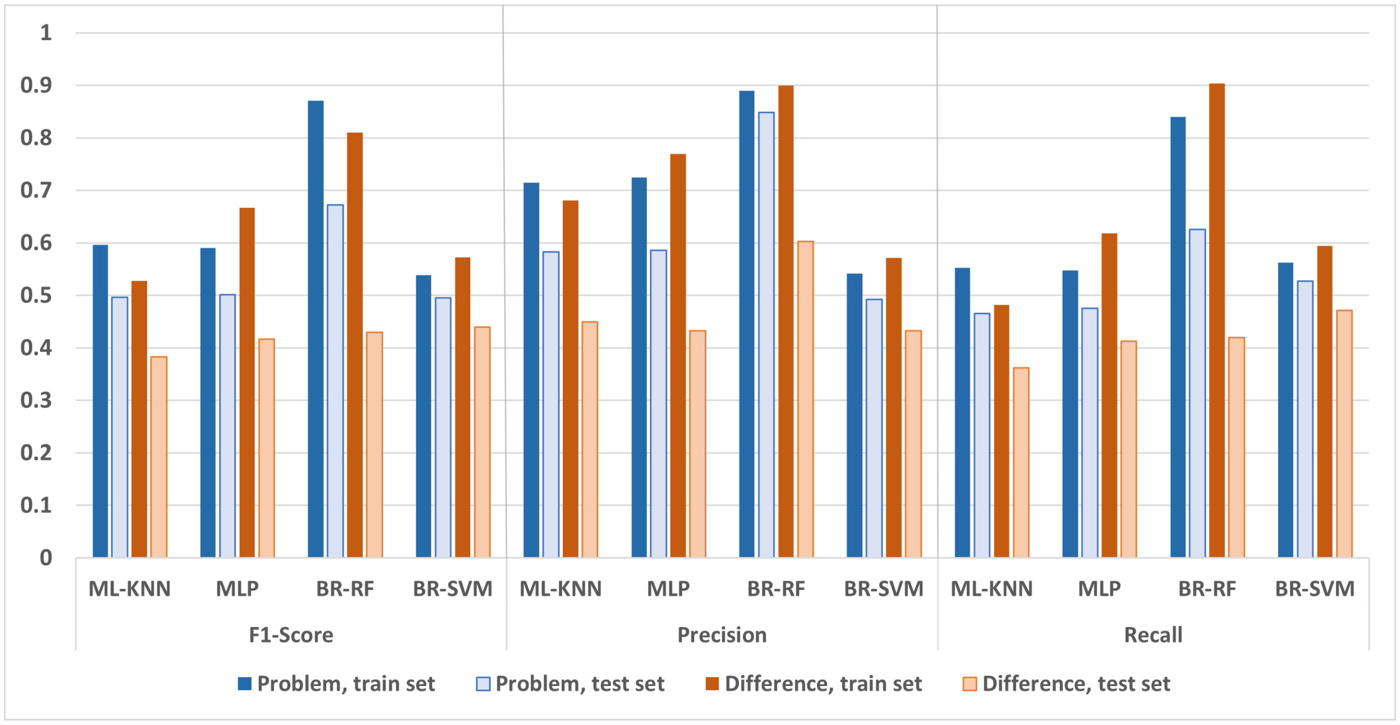


Figure 8: Dementia - Performance measurements of the train/validation and test split for both datasets

