Classification Approaches for Data With Class Imbalance and Biased Labelled Sample

Alexandre de Pinho Uliana

Master's Thesis Psychology,

Methodology and Statistics Unit, Institute of Psychology

Faculty of Social and Behavioral Sciences, Leiden University

Date: 1st of August, 2024

Student number: s3987639

Supervisor: S.M.H. Huisman Ph.D (internal), X.van de Putte (external)

Abstract

This thesis investigates the efficacy of various classification methods in handling data with class imbalance and biased labelled samples, with a specific focus on applications relevant to the Netherlands Labour Authority (NLA). The study explores semi-supervised methods including label propagation, label spreading, and self-learning, in addition to supervised and the Jacobusse & Veenman methods. The performance of these methods is evaluated through simulations representing different scenarios of class imbalance and labelling bias. Key performance metrics such as Precision@100 and algorithmic bias are used to determine the most effective approach under varying conditions. Findings indicate that while supervised methods are adequate for balanced datasets with no labelling bias, semi-supervised methods exhibit superior performance in scenarios characterized by class imbalance and labelling bias, despite their higher algorithmic bias. The study highlights the importance of choosing methods tailored to specific data characteristics and analysis objectives. Moreover, the research stresses the need for future studies to use larger and more diverse datasets, incorporating additional feature types and exploring different levels of class imbalance and labelling bias. This will enhance the generalizability and practical applicability of the results, particularly in the context of large-scale datasets such as those managed by the NLA. The implications of this research extend to improving predictive modeling practices, ensuring better allocation of inspection resources, and promoting fair, healthy, and safe working conditions across various sectors.

Keywords: algorithmic bias, class imbalance, machine learning, labelling bias, semi-supervised learning, supervised learning, predictive modeling, mean reciprocal rank (MRR), pseudo-labelling, random forest classifier, simulation-based approach, data imbalance, model performance, precision@100, label propagation, label spreading, Netherlands Labour Authority, algorithmic bias.

Classification Approaches for Data With Class Imbalance and Biased Labelled Sample

The Netherlands Labour Authority (*Nederlandse Arbeidsinspectie*, or NLA), part of the Ministry of Social Affairs and Employment (*het ministerie van Sociale Zaken en Werkgelegenheid*), is dedicated to ensuring fair, healthy, and safe working conditions, as well as socio-economic security for every worker in the Netherlands (Sociale Zaken en Werkgelegenheid, 2022). The NLA also investigates cases of fraud, exploitation, and organized crime in the labour market. Much of this work is done through close cooperation with employers across different sectors, involving inspections to monitor compliance with labour laws and regulations. However, with nearly 9.8 million employed people in the Netherlands, keeping up with every violation is not an easy task for the NLA's team of only about 1,800 workers (Centraal Bureau voor de Statistiek, 2023; Ministerie van Sociale Zaken en Werkgelegenheid, 2024). To address this challenge, the NLA uses its team's expertise to focus on high-risk segments that require closer attention. Therefore, the NLA selects cases for targeted interventions through direct contact (Cluster 1), indirect influence via other organizations (Cluster 2), and communication strategies (Cluster 3), reaching about 13.9% of companies in Clusters 1 and 2 and finding violations in 23 to 50% of the cases, depending on the labour law. Recently, Machine learning has also been incorporated as a tool to interpret the vast amount of data on labour relations in the Netherlands and extract useful information about employers with a high chance of maintaining unfair, unhealthy, or unsafe working conditions.

Machine learning has become a popular tool in regulatory practices. Crime forecasting tools, which analyze historical data to predict future incidents, have been implemented worldwide, enabling inspectorates to allocate resources more efficiently and effectively (Shah et al., 2021). For example, Germany's *Precobs* system has been applied to forecast residential burglaries, while the Metropolitan Police Service in London has explored predictive models to manage resources more efficiently (Mugari & Obioha, 2021).

The NLA is experimenting with similar machine learning techniques to gain insight into work conditions in the Netherlands. It is developing pilots that use available data to identify patterns and associations between employer characteristics and a higher chance of labour law violations. By doing so, the NLA aims to organize its inspections more effectively, focusing its limited resources on high-risk situations, thereby improving overall compliance with labour laws.

To classify companies as compliers or violators based on their characteristics, supervised machine learning is the most straightforward technique. It operates on the premise that specific features of companies—such as their industry or number of employees—can be indicative of their compliance status. By analysing these features, it becomes possible to predict the likelihood of a company being a violator or complier. Supervised learning algorithms, such as decision trees, support vector machines, and neural networks, are widely used in various domains for classification tasks due to their ability to process large datasets and identify complex patterns (James et al., 2013). These algorithms can be trained on labelled data where the compliance status (complier or violator) is known to develop predictive models. Once trained, these models can be applied to new unlabelled data to predict the compliance status of other companies, thereby enhancing the efficiency and effectiveness of resource allocation for inspections.

In the case of the NLA, labels are known through inspections, where inspectors assess compliance by visiting workplaces to review documents, observe working conditions, and interview employees and management. Findings are documented, feedback is provided to the employer, and corrective actions are recommended, resulting or not in a fine. This systematic approach ensures accurate labelling of companies as compliant or non-compliant. However, these inspections cover only a fraction of the employers in the Netherlands and are not done on a random sample of the population. If the samples were randomly selected and adequately distributed across various sectors and if the number of samples were large enough, the data could likely be used successfully in supervised learning

methods to predict risks of law violations. But this is not the case because inspections are strategically targeted at segments with a higher probability of problematic labour relations, according to the NLA's expertise. This **labelling bias** results in a sample that does not represent the entire population of employers, leading to skewed datasets and causing supervised models to under-perform when applied broadly (Mehrabi et al., 2021). Therefore, despite having data on features from nearly all employers in the Netherlands, only these few select cases are labelled and suitable for inclusion in supervised learning methods by the NLA. Furthermore, societal biases can influence which groups are targeted for inspection, potentially resulting in models that unfairly prioritize certain segments and overlook others (Bruggeman et al., 2023). So, if any bias is present in the labelled sample, such as the disproportionate targeting of specific industries, it could be perpetuated by the supervised machine learning algorithms (Lum & Isaac, 2016).

