



PREDICTING CASE INTENSITY FOR DUTCH ASSERTIVE OUTREACH TEAMS

LARS HANEN

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2107443

COMMITTEE

dr. Giovanni Cassani
dr. Mirella De Sisto

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

May 20th, 2024

WORD COUNT

7675

ACKNOWLEDGMENTS

I am sincerely grateful to Dr. Giovanni Cassani from Tilburg University and Dr. Leonard Vanbrabant from GGD West-Brabant for sharing their expertise throughout this project.

PREDICTING CASE INTENSITY FOR DUTCH ASSERTIVE OUTREACH TEAMS

LARS HANEN

Abstract

In Dutch public mental healthcare (OGGZ), Assertive Outreach (AO) teams aim to support disconnected individuals or those causing nuisance. Estimating the intensity of an incoming nuisance case in terms of days and interventions to allocate, is a challenging task for AO team members when relying solely on rudimentary incoming case notification details. However, estimations prior to investigation in the form of unsolicited house visits are useful for intervention scheduling, because at that point interventionists are allocated to a case irreversibly. This task relates to length-of-stay (LoS) prediction which is done for intensive care units (ICU) and hospitals to predict bed occupation with machine learning models using electronic health record (EHR). Known challenges in this field are changing operational factors, cases where the event of interest has not occurred yet by the study's cut-off date and a scarcity of high LoS cases which disparately impact scheduling. We assess the performance of two LoS prediction models, random forests and neural networks, on our case notification variables with and without undersampling of low intensity cases to assess if AO teams can employ state-of-the-art LoS methods. Through access to a real-time AO team database, we propose a method to deal with cut-off cases instead of employing models with impractical assumptions designed specifically to account for them. Furthermore, we show that a feed-forward neural network outperforms a random forest and that it fares best with an undersampling approach. However, both models tend to under-predict high true intensity cases, suggesting the need for further refinement before practical implementation by AO teams.

1 DATA SOURCE, ETHICS, CODE, AND TECHNOLOGY STATEMENT

This study involves processing anonymous data from human individuals within the operational area of GGD West-Brabant, a regional public health-care provider in the Netherlands. The data are stored and processed on

an external server with restricted access, granted to the author by GGD West-Brabant. Only at an aggregated level the already anonymous data are communicated in this study. We added municipal features to enrich our data, sourced from the Dutch 'Centraal Bureau voor de Statistiek' (CBS) - a governmental institution that gathers statistical information about the Netherlands. These features are publicly available through their [Open Data portal](#). The code used in this study is available on [GitHub](#). All figures and tables in this study are made by the authors, Figure 8 with the use of an open source tool from a [GitHub](#) repository (LeNail, 2019). ChatGPT was used to improve the author's original writing, for paraphrasing, spell checking and grammar. No other typesetting tools or services were used.

2 INTRODUCTION

The research goal of this study is to quantify to what extent the intensity of incoming nuisance cases can be predicted for a Dutch regional public healthcare provider, expressed in days and interventions allocated before a case can be closed.

2.1 *Problem statement*

Within the Dutch public mental healthcare (OGGZ) system, so called Assertive Outreach (AO) teams are charged with attempting to detect and support citizens who tend to lose their connection with society and are either incapable or not willing to ask for help themselves. Through regional crisis care notification centres and the AO team's social networks, non-acute cases regarding mental health problems, dementia, seemingly neglected homes or individuals and other forms of nuisance are being reported by fellow citizens and partner care professionals. Following an intake process, AO team members assess cases and determine whether further investigation is warranted, which typically involves paying the subject a number of unsolicited house visits. Their goal is to prepare subjects for referral to further guidance and/or treatment (Roeg et al., 2012). Therefore, the allocation of AO team interventions cease upon issue resolution or the subject's acceptance of other healthcare professionals' aid.

While AO team members can estimate the need for a follow-up intervention by intervention, estimating overall case intensity before additional information accumulates through interventions is challenging. However, early estimation is crucial for interventionist scheduling because once a personal bond of trust with the subject is established through interventions, changing the interventionist becomes undesirable.

2.2 Societal and scientific impact

This study aims to predict the amount of interventions until case closure and the amount of days until the last intervention as proxies for overall case intensity, before additional information accumulates through interventions. Addressing this research objective could enhance the AO team's understanding of interventionists' workload, thus guiding the allocation of cases to the most appropriate interventionist's agenda. This study has societal impact, as more effective allocation of cases to interventionists' schedules may lead to improved timeliness of healthcare provision by the AO team.

The scientific impact of this study lies in its contribution to the understanding of interventionist workload management within mental healthcare settings. Specifically, we evaluate the applicability of existing methods for patient length-of-stay (LoS) prediction in hospitals to address similar challenges faced by AO teams.

3 RELATED WORK

In healthcare, machine learning aids research and decision-making - e.g. by predicting disease and treatment outcomes, detecting anomalies and summarizing clinical notes (Brnabic & Hess, 2021). Together with the introduction of wearables and mobile devices, the adoption of electronic health records (EHRs), has vastly increased data availability for clinical tasks including modeling risk of mortality, forecasting hospital length of stay and detecting physiologic decline (Harutyunyan et al., 2019). Predicting case intensity for AO teams relates most to a task named length of stay (LoS) prediction, which aims to estimate the amount of days a patient will occupy a hospital or intensive care unit (ICU) bed (Lacerda & Pappa, 2021; Naemi et al., 2021; Wen et al., 2022). In our context, case closure serves as the event of interest instead of discharge. The task's difficulty lies in its dependence on possibly changing operational factors, the scarcity of high LoS cases and cases for which the event of interest has not occurred yet on the cut-off date for a study (censored cases).

3.1 Length of Stay (LoS) prediction for hospital patients

ICU patients require close monitoring, which makes it a data-rich environment where data is typically stored in EHRs (Johnson et al., 2023). Being publicly accessible, the Medical Information Mart for Intensive Care (MIMIC-III) became a benchmark EHR dataset for LoS prediction with mixed data types (Cai et al., 2022; Johnson et al., 2023; Lacerda & Pappa,

2021; Rocheteau et al., 2021). Many different approaches and models have been evaluated on the task, but directly comparing other studies' performance with ours is challenging due to potentially different patient demographics and disease prevalence (Bacchi et al., 2022).

Iwase et al. (2022) used EHR data from a Japanese ICU at Chiba University Hospital to predict short (≤ 1 week) and long (> 2 weeks) stays. Their Random Forest (RF) classifier outperformed a logistic regression model, achieving AU(RO)C values of 0.881 and 0.889, respectively, as reported by Receiver Operating Characteristic (ROC) curves. (Zelege et al., 2023) showed that their XGBoost regressor, another tree-based model, and a linear model, a Ridge regressor, performed best among 8 regression models. Their Ridge regressor outperformed other penalized linear models, namely a Lasso and an ElasticNet regressor.

Deep learning techniques have been evaluated on the task of LoS prediction as well. Cai et al. (2022) proposed a Deep Ordinal neural network for Length of stay Estimation (DOSE) in intensive care units. This work framed the task of LoS estimation as an ordinal classification/regression task, decomposed into a series of binary classifiers trained on less imbalanced samples of the training data, yet the task's framing across studies varies between regression, binary and multi-class classification (Cai et al., 2022; Lacerda & Pappa, 2021). Like most works predicting LoS with neural networks, Cai et al. (2022) experimented with a Long Short-Term Memory (LSTM) architecture because they have shown to leverage the temporal dependencies of EHR data (Lacerda & Pappa, 2021; Rocheteau et al., 2021). Rocheteau et al. (2021) compared an LSTM's performance against a Temporal Pointwise Convolution (TPC) model. This model is designed specifically to address the common challenges of LoS data by combining temporal convolution with pointwise convolution in parallel. It significantly outperformed an LSTM and baseline mean and median predictions, on both EHR datasets they evaluated it on¹.

While most studies ignore LoS data skewness, Naemi et al. (2021) concluded that this may lead to underprediction of LoS and unreliable results. Based on R^2 scores, all models they experimented with performed significantly better after applying resampling techniques. Alternatively, Rocheteau et al. (2021) chose to address the same challenge by employing a Mean Squared Log Error (MSLE) loss function instead of the commonly used Mean Squared Error (MSE). MSLE proportionally penalizes errors by taking the logarithm of the predicted and actual values before computing the squared difference - e.g. a 5-day error is worse in a 2-day context than in a 100-day context. To address skewness they also select a model that

¹ The two benchmark EHR datasets used by this study were 'eICU' and 'MIMIC-IV'. The latter is a more recent version of MIMIC-III.

performs well across six different evaluation metrics. For example, they pointed out that even baseline mean and median predictions lead to good Mean Absolute Deviation (MAD) and MSE metrics due to skewness, but the Kappa score exposes their incompetence compared to more meaningful models. Some studies frame LoS prediction as a classification problem after binning the target, but although this might simplify the task this comes at a cost of utility in terms of scheduling (Rocheteau et al., 2021).