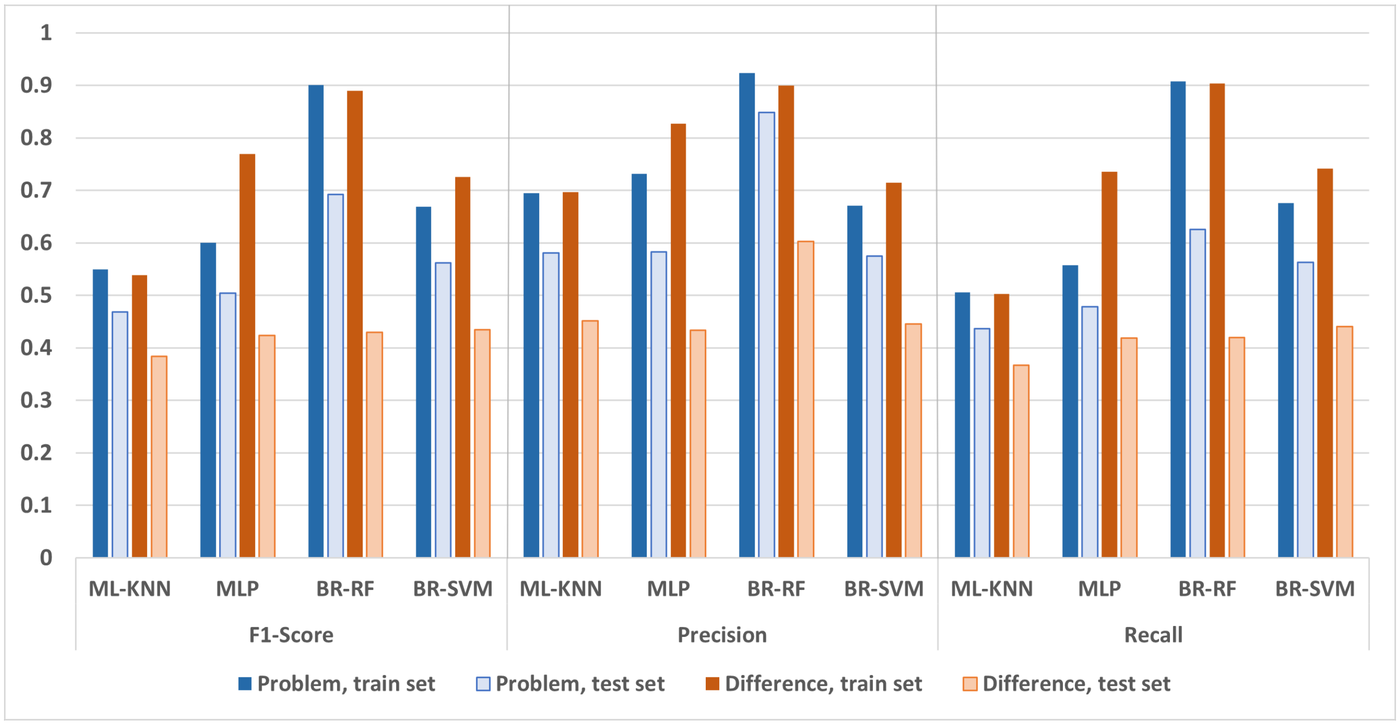


Figure 9: Diabetes - Performance measurements of the train/validation and test split for both datasets