In addition, the **class imbalance** within the population can represent a challenge in the use of prediction models. In this case, the target class, or the violators, represents only a small fraction of employers. This imbalance poses several challenges in effectively utilizing machine learning for law enforcement and compliance tasks. Firstly, class imbalance can lead to biased models that prioritize the majority class (compliers) over the minority class (violators). Without precautions, supervised algorithms trained on imbalanced data tend to achieve high accuracy on the majority class while performing poorly in detecting violators (Jacobusse & Veenman, 2016). Secondly, the scarcity of violator instances makes it difficult for machine learning algorithms to learn meaningful patterns that distinguish violators from compliers. As a result, these algorithms may struggle to generalize and accurately classify unseen cases of non-compliance.

Addressing both class imbalance and labelling bias requires innovative strategies that can better capture the complexities of this type of data (Van Giffen et al., 2022). Techniques such as oversampling the minority class (violators) or under-sampling the majority class (compliers) can be used to balance the dataset (Chawla et al., 2002).

Additionally, cost-sensitive learning, where misclassification costs are incorporated into the learning process, can help create models that pay more attention to the minority class (Elkan, 2001). Ensemble methods, such as boosting, can also be effective in dealing with class imbalance by combining multiple weak classifiers to create a stronger model (Freund & Schapire, 1997). Another approach involves using propensity scores to balance the distribution of features between different groups, ensuring fair comparisons. Propensity scores estimate the effect of a treatment by accounting for covariates that predict receiving the actual treatment (Williamson & Forbes, 2014). They are calculated using a model to predict the likelihood of an instance being in the treatment group (in our case, the labelled sample), given its features. These scores are then used to match or weight instances, helping to mitigate labelling bias and leading to more reliable conclusions in the analysis of compliance and non-compliance among employers.

Semi-supervised methods

In scenarios where there is a large amount of data about the features but only a few labelled cases like what happens in the NLA, semi-supervised learning methods can be particularly useful. Semi-supervised methods use the large amount of unlabelled data to inform the learning process and create better predictive models even with only a small amount of labelled data (Olivier. Chapelle et al., 2006). Techniques such as self-training, label spreading and label propagation can be applied to enhance model performance in such contexts. These methods could help the NLA to make the most of the data at its disposal to better predict and address labour law violations.

One simplified semi-supervised method used by the NLA is commonly known as the **"Jacobusse & Veenman Method"** (Jacobusse & Veenman, 2016). This technique addresses class imbalance and labelling bias by incorporating unlabelled cases into the training process, helping to balance the training dataset. It assumes that most of the unlabelled cases are non-violators, effectively capturing the characteristics of cases not

reached by inspections and balancing out the trade-off of potentially mislabelling some violators as compliers. Additionally, since inspectors often target specific segments, even compliers in these inspections share many traits with violators. So, the "Jacobusse & Veenman Method" improves the model's predictions by excluding true compliers from the training set. However, while this approach can enhance model performance by utilizing unlabelled data, it relies on the assumption of a high class imbalance. If this assumption is incorrect, treating all unlabelled cases as compliers can introduce significant noise into the model, potentially impairing its accuracy. Nevertheless, the "Jacobusse & Veenman Method" remains a practical and straightforward solution for creating more balanced training datasets in the presence of severe class imbalance.

Beyond the "Jacobusse & Veenman Method", other semi-supervised learning techniques offer more dynamic approaches to leveraging both labelled and unlabelled data (Oliveira & Berton, 2023). One such technique is **self-learning** (or self-training), which begins with a base classifier trained on the labelled data and then iteratively labels a subset of unlabelled data based on the classifier's predictions. This approach refines the model by incorporating these newly labelled instances into the training set (Rizve et al., 2021; Triguero et al., 2015). While self-learning can improve performance, it is sensitive to biases in the initial labelled dataset and can amplify errors if the base classifier is poorly calibrated.

Another semi-supervised technique is **label propagation**, which assigns pseudo-labels through a similarity graph constructed from both labelled and unlabelled data. Labels are propagated from known to unknown data points, typically using a fixed similarity matrix and "clamping" mechanisms to manage label confidence (Bengio et al., 2006; Iscen et al., 2019). Label propagation is effective in situations with class imbalance but can potentially amplify existing biases.

Label spreading also uses a similarity graph but differs in its approach to label

distribution. It uses the normalized graph Laplacian to spread the labels, making the method more robust to noise and outliers compared to label propagation (Zhou et al., 2003). However, it also has the potential to propagate biases if the similarity relationships reflect the inherent imbalances in the data.

Although semi-supervised methods can enhance predictive performance, they also have the potential to exacerbate algorithmic bias if not carefully managed. These methods often rely on unlabelled data, which, if unrepresentative or biased, can distort the learning process. Algorithmic bias refers to systematic and unfair deviations in predictions that arise from skewed training data, flawed assumptions, or limitations of the algorithms themselves, potentially leading to unequal model performance across different demographic groups and resulting in discriminatory outcomes (Baker & Hawn, 2022). For example, models trained on disproportionately represented data might unfairly target or neglect certain groups, reinforcing existing inequalities. Therefore, it is necessary to monitor and address these biases to ensure that semi-supervised learning methods contribute to fair and equitable outcomes, rather than perpetuating or amplifying existing disparities.

Research question

In the context of the NLA, there is a significant need to systematically compare methods for addressing samples with labelling bias, using the extensive amount of unlabelled data available. This exploratory study will compare the "Jacobusse & Veenman Method", self-learning, label spreading, label propagation, and a supervised learning method. Additionally, a post-hoc self-learning technique based on the "Jacobusse & Veenman Method" will be tested and discussed. The goal is to address the issues of class imbalance and labelling bias in the use of machine learning in the NLA. Methodologically, it seeks to compare these techniques, aiming to improve the accuracy of predictive models, potentially leading to more targeted and efficient allocation of inspection resources, thereby fostering safer and more equitable working conditions.

Research Question 1: How does the "Jacobusse & Veenman Method" compare to the supervised learning method in terms of precision and algorithmic bias in scenarios with class imbalance and labelling bias?

- **Hypothesis 1**: The "Jacobusse & Veenman Method" will have higher precision and lower algorithmic bias than the supervised learning method in these scenarios.

Research Question 2: Do semi-supervised methods outperform the supervised learning method in scenarios with class imbalance and labelling bias?