Another common challenge in LoS prediction arises from data points for which the event of interest, e.g. recovery, discharge or in our study case closure, has not occurred yet on the cut-off date for a study. Such data points are being referred to as censored cases. This is less important for studies using MIMIC-III given the relatively short LoS durations compared to the study's time span than it is for us, as in our dataset some cases remain opened for hundreds of days (Figure 1). Conventional regression models typically discard cases that did not experience the event of interest by the cut-off date of the study, which may introduce bias (Wen et al., 2022). In a statistical method named time-to-event modeling, also known as survival analysis, such data points still contribute to the analysis by considering them 'right-censored' instead of discarding them. This is valuable because observing the absence of an event until a certain amount of time is informative. The objective of time-to-event models in LoS context is to model the LoS distribution as a function of time, enabling them to predict the amount of time until the event of interest occurs (Wen et al., 2022). However, one of the important underlying assumptions is non-informative censoring. In our case this assumption would be violated if a case's likelihood of being censored relates to the subject's severity of problems or other factors that influence case duration. Another violation of these models' assumptions would take place if the probability of case closure depends on the conditions of other cases' progress, referred to as competing events (Piovani et al., 2021). We can not exclude either violation, hence we propose a different method to deal with censored cases, driven by the real-time character of our dataset (Sections 4.5 and 4.2.3).

Some conceptual differences with LoS prediction apply to our task for AO teams. Namely, EHR integration by AO teams is disregarded due to significant cross-organizational privacy challenges. Besides, the AO team's primary concern lies in accurate predictions at the moment of case notification. Therefore, in contrast to LoS studies using EHR cases with temporal dependencies, this study is limited to using rudimentary, one-time case notification details only.

3.2 Research Questions

In order to achieve the main research objective, we pose the following central research question:

To what extent can machine learning algorithms predict the intervention intensity of AO cases, measured as the amount of interventions until case closure and days until last intervention, using rudimentary case notification data?

The central research question leads to the following five sub-questions. First of all, we evaluate the performance of a Random Forest (RF) regressor as it has shown to be state-of-the-art on similar tasks (Iwase et al., 2022). Unlike Iwase et al. (2022) and Cai et al. (2022) we choose to frame the prediction of case intensity as a multi-output regression task rather than a multi-class (ordinal) classification task, avoiding arbitrary class-sizes and -order and optimizing utility. We compare it to the performance of a multi-output feed-forward neural network (FFNN) with joint MSLE loss (Rocheteau et al., 2021). We diverge from related works' use of recurrent neural network (RNN) architectures such as LSTM because our dataset precludes consideration of temporal dependencies of observations within cases and earlier estimations are more valuable. We designate the Ridge regressor as a baseline model due to its inability to capture non-linear relationships in our data, and because in this study it primarily serves the purpose of facilitating better comparisons with other studies. Additionally, we employ baseline mean and median predictions with the same purpose.

1. **To what extent can a RF and a FFNN accurately predict the amount of days and interventions, compared to baseline Ridge, median and mean predictions, through multi-output regression?**

Addressing the potential inconsistency of individual interventionists' judgments, we perform an out-of-sample experiment to evaluate whether the model's predictions are accurate in case of unseen AO team members. Chapter 4.2.3 describes this in more detail.

2. **To what extent do predictions generalize to out-of-sample interventionists' cases?**

Then, as our data is skewed (Figure 2), we examine how reducing skewness affects predictions in AO teams' context, by undersampling rather than producing ambiguous samples with oversampling scarce high-intensity cases (Naemi et al., 2021).

3. What impact does undersampling of low-intensity cases have on predictions?

As shown in Figure 1, few cases have high intensity. Because of their disparate effect on scheduling, we analyze error patterns over varying case intensity.

4. What impact does case intensity have on prediction error?

To the AO team, understanding what case features contribute to intensity predictions is valuable. Therefore, we pose the ultimate question below.

5. How much do the top 25 features contribute to predictions of case intensity?

4 METHOD

4.1 Dataset description

We use a real-time Structured Query Language (SQL) dataset provided by the Dutch public healthcare service provider of the West-Brabant region's AO team, holding approximately 10,000 entered nuisance cases from 2016 onwards with (multi)categorical, boolean and date variables. Through a database connection with a third-party case management system (CMS), it stores (1) rudimentary case notification data and (2) subsequent self-reported intervention logging by team members. Administrative employees manually enter cases into the CMS after receiving notifications from either regional crisis care notification centres or AO team members who receive cases through their personal channels. As soon as cases are entered they are assigned to the team member operating in the region corresponding to the case's origin. They assess whether an intervention is warranted and, if so, register intervention details retrospectively. If not, they register case closure details.

Direct personal data such as names and address fields were excluded, leaving subjects and employees untraceable. The organization's data protection officer gave clearance to the processing of the data after performing a Data Protection Impact Assessment (DPIA).

Our mixed type case notification data consists of variables such as the date of case notification, registered intervention dates, the subject's date of birth, notification source, whether there is a presence of minors or (domestic) violence, and what presumptions the AO team interventionist allocated to the case has regarding the problem. We added annual municipal information to this data sourced from CBS which is publicly accessible

through the link in Section 1. Figure 1 visualizes the relationship between the days and interventions allocated to closed cases of our dataset. Also, it shows that many cases have low overall intensity.

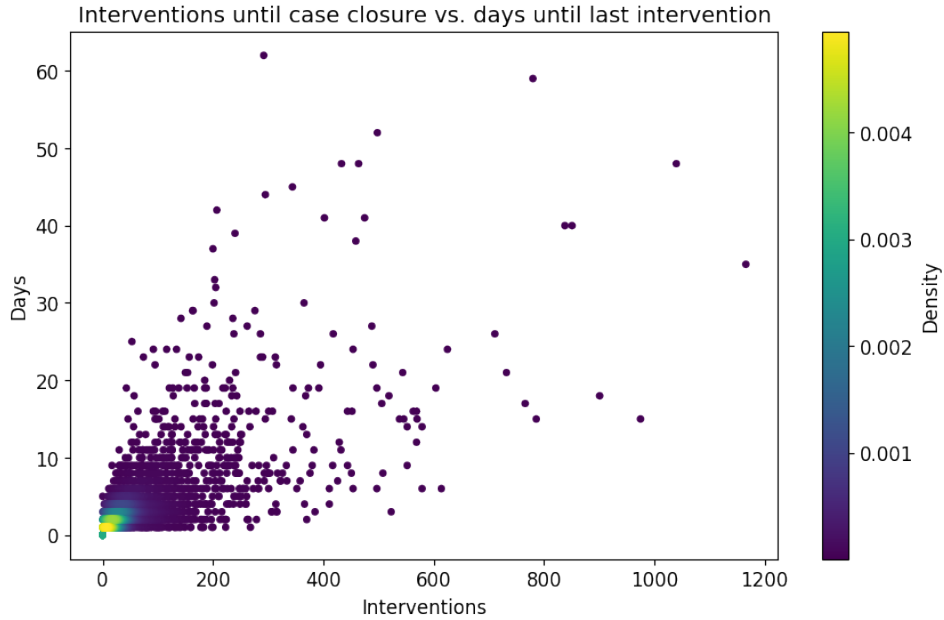


Figure 1: Density plot of the amount of interventions until case closure against days until the last interventions of the entire dataset. The higher the density values, the denser the region is packed with cases.

4.2 Preprocessing

In this section, we outline the preprocessing steps taken to prepare our raw dataset for feature engineering.

4.2.1 Data cleaning

First, we removed cases with notification dates preceding January 8, 2019 as this is the date the AO team started registering interventions. Then, as one case notification source became obsolete over time we removed all cases associated with it. Cases registered under former municipalities that underwent merger were updated with their current municipality names and cases from outside the West-Brabant region were excluded. Finally, some case ID's occur twice in the dataset. We merged those pairs of rows because they turned out to always be complementary, with one row holding case notification variables and the other one holding intervention and closure variables.

4.2.2 Label creation

We construct two label variables as proxies for case intensity: (1) the amount of interventions until case closure and (2) the amount of business days until the last intervention. The calculation of business days includes the end date but excludes the start date, and took into account all Dutch national holidays from 2019 up to and including 2025. If no interventions were executed, the amount of business days was set to zero. As shown in Figure 2, the distribution of labels over the entire dataset are skewed.

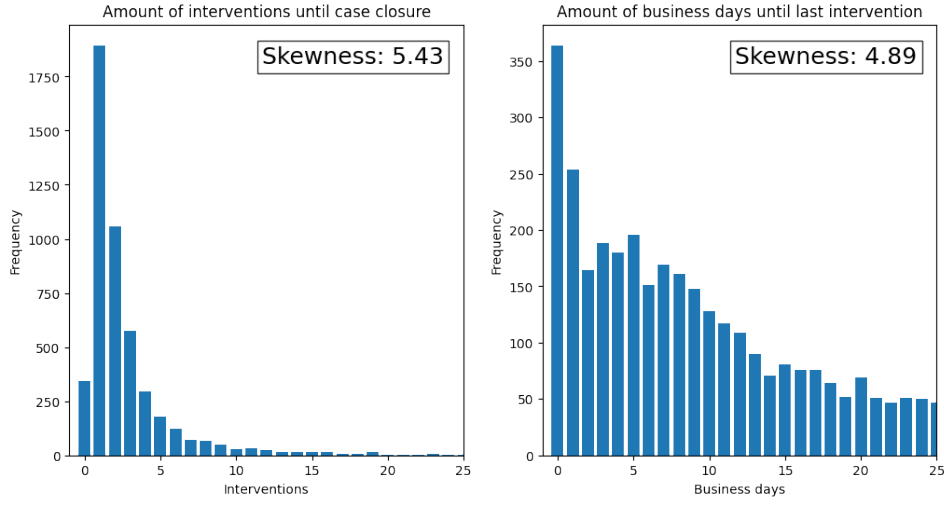


Figure 2: Label distribution of the entire dataset. Both horizontal axes are truncated at $x = 25$ for better visualization of the most dense parts of the distributions.