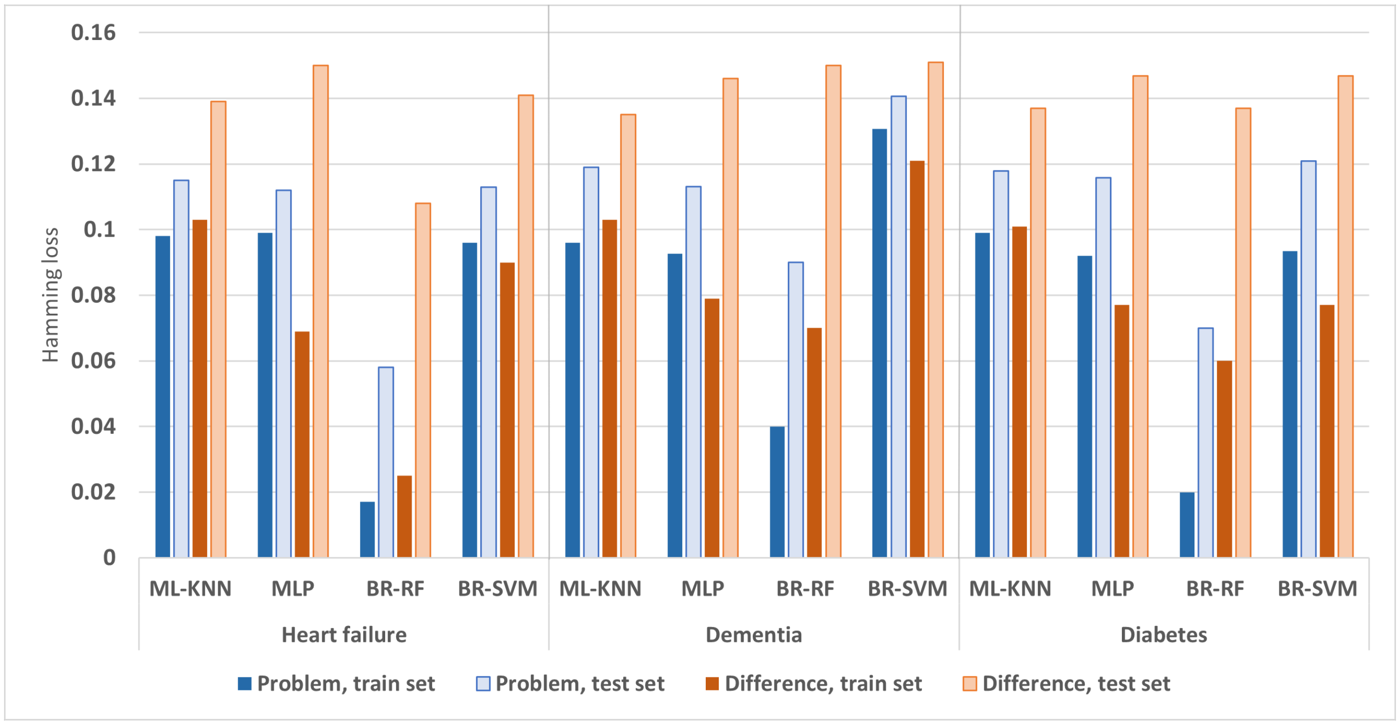


Figure 10: Hamming loss for all three illnesses

resp.). Furthermore, it shows that the test set's performance with the difference datasets is lower for each model compared to the problem dataset. Lastly, the figure shows that for heart failure, the models that use binary relevance as a principle (BR-RF and BR-SVM) have a higher F1-score than those that predict all labels at once (ML-KNN and MLP). This is due to the fact that the recall for the BR models is higher, which indicates that these models predict fewer false negatives.
For both dementia datasets, Figure 8 shows the F1-score, precision and recall, where once again blue visualises the problem datasets and red the difference datasets. For dementia, we see similar trends comparable to the heart failure models. The BR-RF model with the problem dataset is the model with the highest F1-score (0.87 - train, 0.67 - test), precision (0.89 - train, 0.84 - test) and recall (0.84 - train, 0.62 - test). The figure displays that the test set for the problem datasets consistently outperforms the test set for the difference datasets. The least performing model for dementia is the ML-KNN with the difference dataset with an F1-score of 0.53 and 0.38, a precision of 0.68 and 0.45 and a recall of 0.48 and 0.36 (all train and test, resp.).
Figure 9 displays the F1-score, precision and recall for both diabetes datasets, again in blue the problem datasets and in red the difference datasets. For diabetes, there are also similar outcomes compared to heart failure and dementia. The best-performing model for diabetes is the BR-RF with problem dataset with an F1-score of 0.90 and 0.69, a precision of 0.92 and 0.84, and a recall of 0.91 and 0.63 (all train and test, resp.) Again, the test results of the difference datasets are lower than those of the problem datasets. The worst performing model for diabetes is again the ML-KNN in combination with the difference dataset with an F1-score of 0.54 and 0.38, a precision of 0.69 and 0.45 and a recall of 0.50 and 0.37 (all train and test, resp.).
Lastly, we have looked at the hamming loss, visible in Figure 10. The figure shows that the hamming loss for the test set is lowest for the models that use the problem dataset as input compared to those that use the difference dataset as input. This can be linked to the higher performance measurements of the problem datasets, since a higher precision and recall leads to a lower hamming loss. All values for the train set of the problem datasets are below 0.1, except for the BR-SVM model for dementia (0.13). For the test set of the same dataset, all values are between 0.11-0.14, except for the BR-RF models with values of 0.058, 0.09 and 0.07 (heart failure, dementia and diabetes, resp.). Looking at the difference datasets, we see a similar range for the train group with every value below 0.12. With regards to the test group, however, the values are between 0.11-0.16.

4.2 Real-life environment

Based on the performance of the models, we have chosen BR-RF as a model for all three illnesses. All three models use the 'problem' dataset as input, since these show the highest F1-score and the lowest hamming loss for the BR-RF model. This model type is used to create six care plans, two for each illness as explained in Section 3.6. Table 4 shows the number of actions present in the care plans for every used client, one created by an experienced caregiver and the other by the model. The last column shows the number of identical actions present in both care plans. There is at most 25% overlap between the model's and caregiver's care plan.

Table 4: Number of actions in each care plan and number of identical actions present in both care plans.

Illness (# organisation)	Experienced caregiver	Model	# identical actions
Heart failure (1)	5	8	0
Heart failure (2)	7	8	3
Dementia (1)	10	16	4
Dementia (2)	4	12	2
Diabetes (1)	5	5	0
Diabetes (2)	7	7	1

Table 5: Number of caregivers that have chosen each client-specific care plan

Illness (# organisation)	Experienced caregiver	Model
Heart failure (1)[1]	1	4
Heart failure (2)	2	1
Dementia (1)	3	1
Dementia (2)	3	0
Diabetes (1)	4	0
Diabetes (2)	3	0

Table 5 displays the number of caregivers choosing each client's preferred care plan. The number behind the illness shows the number of the organisation, so a distinction is made between the organisations' outcomes. The table shows that for heart failure five caregivers prefer the model's care plan over the experienced caregiver's care plan and for dementia one caregiver prefers the model's care plan.
The number of actions caregivers would add to the heart failure care plan when they have chosen the model's care plan is between 3-5. They would remove 0-2 actions from the care plan. The caregiver who chose the model's plan for dementia would have added 2 actions and removed none.

5 DISCUSSION

In this section, we discuss the results, the influence of missing data and alterations to the data, the relevance of this research for Ecare and home care in general, and relevant recommendations.

5.1 Interpretation of results

To interpret the results, we look at the following aspects: the different types of datasets, models, and the influence of the main illness. We also discuss the comparison of the models to the care plans drawn up by experienced caregivers.

5.1.1 Problem versus difference datasets.

As shown in the results, the test sets of the problem datasets consistently outperform the test sets of the difference datasets. This

[1]One caregiver of organisation 1 only assessed the heart failure client.

can be explained by the fact that the difference datasets experience more sparsity, which leads to poorer results. We do see that the performance of the difference dataset's train set is sometimes higher than the problem datasets (e.g. BR-SVM for all three diseases) even though the test set is lower. This indicates that the model is overfitting on the train set. The sparsity of the difference datasets can once more explain this phenomenon, because a model trained on sparse data is more likely to overfit [37]. The problem datasets, therefore, capture the relationship between the independent and dependent features better and are preferred as input sets over the difference datasets.

5.1.2 Comparison different models.

The binary relevance models have a higher performance on the F1-score and recall than the models that predict all labels at once. As said in the experimental results, the BR models predict fewer false negatives. This is desirable, since one does not want to miss essential actions that will contribute to a client's recovery. This finding is interesting, since the BR models do not consider label dependency and thus miss information about labels that occur often together or that do not occur together at all. These outcomes show that for this particular research, the models perform better when they look at each action individually rather than the whole action set. The multi-label classification research of Huang [45] has looked into the comparison of ML-KNN and BR models for the labelling of tweets. This research has shown that for a multi-label classification problem with six labels, ML-KNN outperforms BR. This does not match with our results. However, the study of Luaces [46] shows that if a multi-label classification problem has a substantial amount of labels (in Luaces' research, more than 75 labels), the BR is competitive and even shows better performance than ensemble methods. They concluded that BR is competitive with more complex approaches when the proposed data has a large number of labels and high label dependency. This would explain why, for this research, the BR models perform better than the other two models, since all three illnesses have a substantial amount of labels and there is a high amount of label dependency present. It also explains why the research of Huang demonstrates that ML-KNN has a higher performance than BR, since they only have six possible labels for their problem.