- **Hypothesis 2**: The semi-supervised methods will have better precision levels than the "Jacobusse & Veenman Method" and supervised learning method in these scenarios.

Methods

This study uses simulations to test 6 machine learning methods under 6 scenarios: 2 levels of class imbalance and 3 levels of labelling bias. All simulations and models in this study were conducted using Python (Van Rossum & Drake Jr, 1995). The complete code can be found in Appendix B.

Simulating a population

First, populations with different proportions of each class are created (see Figure 1 for an illustration). A matrix X with predictor variables (features) and a binary outcome (class labels) is generated (see details in Appendix B). In summary, 20 features are used: 18 follow a normal distribution, with random means between 0 and 1 and random standard deviations between 0.8 and 1.1; and 2 are multimodal, created by mixing different normal distributions. The coefficients for each feature are random values uniformly distributed

between 0 and 1. The class labels 0 (compliers) and 1 (violators) are defined using a logistic function:

$$\eta_i = b_0 + \sum_{j=1}^{20} b_j x_{ij} + e_i,$$

where:

- η_i *is the linear predictor for the i-th observation;*
- b_0 *is the intercept term;*
- b_j *are the coefficients for the j-th feature* x_{ij}*;*
- x_{ij} *is the value of the j-th feature for the i-th observation in Matrix* X*; and*
- e_i *is the error term, following a normal distribution* $N(\mu = 0, \sigma^2 = 2)$*.*

The probability p_i of being class 1 (violator) is modelled using the logistic link function:

$$p_i = logit(\eta_i) = \frac{1}{1 + e^{-\eta_i}},$$

where:

- p_i *is the probability of the i-th observation belonging to class 1 (violator) instead of class 0 (complier); and*
- $logit(\eta_i)$ *transforms the linear predictor* η_i *into the probability* p_i *using the logistic function.*

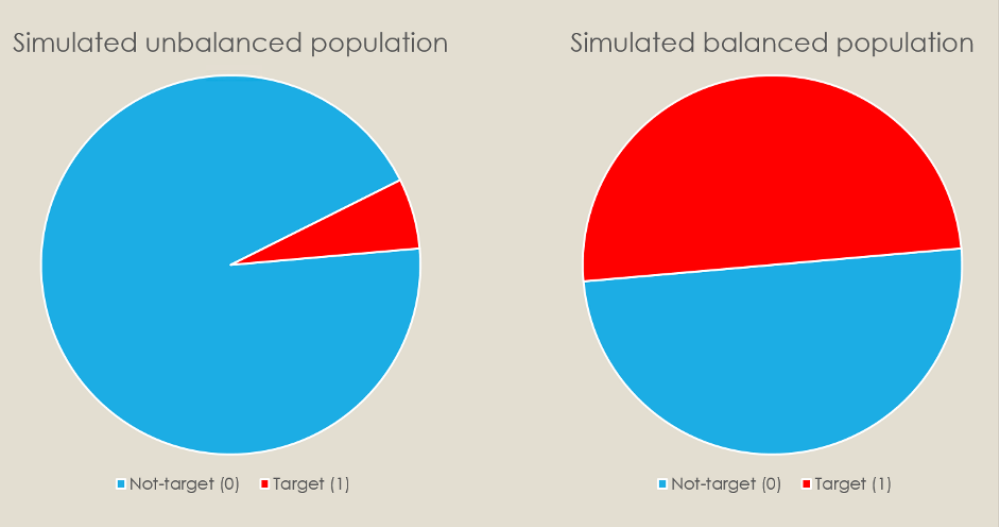

Figure 1. Manipulation of class Proportions in the population.

The intercept is adjusted to set the class proportions while keeping the classification threshold fixed at 0.5. If $p_i \geq 0.5$, the observation is class 1; otherwise, it is class 0. Two levels of class proportions are simulated: equal distribution and 95% class 0. For this study, 20,000 cases are simulated, divided into two populations of equal size and class proportions. The first, referred to as the "current population" in this study, is used as the main population from which a labelled sample is extracted. The second is reserved as a test set.

Selection of the labelled sample

Five percent of the current population, amounting to 500 cases, are marked as labelled and retain their true target values. The remaining 95% (9,500 cases) are marked as unlabelled, with their true labels removed. The labelled sample always includes at least 20% (100) randomly selected cases, reflecting the ideal conditions of the NLA, where approximately 20% of inspections would be random.

The degree of labelling bias is manipulated to simulate different scenarios. In no-bias scenarios, all 500 labelled samples are randomly chosen. With labelling bias, the labelled samples are selected based on a set class proportion. To achieve this, the current

population is sorted based on a linear combination of the two most informative features (see Appendix B for details on how this combination is computed). The labelled cases are sampled from the top of this sorted list until the desired class proportion is reached. This approach aims to mimic the varying levels of focus of inspections on categories of employers with higher rates of infractions.

Three levels of labelling bias are tested: no bias (Figure 2), moderate bias (Figure 3), and extreme bias (Figure 4). An example of the distribution of cases per class in each scenario can be seen in Table 1.