Both labels were ultimately ‘log1p’ transformed to compress a relatively wide range of values to narrower one, which led to a transformed distribution shown in Figure 3. The transformation can be described by the following formula 1, which involves adding 1 to the label value before taking the logarithm to account for zero values:

$$\text{TransformedLabelValue} = \log(1 + \text{LabelValue}) \quad (1)$$

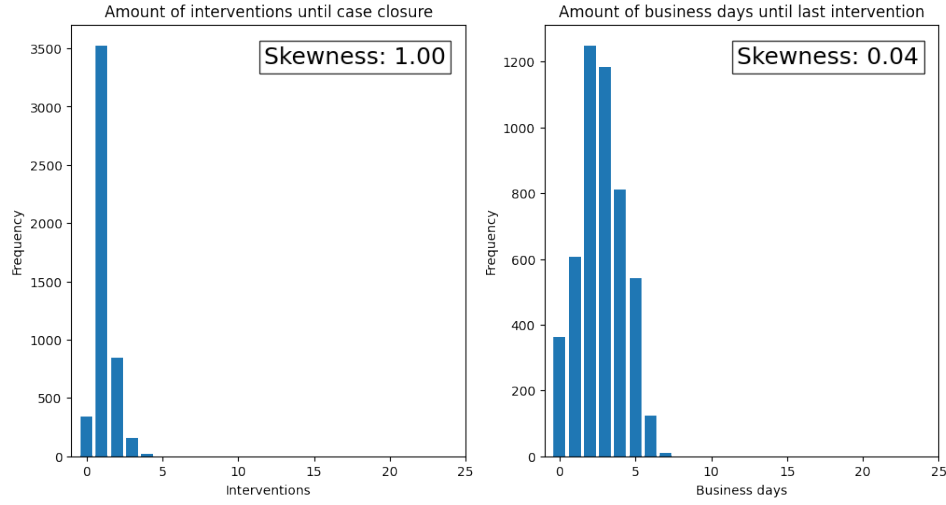


Figure 3: Label distribution of the entire dataset after \log_{1p} transformation. Both horizontal axes are kept equal to those of Figure 2 to emphasize the effect.

4.2.3 Data splitting

TEMPORAL SPLITS As the penultimate preprocessing step, we split the data into four subsets. First, we use case notification dates to split the dataset into a train set, a development set and a test set while preserving chronological order between cases (Table 1). That is, the train set does not contain cases with notification dates more recent than any development or test cases, and the test set exclusively contains cases with notification dates more recent than the train and development sets. This splitting approach represents the AO team interventionists' workflow, as they use lessons learned from past cases to inform their judgment for present cases.

We prioritized maintaining the chronological order while optimizing other subset characteristics. First, the aim was to split a train set with case notifications as recent as possible, while also keeping other subsets large enough to be robust against data variance. Ideally, our model generalizes well to unseen circumstances such as more and less busy periods and new colleagues joining the AO team. Therefore, we optimized our development set to hold over three times the amount of incoming cases per month per interventionist compared to the train set (see Table 1)². No more than one unseen interventionist, representing a new AO team colleague, could be represented in the development set due to the other split optimization constraints.

² Figure 4 represents the amount of cases per subset after all preprocessing steps but before balancing (4.2.4). As preprocessing steps include case in- and exclusion, this figure does not represent the amount of incoming cases per quarter, which is in Table 1.

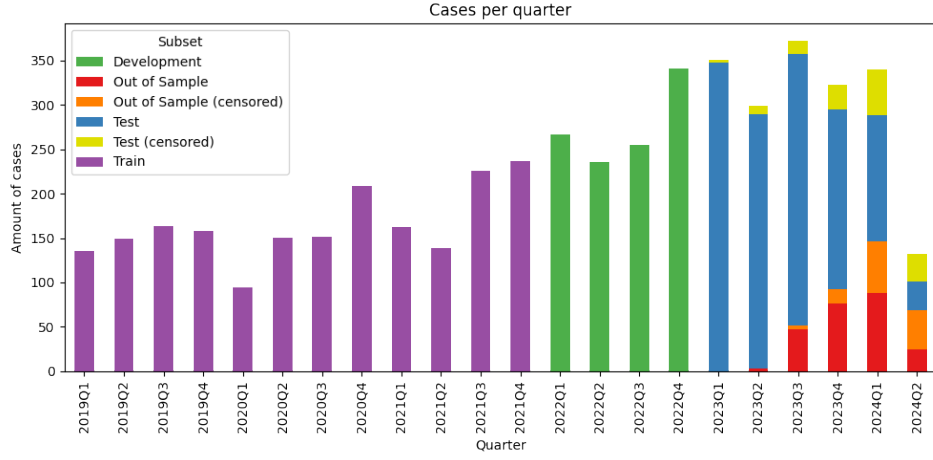


Figure 4: Cases per subset per quarter

Ultimately, we separated test cases from (1) interventionists seen in training and/or development sets, from test cases belonging to (2) unseen interventionists. Test cases from (1) interventionists seen in the train and/or development sets were assigned to a subset that we still refer to as our test set. Cases from (2) unseen interventionists form our out-of-sample (OoS) set (Figure 4). This serves to compare our model’s prediction capabilities in both circumstances. Specifically, interventionists’ judgments regarding intervention allocation directly influence our case intensity labels and it is uncertain how consistent their judgment is. Assessing if predictions generalize to unseen interventionists as well as they do to seen interventionists, is crucial information for considering reliance on predictions for future new colleagues in the AO team.

Split	Year(s)	Unseen ints	% Unseen ints’ cases	Avg. cases per month per int
Train	2019-2021	-	-	2.11
Dev	2022	1	0.5%	6.54
Test	2023-2024	0	0%	4.67
OoS	2023-2024	5	100%	3.01

Table 1: Subset characteristics (‘ints’ = interventionists). The train set holds 26 interventionists’ cases.

RIGHT-CENSORED DATA POINTS So far, we have treated closed and censored cases equally. As table 2 indicates, all subsets contain censored cases on the cut-off date for this study: May 2, 2024. The train and development sets have the fewest censored cases, which matches our

expectations because their case notifications are less recent. Because they are proportionally so little present, we discarded both their cases. For the development set, this also makes model selection easier because we can evaluate models on their performances on closed cases exclusively.

For the test and out-of-sample sets we preserved them, by storing their indices together with our predictions in separate subsets. We take advantage of the real-time character of our dataset by comparing these predictions with the most up-to-date case information we can get, on May 16, 2024 (Section 4.5). Thus, we split two more subsets from the test and Out-of-Sample subsets: the censored test set and the censored Out-of-Sample set.

Split	Year(s)	Cases	Censored cases
Train	2019-2021	1978	4
Dev	2022	1099	9
Test	2023-2024	1457	138
OoS	2023-2024	361	123

Table 2: (Censored) cases per subset. We refer to cases not closed yet on the cut-off date for this study (May 2, 2024) as censored.

4.2.4 Data balancing

Although we could avoid case deletion by upsampling high-intensity cases in our train set, this would also lead to low diversity of synthetic cases because of their scarcity. On the other hand, undersampling until high-intensity cases match the scarcity of low-intensity cases reduces the overall number of cases drastically. For those reasons, for each heavy class of a chosen label variable, we undersampling n cases as a function of the class’s frequency f and a undersampling percentage i . We applied Equation 2 to our days target variable, considering 6 days as our last heavy class and setting the undersampling percentage to 0.6.

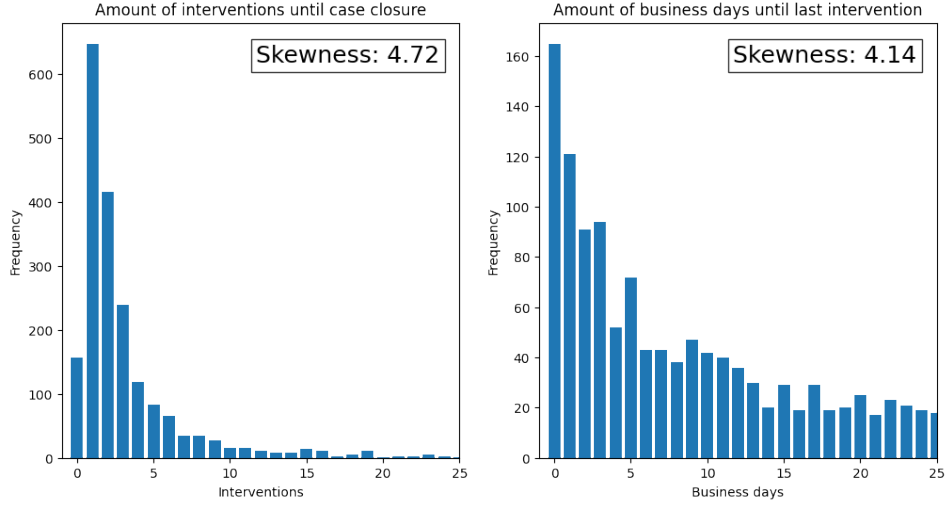


Figure 5: Label distribution of the train set before balancing. Both horizontal axes are truncated at $x = 25$ for better visualization of the most dense parts of the distributions.

$$n = (f_{\text{heavy class}} - f_{\text{first non-heavy class}}) \times i \quad (2)$$

Figures 5 and 6 show the imbalance of both label variables before and after undersampling respectively.