When we compare the BR-RF and BR-SVM, we see that the BR-RF outperforms the BR-SVM, especially for the problem datasets. This matches the results of the research of Song [47]. Song investigated EHR data as input for models. Their results show that random forest outperforms the SVM when only the scalar data of the EHR is used, which matches our discoveries. However, their research tackles a binary classification problem instead of a multi-label classification problem, so the effect of multiple labels on the performance of the models is not shown. If we look at the research of Alonso [48], we see that RF and SVM perform very similarly for a multi-label classification problem with five labels. These results slightly differ from our results, but this difference can occur because the dependent feature set of Alonso experiences less sparsity. Random forest is known to handle sparsity quite well [49].

When we look at the difference between ML-KNN and MLP, we see that they have very similar performance for the test groups. The research of Rajput [30] and the study of Zhang [29] show corresponding results. They demonstrate that for a multi-label classification problem with six and five possible labels, respectively, the ML-KNN and MLP have very similar performance. This is in line with our results.

5.1.3 Comparison between illnesses.

When we compare the performance of the models between the three different illnesses, we see that the models for heart failure have a slightly better overall performance than those for dementia and diabetes. This can be explained by the fact that heart failure has the least amount of labels that need to be predicted, but the most independent features. It is, therefore, easier to find valuable relations between the independent features and the labels.

5.1.4 Hamming loss.

Figure 10 shows that all hamming loss values of the test set are between 0.1-0.16, except for the test set of the problem datasets with BR-RF, which has a hamming loss between 0.058-0.09. These hamming losses are generally low and suggest that the models work quite well. But if we put these values into context and look at the recall as well, we can conclude that the hamming loss is low due to the fact that most labels have been given a zero as a prediction. These zeros are predicted often, because it is more common that an action is not present than that it is part of the care plan, even though this is not necessarily the best suggestion. This is shown in the recall, which takes the false negatives into account. The recall is always too low for the models to be implemented in real life. This shows that it is essential to look at the performance metrics as a whole and not individually, since the hamming loss on itself gives a distorted picture.

5.1.5 Comparison BR-RF's and experienced caregiver's model.

Table 5 shows that only for heart failure multiple caregivers have chosen the model's care plan. For dementia and diabetes, the caregivers almost always preferred the experienced caregiver's care plan. Two of the three caregivers who did not choose the model's plan for heart failure knew the client at hand, so they could have been biased in their decision. The same goes for one dementia and one diabetes client. These results are in line with the F1-score, precision, recall and hamming loss, which all have better performance values for heart failure than dementia and diabetes.

When we look at the changes the caregivers would have made to the model's care plan for heart failure, we see that on average they would have added 4.3 actions and removed 1.5 actions. For the dementia care plan, the caregiver would have added 2 actions. This indicates that even though they prefer the model's care plan, the care plan is still not without its shortcomings.

Considering the notes the caregivers have provided while making a decision, we see the following trends:

- They prefer the model's plan for heart failure, because it focuses more on providing instructions and less on treatments;
- They think the model's care plans for dementia are too lengthy;
- For dementia, the caregivers of organisation 2 conclude that there are actions in the model's care plan that do not make sense (care of urinary tract, while there is no indication that the client has urinary tract problems).

These notes show that sometimes fewer actions suit the situation better and that the models, in this case for dementia, can provide

suggestions that do not fit the client-specific context. The latter can create undesirable situations when inexperienced caregivers blindly trust the provided actions of the model and copy the suggestions without considering the client-specific situation.
Taken together, the findings in this section display that there is promise in the models, especially for heart failure, but that they do not work well enough to implement them right now without facing unsafe consequences. Even if you would use the model as a starting point to save time and let caregivers make adaptations to the model, the risks are too high. If a caregiver still forgets to remove or add an important action, the care plan could result in unsuitable care.

5.1.6 *Performance of the models w.r.t. implementation.*

The best-performing model is the BR-RF with the problem dataset. However, this model only reaches an F1-score of 0.78 for heart failure, 0.67 for dementia and 0.69 for diabetes. These values are too low to implement these models and provide suggestions while caregivers draw up care plans. Especially since the recall is low, as mentioned before. This is also shown in the comparison of the models and an experienced caregiver, where only for heart failure the model's care plan is preferred. These outcomes illustrate that there is potential, but they also suggest that as of now, the models are not suitable for implementation. This research aims to see if machine learning can provide suggestions for actions that are mostly forgotten (among other aspects), but with a low recall, the models cannot be used for this aspect right now.