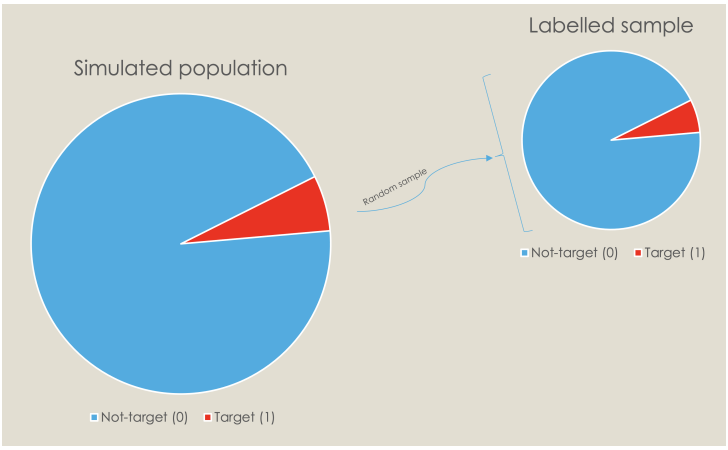

Figure 2. Unbiased labelled sample

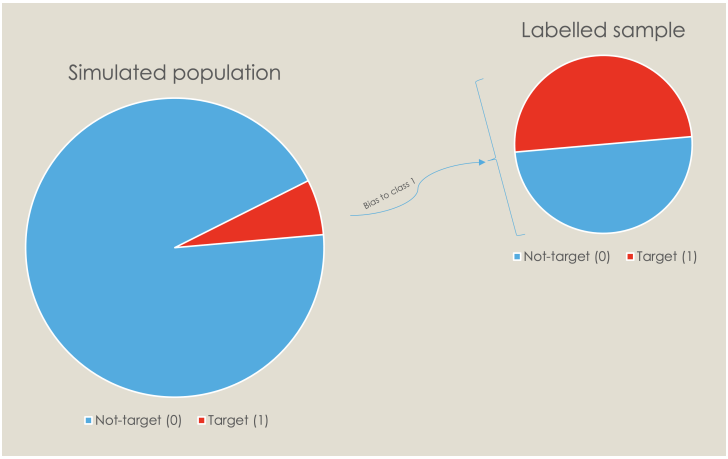

Figure 3. Sample with moderate labelling bias (50% to class 1).

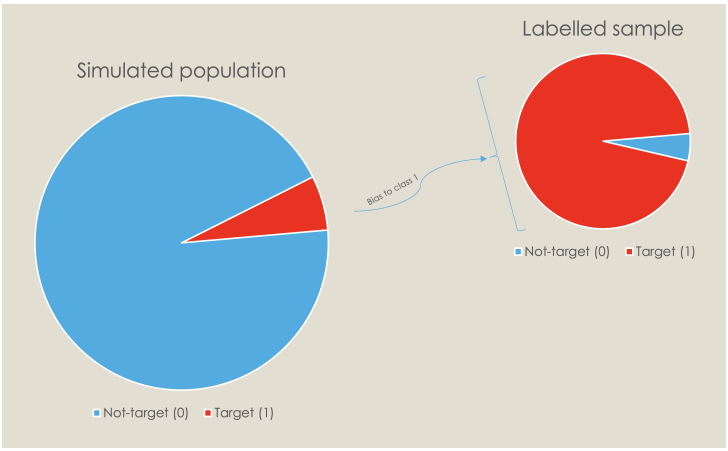

Figure 4. Sample with extreme labelling bias (95% to class 1).

Table 1

Sample of Simulated Composition for Each Scenario - random state 42

Scenario	Class 0 (random)	Class 1 (random)	Class 0 (biased)	Class 1 (biased)
Balanced population, No labelling bias	250	250	0	0
Balanced population, Moderate labelling bias	47	53	203	197
Balanced population, Extreme labelling bias	47	53	0	422
Unbalanced population, No labelling bias	475	25	0	0
Unbalanced population, Moderate labelling bias	99	1	151	249
Unbalanced population, Extreme labelling bias	99	1	0	474

Note. A slight variation in the sample composition can occur between simulations due to random chance, depending on how many of the 20% random cases were selected for each class. In situations of extreme labelling bias and more than 25 cases randomly selected for class 0, like in this example, the labelled sample can exceed 500 cases to achieve a higher proportion of class 1 cases.

Classification methods

For each combination of class imbalance and labelling bias (six scenarios in total), six different approaches are tested on the same test set. Random forest was chosen to be the main classifier for all methods, except label propagation and label spreading, which have their own algorithms based on data similarities. This classifier was chosen because it is commonly used in the NLA, for its ease of interpretation and flexibility even in complex data structures. Additionally, random forest classifiers are robust to over-fitting, especially with a large number of trees.

These are the approaches that are tested:

- **Supervised Method**: A random forest classifier is trained using only the labelled data.
- **"Jacobusse & Veenman Method"** (Jacobusse & Veenman, 2016): A random forest classifier is trained using labelled and unlabelled data, as follows:

    1. The labelled data of class 1 is kept in the training set because class 1 data (violators) are likely to be the target class of interest and the most informative for the classification task. Keeping this data ensures the model can learn the characteristics of this class accurately.
    2. The labelled data of class 0 is removed because the "Jacobusse & Veenman Method" follows the assumption that the labelling process was biased towards class 1. Therefore, labelled cases of class 0 are too similar to those of class 1, meaning that removing them can help to differentiate the classes.
    3. The unlabelled data is included in the training set with a pseudo-label '0'. This approach assumes that unlabelled instances are more likely to be compliers (class 0), helping to balance the training data and providing more examples for

the model to learn the characteristics of class 0 without explicitly using the potentially biased labelled data.

- **"Revised Jacobusse & Veenman Method"**: This ad hoc technique, based on the "Jacobusse & Veenman Method", uses a semi-supervised logic to create an hybrid training set, following these steps:

    1. The unlabelled data is divided into 5 folds.
    2. In 5 rounds, 4 folds are joined with the labelled data and used to predict the class probability of the unlabelled cases in the remaining fold using the "Jacobusse & Veenman Method". For each round, a different training set is created, where labelled data of class 1 is kept, labelled data of class 0 is removed, and unlabelled data is included with a pseudo-label '0'.
    3. After the 5 rounds, a final training sample is created with labelled cases of class 1, unlabelled cases with a probability of over 50% of being class 1, and the remaining unlabelled cases with a pseudo-label '0'. Similar to the "Jacobusse & Veenman Method", the labelled cases of class 0 are excluded from the training set.
    4. A random forest classifier is then trained on this final sample.

- **Self-learning**: The algorithm starts with a small labelled dataset and iteratively expands it by labelling unlablelled data points that it is confident about (O. Chapelle et al., 2009; Oliveira & Berton, 2023). If follows these steps:

    1. Starts with a smaller dataset $D_l = (x_i, y_i)_{i=1}^{n_l}$ where x_i are features of the labelled cases, and y_i are their corresponding labels. Also, has a larger dataset of unlabelled cases $D_u = (x_j)_{j=1}^{n_u}$, where x_j are the features of the unlabelled cases.
    2. A random forest classifier is trained on D_l.
    3. Labels are predicted for D_u using the trained model.

4. Data points are selected from D_u for which the model's prediction confidence exceeds a threshold confidence.
5. These confidently predicted cases are added to D_l with their predicted pseudo-labels.
6. Iterates steps 2 to 5 until either a maximum number of iterations is reached, or until no new pseudo-labels are added, or all unlabelled samples have been labelled.