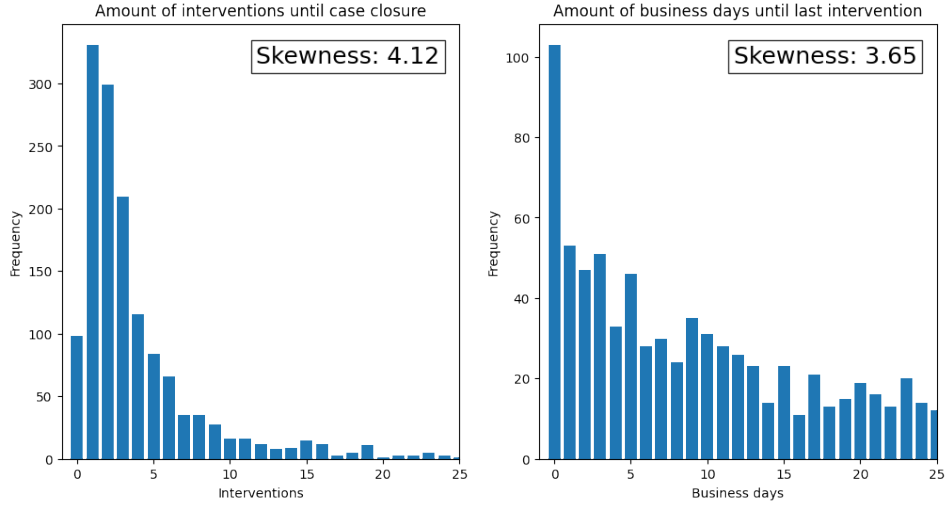


Figure 6: Label distribution of the train set after balancing. Both horizontal axes are truncated at $x = 25$ for better visualization of the most dense parts of the distributions.

4.3 Feature engineering

This section describes how variables from the preprocessed dataset were engineered to features. Eventually, all features were scaled independently using *scikit-learn*'s 'RobustScaler' (Pedregosa et al., 2011).

4.3.1 Dummy-coded features

Four categorical variables, five boolean variables, and one multicategorical variable underwent dummy-coding, with missing values handled as a separate category. This process generated $n - 1$ dummy features for each categorical and boolean variable, and n features for the multicategorical variable, resulting in 52, 14, and 15 features, respectively. Here, n represents the number of categories in the variable. Additionally, a feature representing the number of categories selected in the multicategorical variable was appended.

4.3.2 Date of notification features

We decompose the notification dates of cases into four features: year, month, day of month, and day of week. While the year can be directly extracted, we encode the latter three components using sine functions to capture their cyclical nature (Figure 7). This encoding ensures that each day or month is approximately equidistant from its neighboring time unit steps. Besides capturing cyclical nature, it reduces dimensionality compared to alternatives like dummy-coding.

The date of notification, in combination with the subject's year and month of birth, was also used to construct an age feature.

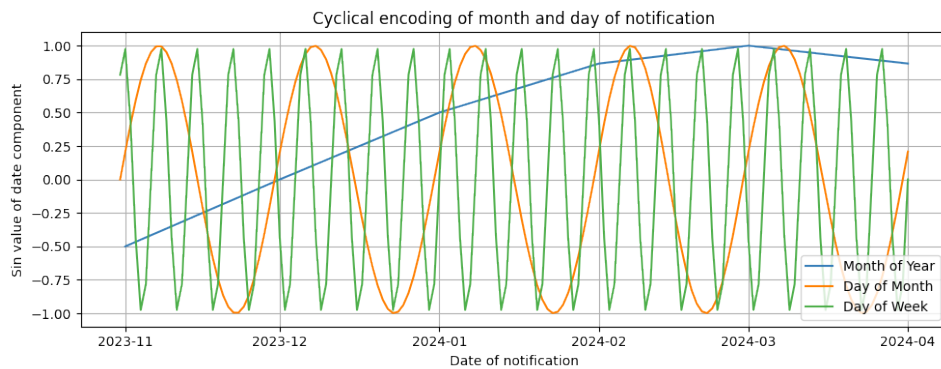


Figure 7: Encoding of cases' notification dates as a sine function.

4.3.3 *Workload features*

We engineered four workload features representing the AO team’s activity in the seven days before the case notification date. Specifically, two features were calculated for the interventionist handling the case: (1) interventions executed and (2) incoming cases. The other two features represent the same but for the entire team. Two other workload features indicate the number of ongoing cases at the notification date, again calculated separately for both the entire team and individual interventionists. Cases excluded from the dataset during cleaning as discussed in section 4.2.1, were included in the workload calculation as they presumably still influenced intervention scheduling.

4.3.4 *CBS municipality features*

The categorical municipality variable was excepted from dummy-coding. Instead, we represent municipalities by appending four annual municipal CBS features: population, population density, population growth per 1000 inhabitants, and social benefit recipients per 1000 inhabitants. If CBS data is not available for the year of notification, we iteratively looked back one year until it is³.

4.4 *Model selection*

We evaluated different hyperparameter setting of the RF and FFNN on our development set. Their hyperparameters were tuned by training a total of 163 models with different settings per train set - an undersampled and a regular one. The hyperparameter search ranges are shown in Table 3. For each train set separately, we choose the best-performing models and retrain it on a combination of the respective train set and development set. For simplicity, we opt not to undersample the development set in either pipeline. Throughout our analysis, all models are evaluated using the same input features. The baseline Ridge’s hyperparamaters were kept untuned for simplicity.

³ We considered CBS data from 2019 onwards. All municipal features of 2019 are present, which guarantees the absence of missing values.

Model	Hyperparameter	Lower	Upper
RF	n_estimators	50	150
	Max depth	10	30
FFNN	Hidden layer neurons	16	256
	Learning rate	0.0001	0.01
	Epochs	50	150

Table 3: Hyperparameter ranges tuned per model. A single non-numeric RF hyperparameter is not included for which we tested ‘squared error’ and ‘absolute error’: the splitting criterion. We tested 3 to 5 values per hyperparameter.

Before evaluating the models’ predictions we applied the following three steps to them:

1. Reverse the log transformation (Section 4.2.2) of both target labels, so we can have readily-interpretable metrics (Section 4.5.1);
2. Set any negative predictions to zero, as case intensity can not be negative;
3. Set any infinite (‘inf’) predictions to the maximum observed intensity in the train set ⁴;
4. Set the days prediction to zero if the interventions prediction is zero, as we assume these cases to have have zero case intensity;

Although our true target variables are integers, we did not round our predictions to have fine-grained evaluations.

4.4.1 Ridge regressor

Next to baseline mean and median predictions that always predict the same value, we employ *scikit-learn*’s multi-output implementation of a linear Ridge regressor as our third and only baseline with learned parameters (Pedregosa et al., 2011). This model builds on the simple strategy of regular linear regression to minimize the difference between the observed and predicted values using techniques such as ordinary least squares (OLS) (McDonald, 2009). Ridge regression uses the same least squares but in combination with L2 regularization to penalize large variations in parameters (Rahman et al., 2022). Our Ridge model’s regularization strength parameter *alpha* is equal to *scikit-learn*’s default setting of 1.

⁴ Negative and infinite values were predicted by our FFNN, both with an undersampling approach and without.

4.4.2 Random Forest regressor

A Random Forest regressor is a tree-based ensemble method capable of handling multi-output regression tasks. Again, we use *scikit-learn*'s implementation which inherently supports multi-output regression. During training this model builds multiple decision trees and consequently outputs the average prediction of all individual trees. Each tree is independently trained on a subset of the data with replacement, which is called bagging and makes trees varied. Random Forests natively provide a measure of feature importance, indicating to what extent features contribute to predictions (Breiman, 2001).

4.4.3 Feed-forward Neural Network

Ultimately, we employ *Keras*, a high-level neural networks API, to implement a multi-output regression FFNN (Chollet et al., 2015). It consists of an input layer, one hidden layer, utilizing Rectified Linear Unit (ReLU) as its activation function, and an output layer. As Figure 8 shows, the input and output layer consist of 99 and 2 neurons respectively. Following (Rocheteau et al., 2021), we used the Mean Squared Log Error (MSLE) loss function to address the problem of skewness. The Adam optimizer is employed to minimize the loss function during training, with a tuned learning rate (Table 3).