5.2 Assumptions and data modifications

The first main assumption for this research is that every experienced caregiver always creates perfect care plans (and thus never makes mistakes). This assumption has been drawn up in consultation with Ecare, but still influences the used data. Even though the models do not perform as desired, we still have to take into account that the used data could bias the models due to these assumptions. This could lead to models creating suggestions based on partly noisy data. However, it is difficult only to include care plans correctly drawn up with 100% certainty.
The following assumption we have made with respect to the used data is that a client's current (or last known) care plan is most suitable for that client. We have not considered older versions of care plans, since we have assumed that the latest version of the care plan would suit the client best at that moment. However, there is always the possibility that the latest version of the care plan contains actions that do not contribute to a client's recovery or misses essential information. Especially if we link this back to the assumption that every caregiver always creates perfect care plans, which we cannot guarantee. Nonetheless, we believe that the chances of this situation occurring are slim and have little to no effect on the performance of the model.
The next alteration that has an impact on the performance of the models is handling the unknown values and the adjustment of the range of the age feature. For the categorical features, we have applied distribution based imputation. By replacing the categorical unknown values with a value based on the distribution of the dataset, we have kept the distribution of these independent features intact, but we have not looked if the replaced values make sense on a client-specific level. For the age feature, we have applied value based imputation by replacing the impossible values with the mean value. Again, we have not looked if these values are logical on a client-specific level. In general, distribution based imputation is preferred over value based imputation for machine learning applications [50]. Nonetheless, we think that for the age feature, the influence would be minimal to change value based imputation to the distribution variant, especially since the most occurring age is always within a range of two years of the mode. Furthermore, we believe that looking at a client-specific level is not worth the time, since it is computationally expensive to verify the feasibility of the new value for each client and using standard imputation techniques should tackle the change of picking an infeasible new value.
The last alteration that we have performed on the data that impacts the models' performance is keeping the label dependency intact during the resampling phase of the data. This has resulted in a slight imbalance in the number of appearances between the labels, due to the fact that the labels that appear more often correlate with a higher number of other labels than the ones that almost never appear. We have chosen this approach, because we wanted to keep the label dependency intact, since we believe that a lot of valuable information can be extracted from the combination of actions within a care plan. However, for the BR based models, resampling per label could have been an approach, since these models in general do not take label dependency into account. Despite that, we have chosen to use the same input data for all models to have an unbiased comparison between the models. Another option could have been removing the labels that almost never appear in a care plan to create more balance among the labels. However, we have decided against this, since the actions that rarely appear are the most interesting actions for a care plan. Those actions make a care plan really client-specific and can add valuable information to a care plan.

5.3 Limitations of the data

A drawback of this research is the amount of independent features we have used. Because we have only used structured data in this research and removed the features where less than 10% of the data was available, the number of possible independent features for each illness was limited. It even resulted in fewer independent features than labels for dementia with the problem dataset. As said in the interpretation of the results, the results show that the heart failure models have a better performance in general, which can be explained by the fact that there are proportionally more independent features than labels. However, adding extra independent features does not always result in better-performing models, as this research also shows. The difference datasets do have more independent features than the problem datasets, but have a lower performance. This is due to the sparsity of the datasets. Therefore, extra independent features must be selected carefully and sparse datasets should be avoided.
Furthermore, we have only taken the data of five healthcare organisations into account. This can lead to unintentional bias, because we have not assessed the demographic variations between the used organisations and the other organisations that are part of Ecare.

This makes the models less generalisable, and it creates more difficulty to apply the models for all organisations that are part of Ecare, especially if it turns out that there is a difference in clients between the organisations.
Lastly, we have only looked at one example care plan at the time, while clients can have multiple illnesses simultaneously. This is a considerable simplification with respect to the real-life situation. Even though the example care plans contain all the essential actions for a specific illness, they do not take the influence of having multiple illnesses into account. A combination of illnesses present in a specific client can lead to extra needed actions for the other illnesses that are currently not considered. This can lead to incomplete care plans. Even though the models do not work properly for the situation where one example care plan is used, it must be noted that the current set-up is thus a simplification, especially since 80.55% of the currently used clients have two or more illnesses. A combination of care plans will lead to a higher number of possible actions (and thus labels), but it will also create a more accurate representation of the possible actions for a client.

5.4 Ethical consequences

During this research, the models have already been compared to experienced caregivers, but have not yet been integrated within the actual design phase of the client-specific care plans. Section 5.1.5 has shown that for now, the models are not ready to be implemented. Nonetheless, it is important to already look into the ethical consequences when such a model will be implemented, and discuss the implications that will arise. Machine learning addresses three areas of ethical concern: privacy and surveillance, bias and discrimination, and the role of human judgement [51]. Especially the role of judgement plays a role in this research. This area is extremely relevant, because the implementation of the models will influence the decision-making process of the caregivers and thus their judgement. It will also raise the question to what extent a model can take over the tasks of a caregiver and to what extent this is even desirable [51]. Especially since machine learning makes suggestions appear like a standalone objective statement rather than showing that suggestions are based on connections within historical data. In our research, the models must only support, and a caregiver should always make the final decision. This is essential, since caregivers deal with client-specific situations that are more often an exception than a rule, and only trained medical personnel can assess the situation at hand. A model in home care should, thus, only have an advisory position and never create a complete client-specific care plan without the involvement of a caregiver.