In this study, this method is operationalised using Scikit-learn's `SelfTrainingClassifier` (*SelfTrainingClassifier*, 2024).

- **Label Propagation**: This technique propagates labels across similar data points on a multidimensional space (Iscen et al., 2019; Zhu & Ghahramanih, 2002). In summary, each point receives information about the label of its neighbours while retaining its initial information. In the end, each unlabelled point is assigned to the class that has provided the most information throughout the iteration process. The algorithm follows these steps (Bengio et al., 2006):

1. An affinity matrix W is constructed to represent the similarity between data points, which are computed using the radial basis function with a hyper-parameter γ controlling how far the influence of a single data point reaches.
2. A diagonal degree matrix D is constructed where each diagonal entry represents the sum of similarities for each data point.
3. A label vector $\hat{Y}^{(0)}$ is initialised with the initial labels or -1 for unlabelled cases.
4. Pseudo-labels for the unlabelled cases are assigned iteratively, so that $\hat{Y}^{(t+1)} \leftarrow D^{-1}W\hat{Y}^{(t)}$ until convergence to $\hat{Y}^{(\infty)}$.
5. The final label of each data point is defined by the sign of $\hat{y}_i^{(\infty)}$.

In this study, this method is operationalised using Scikit-learn's `LabelPropagation` (*LabelPropagation*, 2024).

- **Label Spreading**: This technique also spreads labels across closest data points. However, it normalises the edge weights by computing a graph Laplacian matrix (Bengio et al., 2006; Zhou et al., 2003). As a result, it ensures that labels change gradually and smoothly between nearby points, so that neighboring points in the graph do not have drastically different labels. It follows these steps:

  1. An affinity matrix W is constructed to represent the similarity between data points, which are computed using the radial basis function with a hyper-parameter γ controlling how far the influence of a single data point reaches.
  2. A diagonal degree matrix D is constructed where each diagonal entry represents the sum of similarities for each data point.
  3. The normalized graph Laplacian $\mathcal{L} \leftarrow D^{-1/2}WD^{-1/2}$ is computed.
  4. A label vector $\hat{Y}^{(0)}$ is initialised with the initial labels or -1 for unlabelled cases.
  5. A parameter for α (influence of initial labels on propagated labels) is set.
  6. Pseudo-labels are assigned iteratively, so that $\hat{Y}^{(t+1)} \leftarrow \alpha\mathcal{L}\hat{Y}^{(t)} + (1-\alpha)\hat{Y}^{(0)}$ until convergence to $\hat{Y}^{(\infty)}$.
  7. The final label of each data point is defined by the sign of $\hat{y}_i^{(\infty)}$.

  In this study, this method is operationalised using Scikit-learn's Scikit-learn's `LabelSpreading` (*LabelSpreading*, 2024).

Performance Measures

Since the goal is to optimise inspector efficiency by targeting inspections with a higher likelihood of uncovering violations, precision is the primary performance metric. Therefore, precision at top-100 ("precision@100") is chosen to compare the performance of

different approaches on the same scenario. This metric evaluates the precision for the top 100 instances ranked with the highest probability of belonging to class 1. In the context of inspector allocation, this metric is particularly relevant as it indicates the model's effectiveness in prioritizing the most likely cases with violations.

To complement precision at top-100, algorithmic bias serves as an additional performance measure. This is a broad concept with varying definitions, often encompassing different aspects of fairness and discrimination in automated systems (Baker & Hawn, 2022). In this study, algorithmic bias is objectively measured by assessing how closely the top 100 predicted violators align with actual violators in the test set. This measure utilizes a centroid distance comparison, calculated by determining the Euclidean distance between centroids of the true class 1 instances in the test set and the top 100 predicted class 1 instances.

These combined measures provide a comprehensive evaluation of the model's performance, balancing accuracy with fairness in the prediction outcomes, which is particularly critical in semi-supervised models, which tend to accentuate existing biases in the labelled sample.

Hyperparameters tuning

For each machine learning approach, a set of hyper-parameters must be optimised to best fit the model to the characteristics of the data. During hyper-parameters optimisation, **Mean Reciprocal Rank (MRR)** is chosen as a scorer for model comparison. This method optimises a model to generate an efficient rank of cases with the highest probability of belonging to class 1, aligning well with the primary metric of precision@100 (Efimov, 2023). A direct optimisation of the precision@100 is not always possible for hyper-parameters tuning because cross-validation is used, resulting in many cases in folds with only a couple hundreds cases or less. Especially with class imbalance in

the population, this can represent only very few true cases of class 1 per fold. So, the measurements of precision@100 on these cases would be distorted and not always indicative of a good performance in the test set. MRR, on the other hand, provides a more stable and reliable measure, irrespective of the absolute number of cases in each fold. Essentially, MRR provides a single measure of ranking quality, where higher values indicate better performance, particularly emphasizing the importance of top-ranked items.

To calculate MRR, predicted cases are sorted by the probability of belonging to class 1. The following formula is then applied:

$$RR_q = \frac{1}{1 + \zeta \times rank_q},$$
$$MRR = \frac{1}{|Q|} \sum_{q \in Q} RR_q.$$

Here, RR_q (Reciprocal Rank) is calculated for each relevant item q in the list. The rank, in this case, means how many true instances of class 1 were found up to that point in the ranking of highest prediction probabilities of belonging to class 1. A decay factor ζ of 0.5 is also included to gradually reduce the weight of higher-rank cases, increasing the contribution of lower-ranked cases to the score. For example, if the three instances with the highest predicted probabilities of belonging to class 1 actually belong to classes 1, 0, 1 respectively, then the RR of the two relevant items are calculated as follows: $RR_1 = \frac{1}{1+0.5\times1} = 0.66\overline{6}$ and $RR_3 = \frac{1}{1+0.5\times2} = 0.5$. The Mean Reciprocal Rank (MRR) is the average of the reciprocal ranks of all items up to the position Q. So, in our example, $MRR = \frac{2/3+1/2}{3} = 0.38\overline{8}$. This formula is adapted to only account for the top-100 cases ($Q = 100$), aligning with precision@100 as the main evaluation metric.