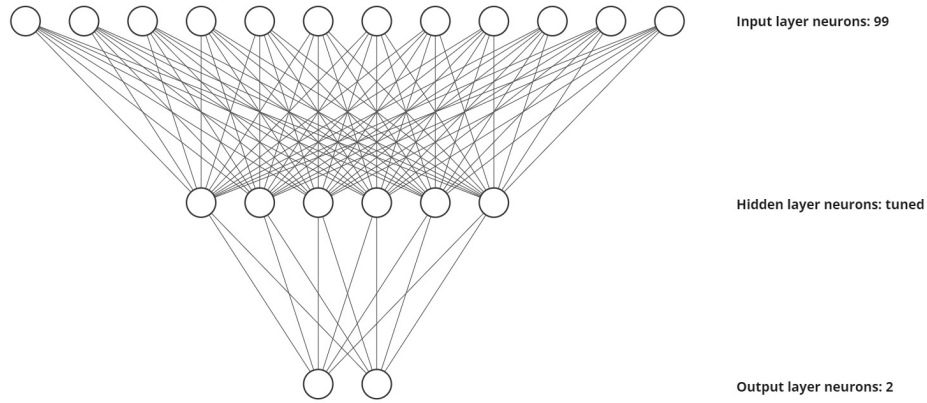


Figure 8: Visual representation of how information flows through our network (from top to bottom). Table 3 shows the hidden layer neurons tuning range. This figure is generated using a visualization tool designed by LeNail (2019).

4.5 Model evaluation

We evaluate the performance of the tuned models and baselines on the test set. This includes the best RF and FFNN with an undersampling approach and without, leading to a total of four models to evaluate excluding baselines. For the out-of-sample evaluation, we use only one of the undersampling approaches based on the former comparison. However, we still evaluate both tuned models to evaluate how unseen interventionists affect their performances. On the censored test and out-of-sample sets, we only evaluate the best tuned model with again the same undersampling approach. For model evaluation, the same transformation steps were applied to predictions as we did for model selection (Section 4.4).

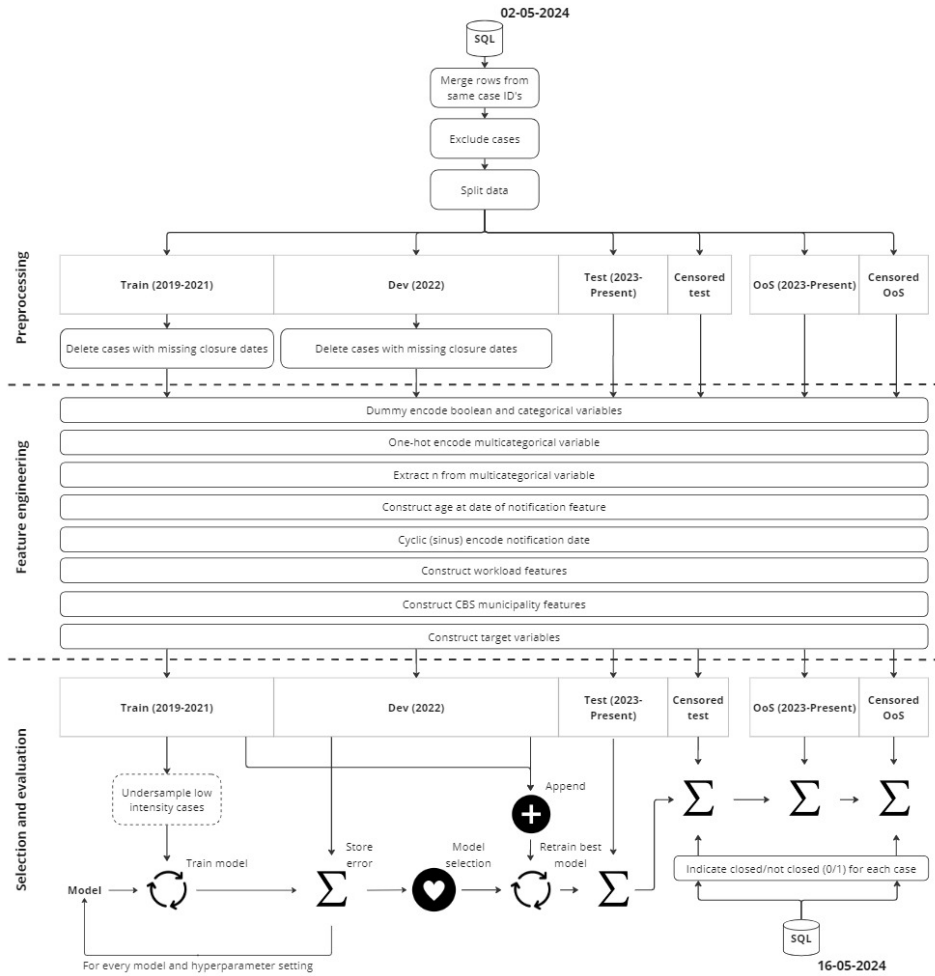


Figure 9: Pipeline flowchart. The application of undersampling depends on the approach. For the censored sets we report evaluation metrics of cases closed between our cut-off date, May 2, 2024, and May 16, 2024.

4.5.1 Performance metrics

Following Rocheteau et al. (2021), we report on Mean Absolute Deviation (MAD), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), Mean Squared Log Error (MSLE) and coefficient of determination (R^2)⁵. Although we prefer readily-interpretable evaluation metrics, we also use relatively little interpretable metrics to detect less meaningful models not performing well across all metrics. A model is particularly meaningful for the AO team if it avoids both substantial overpredictions of low-intensity cases and underestimations for high-intensity cases.

MAPE quantifies the average percentage difference between predicted and actual values. Rocheteau et al. (2021) tailored the MAPE to the task at hand to avoid unbounded percentages for LoS values close to zero. We did not need this modification, as our discrete intensities bound MAPE by definition. It is a good metric for the task at hand as it particularly identifies models overestimating low-intensity cases well (Equation 3), just like MSE does this especially for large errors overall.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{Y_{\text{true},i} - Y_{\text{pred},i}}{\max(Y_{\text{true},i}, 1)} \right| \times 100 \quad (3)$$

The MSLE measures the mean squared difference between the natural logarithm of the predicted and true values, which is also used as a loss function for our feed-forward neural networks. While the metric is suitable for this task, it is less interpretable than the MAD, so we report both.

Although using multiple metrics results in a more holistic approach than using one, it also complicates model selection because metrics might disagree on which model performs best. Therefore, we performed a tailored selection procedure. For each metric and target variable independently, we ranked model performances. The best-performing model is the model whose lowest ranking is higher than all other models' lowest rankings.

4.5.2 Evaluation of censored cases

Evaluating the best-performing models on cases that were censored on our cut-off date presents a challenge. Some cases might have concluded by May 16, 2024 (10 business days later), while others may not. For those concluded by this date, we apply the same evaluation metrics as used for the test and out-of-sample sets. For ongoing cases, we distinguish between

⁵ Rocheteau et al. (2021) reported one more evaluation metric than we did: Cohen's linear weighted Kappa Score. It involves binning data because it is intended for ordered classification tasks rather than regression, something we wish to avoid because defining bins is often arbitrary

predictions surpassing and falling short of the observed intensity up to that point. As an evaluation we report their respective quantities.

4.5.3 Error analysis

As the main objective of estimating case intensity is AO team scheduling, we investigate the effect of case intensity on our models' performances. We group test cases by their discrete true values and calculate the mean error for each group, by subtracting predicted intensities from their corresponding true values.

5 RESULTS

In this chapter, we evaluate our models' performances on the test, out-of-sample and censored sets. The RF and FFNN hyperparameters that led to the best predictions on the development set are presented in Table 4.

Model	Hyperparam.	Train set	
		Regular	Undersampled
RF _{best}	Criterion	Absolute error	Squared error
	Max depth	20	30
	n_estimators	150	150
FFNN _{best}	Neurons	16	32
	Adam learning rate	0.0001	0.0001
	Epochs	100	150

Table 4: Hyperparameter setting of best models per train set

The mean and median baseline predictions always predict the mean and median of their corresponding train set, as shown in Table 5.

Baseline	Regular		Undersampled	
	Days	Interventions	Days	Interventions
Mean	16.3	2.2	19.5	2.5
Median	16.0	2.0	18.0	2.0

Table 5: Mean and median values of combined train and development set. The mean and median baselines predict these for every case in the test and out-of-sample sets.

5.1 Test set evaluation

In this section we first focus on our models' performances with and without an undersampling approach separately, focusing on model comparison. Then, we focus on resampling comparison by presenting performance improvement or deterioration of all models across all metrics. Finally, we show censored test set performance and feature importances.

5.1.1 Undersampling approach

Table 6 shows our baseline models' performances above the dotted line, and the models for model comparison underneath the line. For all metrics except for the R^2 , lower is better.

Our observations on the baseline performances are as follows. Mean and median models perform relatively well on MAD and MAPE, while the MSE, MSLE and R^2 betray them when compared to the Ridge model. This is especially true on the days predictions, as the performance drop is even bigger there. The Ridge model, however, fares worse with the MAD and MAPE than the mean and median predictions. In absolute sense, from the R^2 scores we observe that only the Ridge model reaches a positive score on the interventions target. R^2 scores go as low as -0.41, indicating that no variability in the days target is explained.

For the models to be compared in this study, we observe the following. On the interventions target, our RF outperforms the Ridge baseline on all metrics. It is more alike on the days target. It outperforms the mean and median predictions on the MSE, MSLE and R^2 on both targets, although mean and median predictions are better according to the MAD and MAPE. The FFNN outperforms the RF on all metrics. On the interventions target it also largely outperforms the mean and median predictions, even on MAD and MAPE which have been exploited well by them so far. On the days target it did not reach their performance, although is came close here as well.