5.5 Recommendations and future research

We do not recommend using any of the models proposed in this report based on the current datasets to suggest actions for client-specific care plans. Making predictions on medical data and people must be done carefully, and it is, therefore, not responsible to use these models with the current assumptions and outcomes of the performance measurements. Especially since the values of the performance metrics are too low and suggestions must be given carefully. However, we do have some recommendations to improve the performance of our models.
The first recommendation is to extend the independent feature set and add personal information. The research of Song [47] shows that extracting information from notes from EHRs increases the performance of machine learning models. There is a lot of client-specific information available in the personal information text fields that could increase the performance of the models, such as types of medicine used, emotional state, hospital visits, character traits, family history or substance abuse. To extract interesting information from these text fields, text mining has to be applied. One effective method for text mining is BERT (Bidirectional Encoder Representations from Transformers), designed by Google AI Language [52]. The main advantage of the structure of BERT is that it takes the semantic meaning, i.e. the relationship between words and sentences, into account. This is useful, because the context of words is more important than the definition of a word itself. There are already many pre-trained models available that use BERT, which can be used as a starting point to incorporate the personal information in the models. The University of Groningen has even designed a BERT model that is pre-trained on Dutch data, named BERTje [53] that is also partly trained on healthcare-related text. We recommend adding the interesting independent features to the problem datasets, since these datasets outperform the difference datasets.
The second recommendation is to use multiple example care plans at once instead of using only one example care plan for a model. By combining the example care plans, the model will be able to take multiple illnesses into account at once. This can lead to more accurate client-specific care plans, since more than 80% of the clients have multiple illnesses. One approach is to apply binary relevance models and train the models for each possible label. When the illnesses of a client are known, the model will only predict the labels that are related to the illnesses. The upside of this approach is that the labels can be filtered on the relevant actions while providing suggestions. The downside is, once again, that label dependency cannot be taken into account when making predictions. Another method that could be applied is the Label Powerset method, which creates a separate class for each possible label combination [44]. This will tackle part of the sparsity problem that is created when you combine multiple example care plans, but the data needs to be resampled heavily to compensate for the label combinations that are almost never present.
Based on the comparison between the experienced caregiver's and the model's care plan, we also suggest examining the advantages of providing suggestions over applying treatments for heart failure clients and exploring what makes care plans lengthy. These are the two main points that were addressed in the caregivers' notes. This could provide insights into why the models show potential, but currently do not completely work as intended. These insights can help improve the models by for example only suggesting the top five-ten suitable actions or mainly focusing on actions that are advise-related, depending on the situation.
Lastly, when a model is implemented in PUUR, it is important to make sure that it will increase either the quality of the care plans or decrease the time of creating a care plan while the quality stays the same (preferably both). This can be researched in a case study where, for a small amount of clients, care plans are created using the suggestions of the model and compare the recovery time and overall satisfaction of clients with a control group, and measure

the time that it takes to create a care plan. Furthermore, it is of great importance that when a model is implemented in PUUR, it should only support caregivers and not take over the decision process of the caregivers. The judgement of the caregivers should always be prioritised over the suggestions of the models, as stated in Section 5.4. It is, therefore, of great importance that the influence of visualisation techniques on the suggestions is researched and taken into account to make sure that the model really contributes to creating care plans and does not have a counterproductive effect.

6 CONCLUSION

Creating client-specific care plans is a complex and time-consuming task. In this research, we have presented several supervised machine learning models that can support this process by suggesting actions for client-specific care plans. We have focused on clients with one of the following illnesses: heart failure, dementia or diabetes. The compared models for this specific problem are ML-KNN, MLP, BR-RF and BR-SVM, and are based on the client's EHRs and predefined example care plans.

After fine-tuning, the BR-RF shows the best performance for all three illnesses with a weighted precision between 0.84-0.89, a recall between 0.53-0.63 and an F1-score between 0.63-0.78. We have also looked at the hamming loss, which is between 0.05-0.1. These hamming losses indicate that the models work quite well, but if we put them into perspective and look at the recall as well, we can conclude that the hamming loss is low due to the fact that most labels have been given a zero as a prediction.

To see the effects of machine learning in a real-life environment, we have compared care plans created by the BR-RF models to care plans created by an experienced caregiver. From the eight caregivers that compared three client-specific situations, six chose the model's care plan over the experienced caregiver's care plan.i Looking at the performance of the models and the comparison with the caregiver, the models are currently underperforming and can, therefore, not be used in a real-life environment. Additional data from personal information must be gathered to increase the performance of the models, and the models need to be revised after alterations before implementing them in the software.

ACKNOWLEDGMENTS

First, I would like to thank Ecare and especially Noah Letwory for giving me the chance to do interesting research in my field of study. I have learned a lot at Ecare and am glad to have worked in such an open company with helpful colleagues. Noah really helped me with all our (brainstorming) sessions and gave me many new insights that have made a big difference during my thesis. Next, I would like to thank Sebastian Rachuba for the guidance and supportive conversations that were very helpful for my process, especially when things did not work the way as planned. I also want to thank Abhishta Abhishta for the insights he has given me regarding the processing of the data. Lastly, I want to thank my family and fellow graduating peers for all the coffee breaks, pep talks and *'gezelligheid'* that made graduating during the summer a bit better :). I really appreciate the help and input I have received during my thesis, so thank you all!

- Cindy Pistorius, October 29, 2023, Enschede

REFERENCES

[1] Vektis. *Feiten en cijfers over wijkverpleging | Vektis.nl.* 2022. URL: https://www.vektis.nl/intelligence/publicaties/factsheet-wijkverpleging-2022.

[2] K S Martin. *The Omaha System: A Key to Practice, Documentation, and Information Management.* Elsevier Saunders, 2005. ISBN: 9780721601304. URL: https://books.google.nl/books?id=2hdtAAAAMAAJ.

[3] Eliezer Bose et al. "Machine Learning Methods for Identifying Critical Data Elements in Nursing Documentation". In: *Nursing Research* 68.1 (Jan. 2019), pp. 65–72. ISSN: 15389847. DOI: 10.1097/NNR.0000000000000315. URL: https://journals.lww.com/nursingresearchonline/Fulltext/2019/01000/Machine_Learning_Methods_for_Identifying_Critical.9.aspx.

[4] Omaha System Support. *Het Omaha classificatiesysteem.* 2014. URL: https://www.omahasystem.nl/over-omaha-system/werken-met-het-omaha-system/.

[5] Batta Mahesh. "Machine learning algorithms-a review". In: *International Journal of Science and Research (IJSR).[Internet]* 9 (2020), pp. 381–386.

[6] L Guo et al. "Multi-label classification methods for green computing and application for mobile medical recommendations". In: *IEEE Access* 4 (2016), pp. 3201–3209. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2578638.