In this research, the following hyper-parameters of the random forest classifier were optimised for the best MRR:

- the *number of trees* in the forest, ranging from 10 to 200, which determines the

number of trees in the forest;

- the *percentage of features* to consider when looking for the best split, ranging from 0.1 to 0.725, which controls the number of features to consider in each tree;
- the *maximum depth* of the tree, ranging from 2 to 20, which helps prevent over-fitting by limiting the tree's growth;
- the *minimum number of samples required to split* an internal node, ranging from 2 to 10, which specifies the minimum number of samples required to split an internal node and helps control over-fitting;
- and the *minimum number of samples required to be at a leaf node*, ranging from 1 to 10, which defines the minimum number of samples required to be at a leaf node and also helps in preventing over-fitting.

Semi-supervised methods are particularly sensitive to their hyper-parameters, as their performance heavily depends on the correct identification and propagation of pseudo-labels, which can be significantly affected by sub-optimal settings. For the self-learning method, beyond the random forest hyper-parameters, the following were also optimised:

- *Criterion*: Determines how pseudo-labels are selected: either probability threshold or the labels of the neighbouring cases.
- *Probability threshold*: The probability threshold for the assignment of the pseudo-label, if using the threshold criterion (ranging from 0.1 to 1 in increments of 0.1).
- *Number of neighbours*: The number of neighbours (3, 5, or 10) consulted to predict a pseudo-label, if using the criterion of similarity with the neighbouring cases.

For label propagation and label spreading, the following hyper-parameters were optimised:

- *Gamma (γ)*: A parameter for the RBF (Radial Basis Function) kernel that defines the influence range of a single training example (0.1, 1, 10, 20, 30). Higher values make the influence range smaller.
- *Alpha (α)*: Specifically for the label spreading method, a clamping factor that controls the influence of the true labels (0.1, 0.2, 0.3, 0.5). Higher values mean more reliance on the true labels during spreading.

The primary hyper-parameter optimisation techniques used in this research are genetic algorithms. These search heuristics mimic natural selection to identify optimal solutions (Brownlee, 2011). They operate on a population of potential solutions, or genes, applying selection, crossover, and mutation to evolve the solutions over several generations, aiming to find the best set of hyper-parameters that maximize a fitness function. To optimise the hyper-parameters of the random forest classifiers, a genetic algorithm was implemented using `GASearchCV` (*GASearchCV — sklearn genetic opt 0.10.1 documentation*, 2023). Populations of 50 sets of hyper-parameters were tested using 5-fold cross-validation over a maximum of 40 generations. The algorithm was set to stop if there was no change in the average MRR after 2 generations, indicating population convergence. Additionally, a mutation probability of 10% was used, meaning each gene had a 10% chance of random alteration, maintaining diversity and avoiding local optima. These hyper-parameters were optimised separately for each application of the random forest classifier. For instance, in the "Revised Jacobusse & Veenman Method", optimisation was conducted for the models that calculated classes probabilities before assigning pseudo-labels, and one more time for the final model evaluation.

For optimising the label spreading method, the data must indicate unlabelled cases with a pseudo-class -1. `GASearchCV` cannot be used as it interprets -1 as a new class. Therefore, `TPOTClassifier` (Olson, 2024) was used to apply a genetic search of the best hyper-parameter set. This method also uses 5-fold cross-validation to test populations of

20 hyper-parameter sets per generation, stopping upon convergence to the optimal set.

For Self-Learning and Label Spreading, a full grid search was employed, given the few parameters involved.

Analysis

In this study, the statistical approach uses simulation-based evaluations rather than conventional statistical tests. The simulations allow for controlled experimentation across the various scenarios of class imbalance and labelling bias. Unlike empirical studies, simulations generate data under specified conditions, bypassing the need for assumptions checking typically associated with empirical data. The adequacy of this approach lies in its ability to systematically vary conditions, mimicking real-world scenarios of interest to the NLA.

As previously mentioned, each simulation contains 6 scenarios, including 2 levels of class imbalance and 3 levels of labelling bias. In each scenario, 6 different machine learning techniques are applied, and their performance is measured. To account for the inherent variability in simulation-based studies, the entire simulation process is repeated 42 times with different random states. Throughout these iterations, performance metrics are calculated and saved for each scenario and each method. The results are then averaged across all seeds, resulting in informative tables and plots about the performance of each model in every scenario.

Visualising the simulations

Figure 5 displays examples of simulated populations with unbalanced and balanced classes.

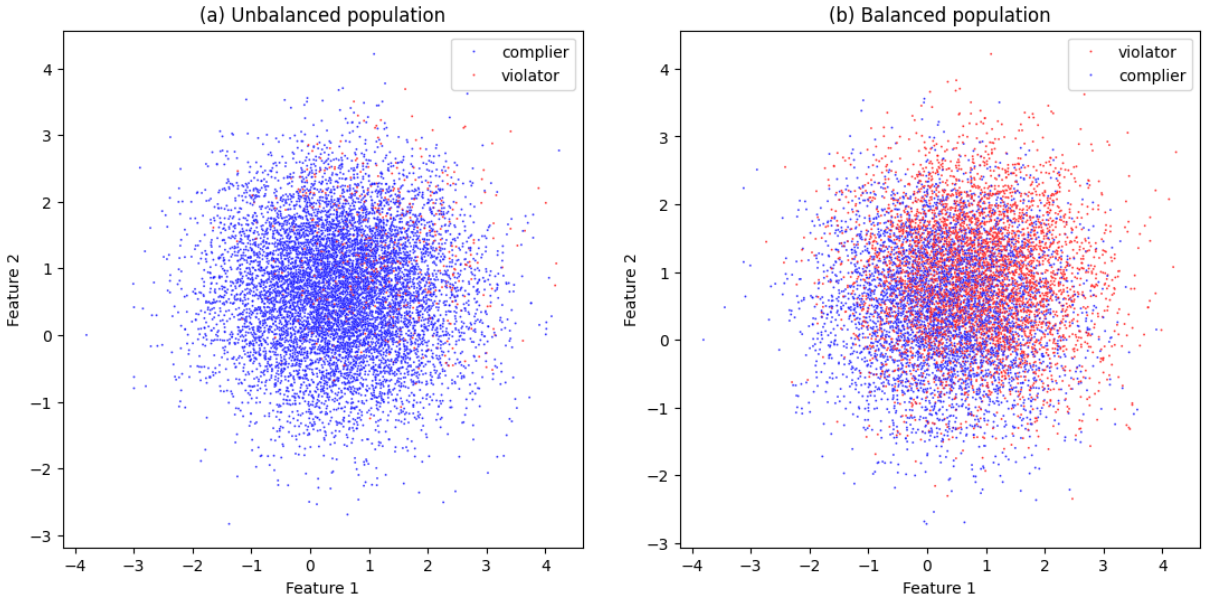

Figure 5. Examples of simulated populations with 95% class 0 cases (a) and balanced classes (b), plotted with respect to their two most informative features. Although there is a visual overlap of points representing the cases, the number of violators and compliers is the same in plot b, representing the conditions illustrated on the right side of Figure 1.