Model	Days					Interventions				
	MAD	MAPE	MSE	MSLE	R^2	MAD	MAPE	MSE	MSLE	R^2
Mean	8.98	6.39e14	2087	3.11	-0.41	0.74	2.36e14	5.11	0.25	-0.14
Median	9.06	6.23e14	2091	3.16	-0.41	0.90	2.06e14	5.39	0.28	-0.20
Ridge	13.28	3.08e15	1386	1.49	0.06	1.11	2.92e14	4.69	0.28	-0.05
RF _{best}	13.34	2.73e15	1387	1.47	0.06	1.02	2.42e14	4.24	0.24	0.05
FFNN _{best}	9.37	1.66e15	1379	1.25	0.07	0.76	1.84e14	3.74	0.20	0.17

Table 6: Test set performances with undersampling

5.1.2 Regular approach

Table 7 shows performance metrics for an approach without undersampling. For the baselines, we observe largely the same relative performances for the mean and median models. However, the Ridge models' performance compared to those baselines is worse, especially on the interventions target.

Still, our RF and FFNN outperform all baselines on MSE, MSLE and R^2 , but not on MAD and MAPE. Moreover, The FFNN still performs better than the RF across all metrics.

Model	Days					Interventions				
	MAD	MAPE	MSE	MSLE	R^2	MAD	MAPE	MSE	MSLE	R^2
Mean	9.15	6.04e14	2095	3.22	-0.42	0.83	2.20e14	5.26	0.26	-0.17
Median	9.17	6.00e14	2096	3.23	-0.42	0.90	2.06e14	5.39	0.28	-0.20
Ridge	14.36	3.53e15	1437	1.59	0.03	1.35	3.31e14	5.63	0.34	-0.26
RF _{best}	12.84	2.86e15	1413	1.48	0.05	0.99	2.36e14	4.24	0.24	0.05
FFNN _{best}	10.17	2.14e15	1349	1.28	0.09	0.80	2.13e14	4.07	0.21	0.09

Table 7: Test set performances without undersampling

5.1.3 Comparison of approaches

Table 8 shows the improvement or deterioration in terms of percentages of all models after applying an undersampling approach. In line with the mean and median predictions in Table 5, undersampling did not change median interventions predictions. The other predictions of the mean and median models did not unanimously lead to improvement or deterioration, although we observe a decreasing R^2 score for the interventions target. The latter is also true for the Ridge model, but undersampling led to improved performance according to all its other metrics.

Undersampling improved the RF and FFNN according to 7 and 8 of the 10 metrics respectively. Interestingly, while it led to a decreasing interventions R^2 score for the mean and Ridge models, it substantially improved the FFNN on this aspect.

Model	Days					Interventions				
	MAD	MAPE	MSE	MSLE	R ²	MAD	MAPE	MSE	MSLE	R ²
Mean	1.89	-5.60	0.40	3.54	-2.38	12.16	-6.90	2.94	4.00	-17.65
Median	1.21	-3.78	0.26	2.22	-2.38	0.00	0.00	0.00	0.00	0.00
Ridge	8.13	14.37	3.63	6.71	100.00	21.62	13.20	20.04	21.43	-80.77
RF _{best}	-3.75	4.78	1.87	0.68	20.00	-2.94	-2.34	0.00	0.00	0.00
FFNN _{best}	8.54	28.67	-2.22	2.40	-22.22	5.26	15.52	8.82	5.00	88.89

Table 8: Test set performances **improvement**/**deterioration** after undersampling (%)

5.1.4 Censored test set evaluation

Here, we report performances for an undersampling approach, as it improved performances on the regular test set.

138 test cases were censored on May 2, 2024. On May 16, 2024, 5 of those cases were deleted from the dataset by the AO team. 119 of them were still not closed. 14 were closed in the meantime, for which we report the prediction performance in Table 9.

Model	Days					Interventions				
	MAD	MAPE	MSE	MSLE	R ²	MAD	MAPE	MSE	MSLE	R ²
FFNN _{best}	32.65	2.36	5578	2.48	-0.27	0.65	0.76	9.26	0.36	-0.11

Table 9: Prediction performances on 14 test cases closed between 2 and 16 May, 2024

From the cases that were still not closed, 46 exceed our predictions on both targets already. 9 cases exceed only our intervention predictions and 15 only our days predictions. 49 cases do not exceed our predictions on either target yet.

5.2 Out-of-Sample set evaluation

Compared to the test set our baseline mean and median models predict the out-of-sample days target better. Their MAPE's deteriorated on both targets, however. The Ridge model improves on 4 out of 10 metrics. All three still do not capture any substantial proportion of the variability in either target according to the R².

The RF deteriorates on all metrics but the days MSE, compared to its test performance. Also, it is now outperformed by the Ridge model on all

metrics. The FFNN still outperforms the Ridge model on 4 metrics, yet by smaller margins.

Model	Days					Interventions				
	MAD	MAPE	MSE	MSLE	R ²	MAD	MAPE	MSE	MSLE	R ²
Mean	6.15	1.29e15	1170	2.65	-0.39	0.83	5.32e14	5.97	0.31	-0.13
Median	6.17	1.29e15	1171	2.66	-0.39	0.90	4.99e14	6.09	0.32	-0.16
Ridge	9.98	6.09e15	834	1.60	0.01	0.80	7.04e14	5.16	0.29	0.02
RF _{best}	14.01	1.10e16	841	2.09	0.00	1.15	1.10e15	5.61	0.37	-0.06
FFNN _{best}	8.00	4.82e15	915	1.60	-0.08	0.74	6.96e14	5.55	0.32	-0.05

Table 10: Out-of-Sample set performances with undersampling

5.2.1 Censored Out-of-Sample set evaluation

123 OoS cases were censored on May 2, 2024. On May 16, 2024, one of those case was deleted from the dataset by the AO team. 102 of them were still not closed. 20 were closed in the meantime, for which we report the prediction performance in Table 11.

Model	Days					Interventions				
	MAD	MAPE	MSE	MSLE	R ²	MAD	MAPE	MSE	MSLE	R ²
FFNN _{best}	11.45	0.52	996	0.94	-0.08	0.77	0.52	13.75	0.44	-0.04

Table 11: Prediction performances on 20 censored OoS cases closed between 2 and 16 May, 2024

From the cases that were still not closed, 32 exceed our predictions on both targets already. 5 cases exceed only our intervention predictions and 10 only our days predictions. 55 cases do not exceed our predictions on either target yet.

5.2.2 Feature importances

Analysing Figures 10 and 11⁶, we note that the top 5 features only has one common feature: 'Vermoeden_van_problematiek=Vervuiling'. This translates to pollution being the suspected problem of the case. To the FFNN, the runner up feature is the amount of suspected problems selected by the team. Then, three features indicating the notification source (peer consultation, an interventionists' network and mail/website) follow. To the

⁶ Feature importances are presented for the undersampling approach. Alternatively, appendix 7 shows them for the approach without undersampling.

RF the most important feature is the population density of the corresponding municipality, followed up by case notification features indicating the initial referral advice and the subjects' age.

The RF's feature importances are more skewed than the FFNN's. Case notification features are important to both of them, as 14 and 15 of them are in the top 25 respectively. 5 of the interventionists are within the top 25 for the FFNN, barely 1 for the RF.