[7] Márcia Gonçalves de Oliveira, Patrick Marques Ciarelli, and Elias Oliveira. "Recommendation of programming activities by multi-label classification for a formative assessment of students". In: *Expert Systems with Applications* 40.16 (2013), pp. 6641–6651. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2013.06.011. URL: https://www.sciencedirect.com/science/article/pii/S0957417413003916.

[8] N Yanes et al. "A Machine Learning-Based Recommender System for Improving Students Learning Experiences". In: *IEEE Access* 8 (2020), pp. 201218–201235. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3036336.

[9] Min-Ling Zhang and Zhi-Hua Zhou. "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization". In: *IEEE Transactions on Knowledge and Data Engineering* 18.10 (2006), pp. 1338–1351. ISSN: 1558-2191. DOI: 10.1109/TKDE.2006.162.

[10] Jingcheng Du et al. "ML-Net: multi-label classification of biomedical texts with deep neural networks". In: *Journal of the American Medical Informatics Association* 26.11 (Nov. 2019), pp. 1279–1285. ISSN: 1527-974X. DOI: 10.1093/jamia/ocz085. URL: https://doi.org/10.1093/jamia/ocz085.

[11] Djavan De Clercq et al. "Multi-label classification and interactive NLP-based visualization of electric vehicle patent data". In: *World Patent Information* 58 (2019), p. 101903. ISSN: 0172-2190. DOI: https://doi.org/10.1016/j.wpi.2019.101903. URL: https://www.sciencedirect.com/science/article/pii/S0172219018300851.

[12] Yanmin Sun, Andrew K C Wong, and Mohamed S Kamel. "Classification of imbalanced data: A review". In: *International journal of pattern recognition and artificial intelligence* 23.04 (2009), pp. 687–719. ISSN: 0218-0014.

[13] Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. "A review of methods for imbalanced multi-label classification". In: *Pattern Recognition* 118 (2021), p. 107965. ISSN: 0031-3203.

[14] X. Y. Liu, J Wu, and Z. H. Zhou. "Exploratory Undersampling for Class-Imbalance Learning". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.2 (2009), pp. 539–550. ISSN: 1941-0492. DOI: 10.1109/TSMCB.2008.2007853.

[15] Francisco J Castellanos et al. "Oversampling imbalanced data in the string space". In: *Pattern Recognition Letters* 103 (2018), pp. 32–38. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2018.01.003. URL: https://www.sciencedirect.com/science/article/pii/S0167865518300035.

[16] Haseeb Ali et al. "A review on data preprocessing methods for class imbalance problem". In: (Oct. 2019), pp. 390–397. DOI: 10.14419/ijet.v8i3.29508.

[17] M Zeng et al. "Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data". In: *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS).* 2016, pp. 225–228. DOI: 10.1109/ICOACS.2016.7563084.

[18] Gustavo Batista, Ana Bazzan, and Maria-Carolina Monard. *Balancing Training Data for Automated Annotation of Keywords: a Case Study.* Jan. 2003, pp. 10–18.

[19] K Chen, Bao-Liang Lu, and J T Kwok. "Efficient Classification of Multi-label and Imbalanced Data using Min-Max Modular Classifiers". In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings.* 2006, pp. 1770–1775. ISBN: 2161-4407. DOI: 10.1109/IJCNN.2006.246893.

[20] Min-Ling Zhang et al. "Towards class-imbalance aware multi-label learning". In: *IEEE Transactions on Cybernetics* 52.6 (2020), pp. 4459–4471. ISSN: 2168-2267.

[21] Jose M Moyano et al. "Review of ensembles of multi-label classifiers: Models, experimental study and prospects". In: *Information Fusion* 44 (2018), pp. 33–45. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2017.12.001. URL: https://www.sciencedirect.com/science/article/pii/S1566253517307169.

[22] Grigorios Tsoumakas et al. "Correlation-based pruning of stacked binary relevance models for multi-label learning". In: *Proceedings of the 1st international workshop on learning from multi-label data.* 2009, pp. 101–116.

[23] Guo Haixiang et al. "Learning from class-imbalanced data: Review of methods and applications". In: *Expert Systems with Applications* 73 (2017), pp. 220–239.

ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2016.12.035. URL: https://www.sciencedirect.com/science/article/pii/S0957417416307175.

[24] Yanmin Sun et al. "Cost-sensitive boosting for classification of imbalanced data". In: *Pattern Recognition* 40.12 (2007), pp. 3358–3378. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2007.04.009. URL: https://www.sciencedirect.com/science/article/pii/S0031320307001835.

[25] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. "A review of multi-label classification methods". In: *Proceedings of the 2nd ADBIS workshop on data mining and knowledge discovery (ADMKD 2006)*. 2006, pp. 99–109.

[26] Min-Ling Zhang et al. "Binary relevance for multi-label learning: an overview". In: *Frontiers of Computer Science* 12.2 (2018), pp. 191–202. ISSN: 2095-2236. DOI: 10.1007/s11704-017-7031-7. URL: https://doi.org/10.1007/s11704-017-7031-7.

[27] Min-Ling Zhang and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning". In: *Pattern Recognition* 40.7 (2007), pp. 2038–2048. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2006.12.019. URL: https://www.sciencedirect.com/science/article/pii/S0031320307000027.

[28] Anke Meyer-Baese and Volker Schmid. "Foundations of Neural Networks". In: *Pattern Recognition and Signal Analysis in Medical Imaging* (Jan. 2014), pp. 197–243. DOI: 10.1016/B978-0-12-409545-8.00007-8.