Figure 6 illustrates the simulation of four different types of samples derived from a population with unbalanced classes (identical to the one in Figure 5, a).

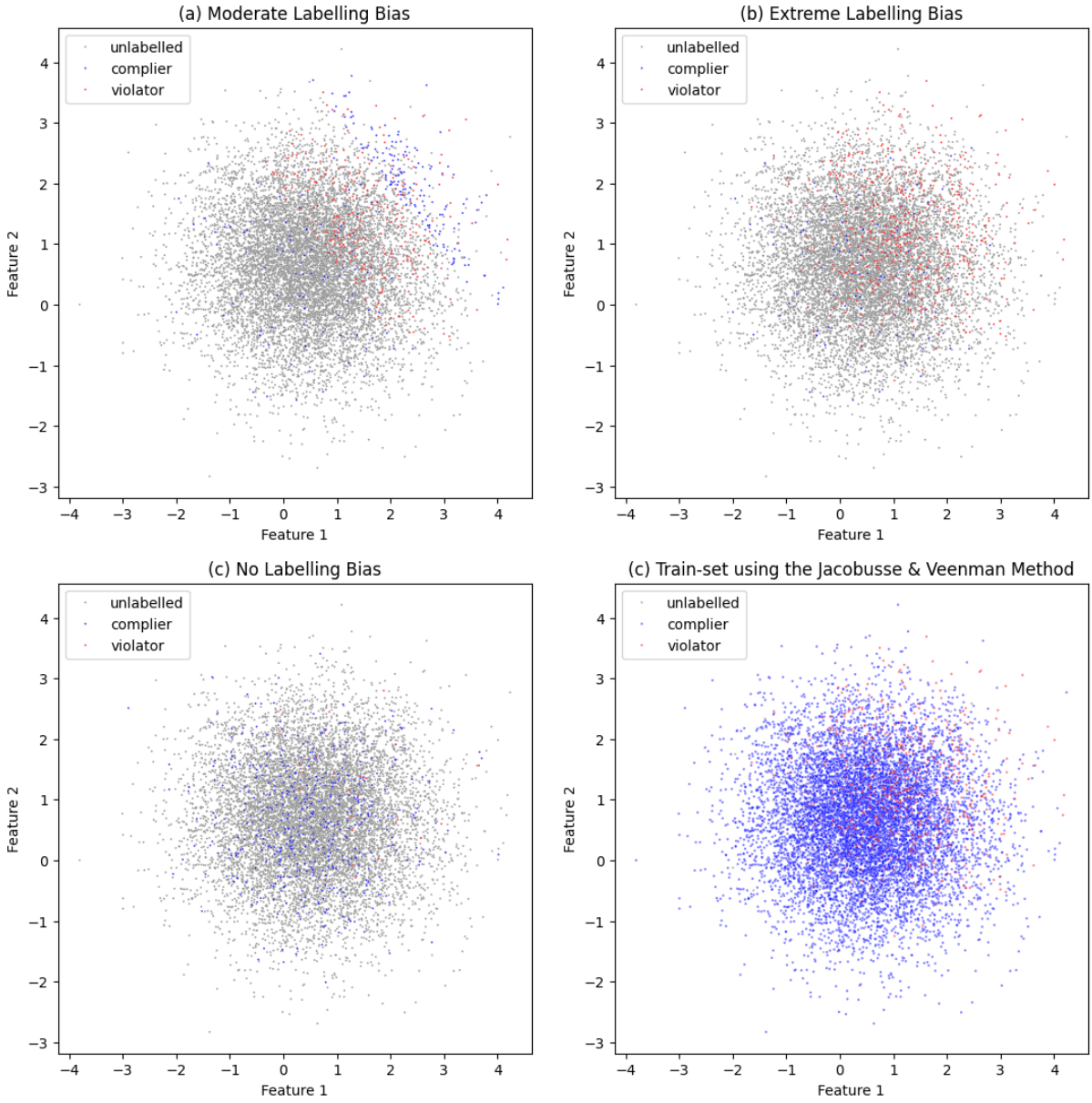

Figure 6. Different samples extracted from an unbalanced population. Labelling bias is evident in plots a and b, where coloured (labelled) cases cluster on one side, showing varying proportions of class 1. Extreme labelling bias results in mostly red dots, indicating an inverse proportion to the actual class distribution in the population. With no labelling bias (c), class proportions and distributions match the original population. Plot d shows a train-set after applying the "Jacobusse & Veenman Method" on a population with imbalanced classes and extreme labelling bias (as in plot b). Here, unlabelled dots (previously grey) now have the pseudo-label '0' (blue), while labelled class 0 cases have been removed (now in grey).

Results

This study aimed to compare a supervised machine learning method with semi-supervised techniques for handling imbalanced and biased-labelled data while using available unlabelled instances, mimicking scenarios encountered by the NLA. Forty-two simulation iterations were conducted to assess each method's performance based on precision@100 and algorithmic bias. Other 3 performance metrics were also collected: global precision, AUPRC, and precision@50, resulting in a total of 5 metrics. Each iteration included 2 levels of class imbalance (50% for each class and 5% for class 1) and 3 levels of labelling bias (no bias, moderate bias, and extreme bias), resulting in 6 simulated scenarios for each of the 6 tested methods. In total, methods were applied 42 iterations $\times$ 2 levels of class imbalance $\times$ 3 levels of labelling bias $\times$ 6 methods = 1512 times, resulting in the collection of 1512 $\times 5 = 7560$ performance measures.