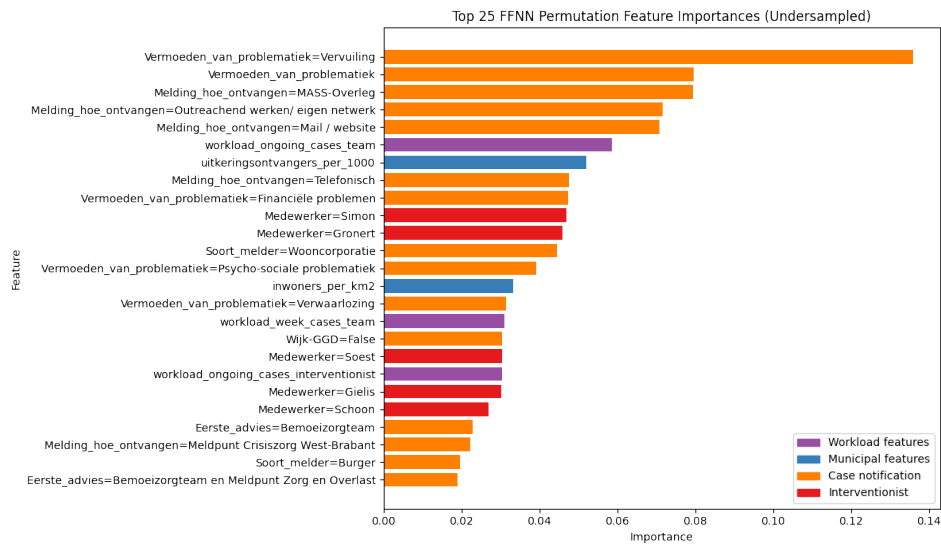


Figure 10: Top 25 most important features to the FFNN in Dutch

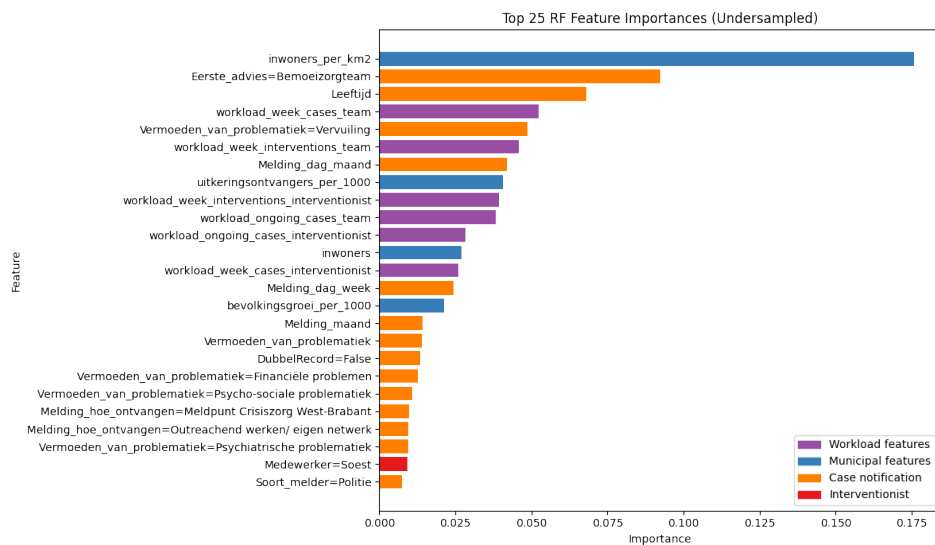


Figure 11: Top 25 most important features to the RF in Dutch

5.3 Errors over case intensity

Moving away from reporting collective performance metrics, we analyze errors over case intensity in Figure 12. Here, a perfect model would display a horizontal line on the x-axis. We observe largely similar error patterns of both models with both approaches. For both targets, our models slightly over-predict low-intensity cases with up to approximately 25 days and 3 interventions. Interestingly, this seems to approximate the values our mean and median baselines predict for every case with an undersampling approach (Table 5).

Cases with intensities above these thresholds are under-predicted. The higher a case's true intensity, the more severe our under-predictions are. For the days target, this relationship is nearly directly proportional. For the interventions target, it is less strong but still severe - e.g. cases with a true intensity of 35 interventions are predicted with a mean error of roughly 30 interventions.

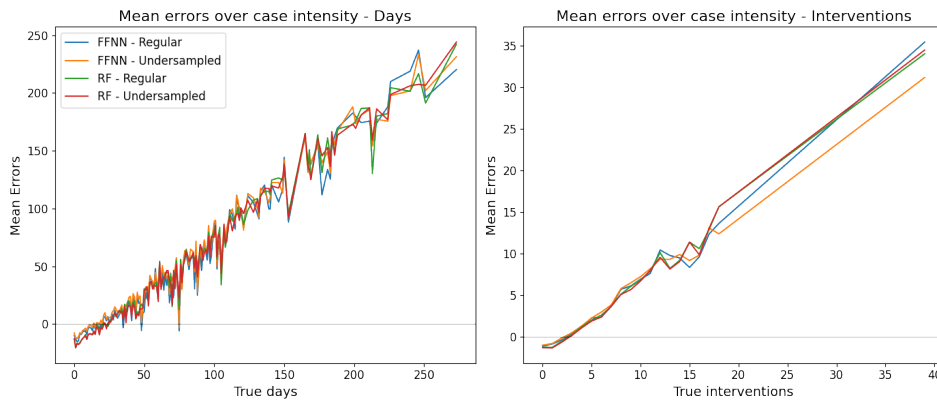


Figure 12: Error analysis: true minus predicted intensities

6 DISCUSSION

The main objective of this study was to predict case intensity from rudimentary case notification details before additional information accumulates through interventions. To this end, we compared the performances of RF and FFNN with and without an undersampling approach. Furthermore, we aimed to determine how well our predictions generalize to unseen interventionists and censored cases. Finally, we analyzed error patterns over case intensity and reported feature importances.

6.1 Results discussion

To answer the central research question we measured the prediction performances of RF and FFNN on the test set with seen interventionists, compared to baseline predictions. We showed that the FFNN was superior to the RF, with and without undersampling. Always predicting the mean or median fares better with the MAD and MAPE than the FFNN and RF, but the other metrics expose that they do not capture the data’s variability. The FFNN interventions predictions compete relatively well with mean and median predictions on their best metrics, compared to the days predictions.

Rocheteau et al. (2021) reported the same performance metrics of their best performing TPC model as well as mean and median baselines, which were 3.47 and 1.67 days for the ‘eICU’ dataset and 5.70 and 2.70 for the ‘MIMIC-IV’ dataset respectively. According to the R^2 scores shown in Table 12, their baselines were unable to explain any variance of the LoS target as well. Due to potentially different patient demographics and disease prevalence, it is challenging to compare their performance with ours (Bacchi et al., 2022).

Dataset	Model	MAD	MAPE	MSE	MSLE	R^2
eICU	Mean	3.21	395.7	29.5	2.87	0.00
	Median	2.76	184.4	32.6	2.15	-0.11
	TPC	$1.78 \pm .02$	63.5 ± 4.3	$21.7 \pm .5$	$0.7 \pm .03$	$0.27 \pm .02$
MIMIC-IV	Mean	5.24	474.9	77.7	2.8	0.00
	Median	4.60	216.8	86.8	2.09	-0.12
	TPC	$2.39 \pm .03$	47.6 ± 1.4	46.3 ± 1.3	$0.39 \pm .02$	$0.40 \pm .02$

Table 12: LoS (days) prediction performances as reported by Rocheteau et al. (2021)

In contrast to what Naemi et al. (2021) found on LoS data, undersampling did not unanimously improve or deteriorate the mean and median

predictions. The values they predict, presented in Table 5, might be influenced too little by the removal of low-intensity cases to observe a substantial difference. This corresponds to a skewness that was still severe after undersampling (Figure 6). However, undersampling led to an improvement of both the RF and FFNN. For the latter this was especially true for the interventions target.

Both the RF and FFNN perform worse on unseen interventionists, while baseline performances remained more stable. This means that predictions for new AO team members will be less reliable. To the AO team this might be an indication of varying interventionists' decisions regarding intervention allocation or registration.

The case statuses on the cut-off date, May 2, 2024, did not change much yet on May 16, 2024. Only 12 cases from the censored test and out-of-sample sets were closed in the meantime, so reported performances are likely sensitive to individual case variance. Getting statuses even later than May 16, 2024 would eventually solve this. Also, relatively little of the censored test and out-of-sample cases exceeded our predictions yet, but no definitive performance can be estimated for them. Still, by showing that this method works conceptually, this could contribute to studies dealing with censoring and having access to a real-time dataset. Splitting data on its temporal order proves advantageous in this regard, as it caused the train set to hold only 4 censored cases out of 1978.

Our error analysis showed that on average, cases with an above mean or median intensity are severely under-predicted by all models. For the days target it is under-predicted by as much as the true case duration in days. Cases below those thresholds are slightly over-predicted. This shows that predictions are less reliable as case intensity increases, which is problematic for the AO team because high intensity cases impact scheduling disparately. It is imperative for the AO team to further refine the model to minimize under-predictions for high-intensity cases before implementation.

Feature importances showed that relatively many case notification variables contribute to predictions compared to workload, municipal and interventionist features. A case's suspected problem being pollution or not helped both the RF and FFNN to predict case intensity. This finding prompts further exploration to understand the specific impact of combined problems. However, prioritizing the model's shortcomings concerning reliability is warranted.

6.2 Limitations

The primary limitation of our study is the anonymous character of its data, as the subjects of cases are being anonymized and therefore can not be

related to possible earlier cases related to the same subject. For that reason we were not able to leverage temporal dependencies in that sense, unlike studies working with EHR data.

Second, the evaluation of our predictions on censored cases in the test and out-of-sample set is not ideal, as even the most up-to-date cases statuses we could get indicated that many censored cases were still not closed yet. This potentially biases our reported results. The removal of censored cases from the train set may have also biased our parameter estimates, but we consider this a minor limitation because this only concerns 13 cases in total.

Finally, this study is limited by varying AO team operational circumstances. Over the years the team might have revised their protocols regarding intervention allocation and motives to close cases. Stable circumstances might lead to better generalizability over time. This is a significant limitation in this study as we also wish to maintain temporal order throughout our analysis, which prevented us from using the data with most recent case notifications to train models. For that reason, the possibility of operational factors being outdated in the train set was relatively high compared to the other subsets.

6.3 *Future research*

First, future research could increase utility of predictions for AO teams by making them generalize better to new cases. This is especially important to solve for seen interventionists' cases, because the AO team could decide to only use predictions for new interventionists once they are represented in the train data enough. Exploring predictive power within free-text notification variables, such as the message accompanying the case notification, is an interesting avenue in this regard.

Then, although we evaluated an undersampling approach our data was still heavily skewed afterwards. This is partly because the models were trained on a merged dataset, of which one part was not undersampled. Future works could evaluate different undersampling ratios to see how much more low-intensity cases can be discarded from the train data.

Features crafted and sourced from external data sources with the help of AO team domain knowledge were among the most important features. This stresses the importance of collaboration between future researchers and AO team members, and leads us to think that adding even more relevant features could improve predictive performance.