[29] Liang Zhang et al. "Using multi-label classification for acoustic pattern detection and assisting bird species surveys". In: *Applied Acoustics* 110 (2016), pp. 91–98. ISSN: 0003-682X. DOI: https://doi.org/10.1016/j.apacoust.2016.03.027. URL: https://www.sciencedirect.com/science/article/pii/S0003682X16300603.

[30] Nikhil Kumar Rajput and Bhavya Ahuja Grover. "A multi-label movie genre classification scheme based on the movie's subtitles". In: *Multimedia Tools and Applications* 81.22 (2022), pp. 32469–32490. ISSN: 1573-7721. DOI: 10.1007/s11042-022-12961-6. URL: https://doi.org/10.1007/s11042-022-12961-6.

[31] Lena Tenenboim-Chekina, Lior Rokach, and Bracha Shapira. "Identification of label dependencies for multi-label classification". In: *Working notes of the second international workshop on learning from multi-label data*. Citeseer, 2010, pp. 53–60.

[32] Ying Yu, Witold Pedrycz, and Duoqian Miao. "Multi-label classification by exploiting label correlations". In: *Expert Systems with Applications* 41.6 (2014), pp. 2989–3004. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2013.10.030. URL: https://www.sciencedirect.com/science/article/pii/S0957417413008476.

[33] Eva Gibaja and Sebastián Ventura. "Multi-label learning: a review of the state of the art and ongoing research". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4.6 (2014), pp. 411–444. ISSN: 1942-4787.

[34] Anwesha Law and Ashish Ghosh. "Multi-label classification using a cascade of stacked autoencoder and extreme learning machines". In: *Neurocomputing* 358 (2019), pp. 222–234. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.05.051. URL: https://www.sciencedirect.com/science/article/pii/S092523121930757X.

[35] Omar Y Al-Jarrah et al. "Efficient Machine Learning for Big Data: A Review". In: *Big Data Research* 2.3 (2015), pp. 87–93. ISSN: 2214-5796. DOI: https://doi.org/10.1016/j.bdr.2015.04.001. URL: https://www.sciencedirect.com/science/article/pii/S2214579615000271.

[36] D Rohaan, E Topan, and C G M Groothuis-Oudshoorn. "Using supervised machine learning for B2B sales forecasting: A case study of spare parts sales forecasting at an after-sales service provider". In: *Expert Systems with Applications* 188 (2022), p. 115925. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.115925. URL: https://www.sciencedirect.com/science/article/pii/S0957417421012793.

[37] Gareth James et al. *An Introduction to Statistical Learning*. 2nd ed. Springer New York, NY, 2021.

[38] Francisco Charte et al. "MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation". In: *Knowledge-Based Systems* 89 (2015), pp. 385–397. ISSN: 0950-7051.

[39] Rodolfo M Pereira, Yandre M G Costa, and Carlos N Silla Jr. "MLTL: A multi-label approach for the Tomek Link undersampling algorithm". In: *Neurocomputing* 383 (2020), pp. 95–105. ISSN: 0925-2312.

[40] Charu C Aggarwal. "Neural networks and deep learning". In: *Springer* 10.978 (2018), p. 3.

[41] Indraneel Dutta Baruah. *Activation Functions and Loss Functions for neural networks — How to pick the right one?* July 2021.

[42] Saurabh Karsoliya. "Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture". In: *International Journal of Engineering Trends and Technology* 3.6 (2012), pp. 714–717. ISSN: 2231-5381.

[43] Gaurang Panchal et al. "Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers". In: *International Journal of Computer Theory and Engineering* 3.2 (2011), pp. 332–337.

[44] Dhatri Ganda and Rachana Buch. "A survey on multi label classification". In: *Recent Trends in Programming Languages* 5.1 (2018), pp. 19–23.

[45] Anzhong Huang et al. "Research on multi-label user classification of social media based on ML-KNN algorithm". In: *Technological Forecasting and Social Change* 188 (2023), p. 122271. ISSN: 0040-1625.

[46] Oscar Luaces et al. "Binary relevance efficacy for multilabel classification". In: *Progress in Artificial Intelligence* 1 (2012), pp. 303–313. ISSN: 2192-6352.

[47] Jiyoun Song et al. "Clinical notes: An untapped opportunity for improving risk prediction for hospitalization and emergency department visit during home health care". In: *Journal of Biomedical Informatics* 128 (2022), p. 104039. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2022.104039. URL: https://www.sciencedirect.com/science/article/pii/S1532046422000557.

[48] Dorian Ruiz Alonso et al. "Multi-label classification of feedbacks". In: *Journal of Intelligent & Fuzzy Systems* 42.5 (2022), pp. 4337–4343. ISSN: 1064-1246.

[49] Isak Karlsson and Henrik Boström. "Handling sparsity with random forests when predicting adverse drug events from electronic health records". In: *2014 ieee international conference on healthcare informatics*. IEEE, 2014, pp. 17–22. ISBN: 1479957011.

[50] Maytal Saar-Tsechansky and Foster Provost. "Handling missing values when applying classification models". In: (2007).

[51] Christina Pazzanese. "Ethical concerns mount as AI takes bigger decision-making role in more industries". In: *The Harvard Gazette* 26 (2020).

[52] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[53] Wietse De Vries et al. "Bertje: A dutch bert model". In: *arXiv preprint arXiv:1912.09582* (2019).