Finally, our second limitation also opens up an opportunity for future research. An earlier cut-off date compared to the most recent date we can get case statuses on, would reduce the potential bias in performance

reports. Alternatively, the development of time-to-event models with less assumptions could make them better applicable in AO team case intensity prediction.

7 CONCLUSION

This study contributes to research regarding tasks similar to case intensity and LoS prediction by answering the following research questions:

To what extent can machine learning algorithms predict the intervention intensity of AO cases, measured as the amount of interventions until case closure and days until last intervention, using rudimentary case notification data?

Our study shows that our models trained on rudimentary case notification data perform well on around mean and median true case intensities. However, as mean errors tend to be as big as true case intensities themselves, the model's performance does not meet the AO team's requirements currently. This is especially problematic because of the fact that high case intensities disparately impact AO team scheduling.

1. **To what extent can a RF and a FFNN accurately predict the amount of days and interventions, compared to baseline Ridge, median and mean predictions, through multi-output regression?**

Our model comparison shows that the FFNN performs superior to the RF on the AO team's data. However, it still fails to predict case intensities that deviate from the most dense intensities, around the mean and median, accurately enough.

2. **To what extent do predictions generalize to out-of-sample interventionists' cases?**

Our study shows that predicting case intensity for unseen interventionists leads to a performance drop. This could suggest variability in interventionists' decisions regarding intervention allocation or registration, and emphasises the importance of the train set representing AO team members for whom case intensities are predicted.

3. **What impact does undersampling of low-intensity cases have on predictions?**

The comparison between undersampling approaches shows that although it did not lead to unanimous improvement of all models, it improved the

superior FFNN model. For the interventions target this had the largest effect.

4. What impact does case intensity have on prediction error?

Our error analysis showed a clear relationship between case intensity and mean errors, as the errors tend to get bigger as case intensity goes up. For the days target the mean errors were approximately as high as the true intensity of cases, indicating poor predictive capability.

5. How much do the top 25 features contribute to predictions of case intensity?

Few features carry relatively much importance, as importance distributions were skewed. While importances varied over models, pollution turned out to be a stable important predictor as the suspected problem of a case. Next to case notification features, municipal and workload features were among the most prominent ones.

REFERENCES

- Bacchi, S., Tan, Y., Oakden-Rayner, L., Jannes, J., Kleinig, T., & Koblar, S. (2022). Machine learning in the prediction of medical inpatient length of stay. *Internal medicine journal*, 52(2), 176–185.
- Breiman, L. (2001). Random forests. *Machine learning*, 45, 5–32.
- Brnabic, A., & Hess, L. M. (2021). Systematic literature review of machine learning methods used in the analysis of real-world data for patient-provider decision making. *BMC medical informatics and decision making*, 21(1), 1–19.
- Cai, D., Song, M., Sun, C., Zhang, B., Hong, S., & Li, H. (2022). Deep ordinal neural network for length of stay estimation in the intensive care units. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 3843–3847.
- Chollet, F., et al. (2015). Keras.
- Harutyunyan, H., Khachatrian, H., Kale, D. C., Ver Steeg, G., & Galstyan, A. (2019). Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1), 96.
- Iwase, S., Nakada, T.-a., Shimada, T., Oami, T., Shimazui, T., Takahashi, N., Yamabe, J., Yamao, Y., & Kawakami, E. (2022). Prediction algorithm for icu mortality and length of stay using machine learning. *Scientific reports*, 12(1), 12912.
- Johnson, A. E., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Pollard, T. J., Hao, S., Moody, B., Gow, B., et al. (2023). MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data*, 10(1), 1.
- Lacerda, A., & Pappa, G. L. (2021). Deep thompson sampling for length of stay prediction. *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–7.
- LeNail, A. (2019). Nn-svg: Publication-ready neural network architecture schematics. *J. Open Source Softw.*, 4(33), 747.
- McDonald, G. C. (2009). Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1), 93–100.
- Naemi, A., Schmidt, T., Mansourvar, M., Ebrahimi, A., & Wiil, U. K. (2021). Prediction of length of stay using vital signs at the admission time in emergency departments. In *Innovation in medicine and healthcare: Proceedings of 9th kes-inmed 2021* (pp. 143–153). Springer.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

- Piovani, D., K Nikolopoulos, G., & Bonovas, S. (2021). Pitfalls and perils of survival analysis under incorrect assumptions: The case of covid-19 data. *Biomedica*, 41, 21–28.
- Rahman, M. M., Kundu, D., Suha, S. A., Siddiqi, U. R., & Dey, S. K. (2022). Hospital patients' length of stay prediction: A federated learning approach. *Journal of King Saud University-Computer and Information Sciences*, 34(10), 7874–7884.
- Rocheteau, E., Liò, P., & Hyland, S. (2021). Temporal pointwise convolutional networks for length of stay prediction in the intensive care unit. *Proceedings of the conference on health, inference, and learning*, 58–68.
- Roeg, D., Voogt, M., van Assen, M., & Garretsen, H. (2012). De effecten van bemoeizorg. *De resultaten van een onderzoek bij Nederlandse bemoeizorgteams*, Tranzo: Tilburg University, 7.
- Wen, Y., Rahman, M. F., Zhuang, Y., Pokojovy, M., Xu, H., McCaffrey, P., Vo, A., Walser, E., Moen, S., & Tseng, T.-L. B. (2022). Time-to-event modeling for hospital length of stay prediction for covid-19 patients. *Machine learning with applications*, 9, 100365.
- Zeleeke, A. J., Palumbo, P., Tubertini, P., Miglio, R., & Chiari, L. (2023). Machine learning-based prediction of hospital prolonged length of stay admission at emergency department: A gradient boosting algorithm analysis. *Frontiers in Artificial Intelligence*, 6.

APPENDIX A - FEATURE IMPORTANCES WITHOUT UNDERSAMPLING

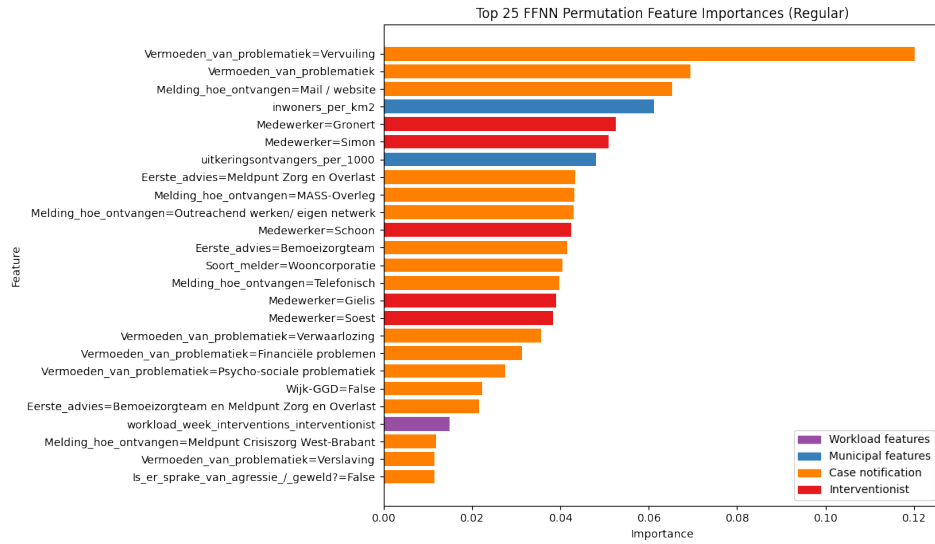


Figure 13: Top 25 most important features to the FFNN - without undersampling

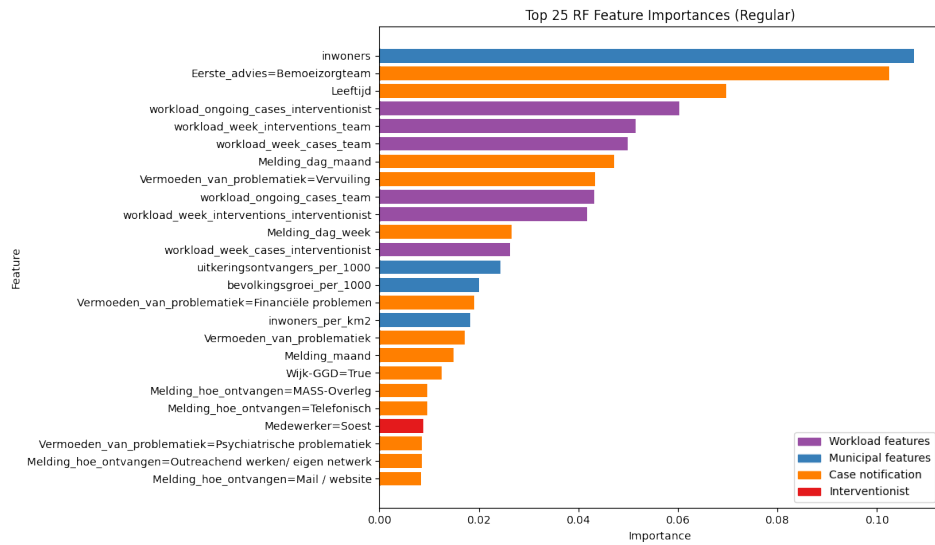


Figure 14: Top 25 most important features to the RF - without undersampling