Results of the simulations

The performance summary of all six models in populations with and without class imbalance can be seen in Figures 7 and 8. The plots at the bottom include, in grey, the indication of the average distance between the centroids of the true cases of class 1 in the sample and the test set. These bars serve the purpose of providing additional insight into the characteristics of the sample in each scenario. They are distinct from the algorithmic bias represented by the coloured bars specific to each method, which refer to the distance between the top 100 cases with the highest predicted probability of belonging to class 1 and the real class 1 cases in the test set.

Figure A1 (Appendix A) presents the outcomes of the six methods in one of the simulations (random state 42). The plots show the top-100 cases ranked by their predicted probability of belonging to class 1, distinguishing between true positives and false positives. Additionally, the figure displays the centroids of these 100 predicted cases and of the true

centroid of class 1 cases in the population, giving an indication of the magnitude of the algorithmic bias en each situation.

Results indicate variations in precision@100 and algorithmic bias across different methods and scenarios (the complete list of indicators for all methods in all scenarios is available in Tables A1 and A2, in Appendix A).

Findings Related to Hypothesis 1. The results indicate that, in scenarios with class imbalance and labelling bias, the "Jacobusse & Veenman Method" consistently outperformed the supervised learning method in terms of precision@100. This indicator is superior in both the situations of moderate labelling bias ($M = 0.538, SD = 0.069$, compared to $M = 0.414, SD = 0.066$ of the supervised method) and extreme labelling bias ($M = 0.6, SD = 0.077$, compared to $M = 0.525, SD = 0.071$). In addition, the algorithmic bias is lower for the "Jacobusse & Veenman Method" compared to the supervised method at all levels of labelling bias (see Figure 7 and Table A1). Thus, the findings support Hypothesis 1, giving evidence that the "Jacobusse & Veenman Method" provides better performance under these conditions.

Findings Related to Hypothesis 2. Semi-supervised methods based on point similarity (label spreading and label propagation) had higher precision@100 compared to the supervised learning and the "Jacobusse & Veenman" methods. This was not the case for the self-learning method, which showed low precision@100 in all scenarios (See Figure 7 and Table A1). The label propagation method reached the highest overall precision@100 in the simulations with unbalanced populations and labelling bias ($M = 0.632, SD = 0.075$ for moderate bias and $M = 0.686, SD = 0.069$ for extreme bias). Nevertheless, both label propagation and label spreading showed an increase in algorithmic bias, which is an important consideration for their practical application.

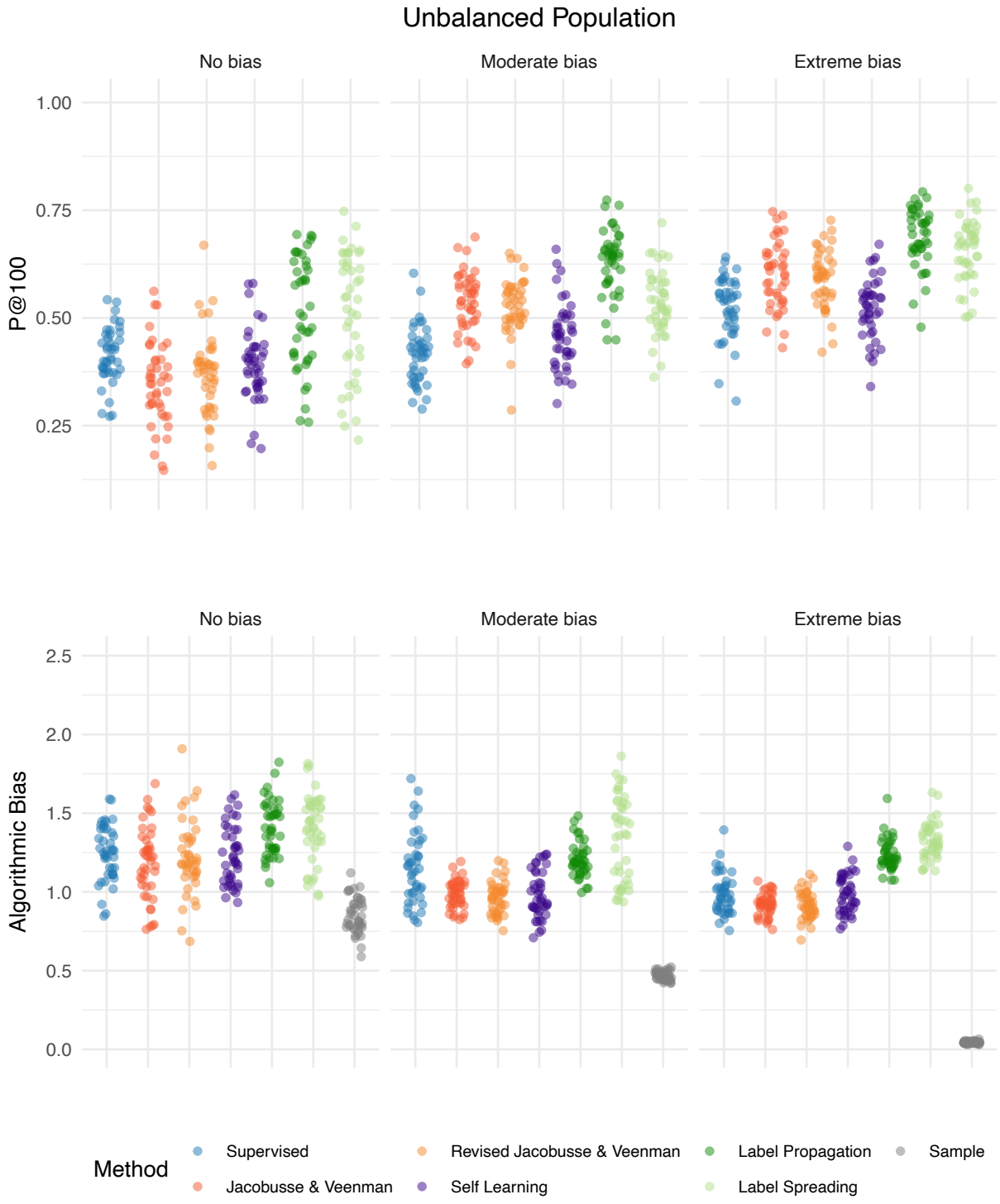

Figure 7. Performance of methods on unbalanced classes. Dots represent individual simulations. The grey bars in the bottom plots indicate the average distance between the true class 1 centroids in the sample and test set, distinct from the method-specific algorithmic bias shown by colored bars.

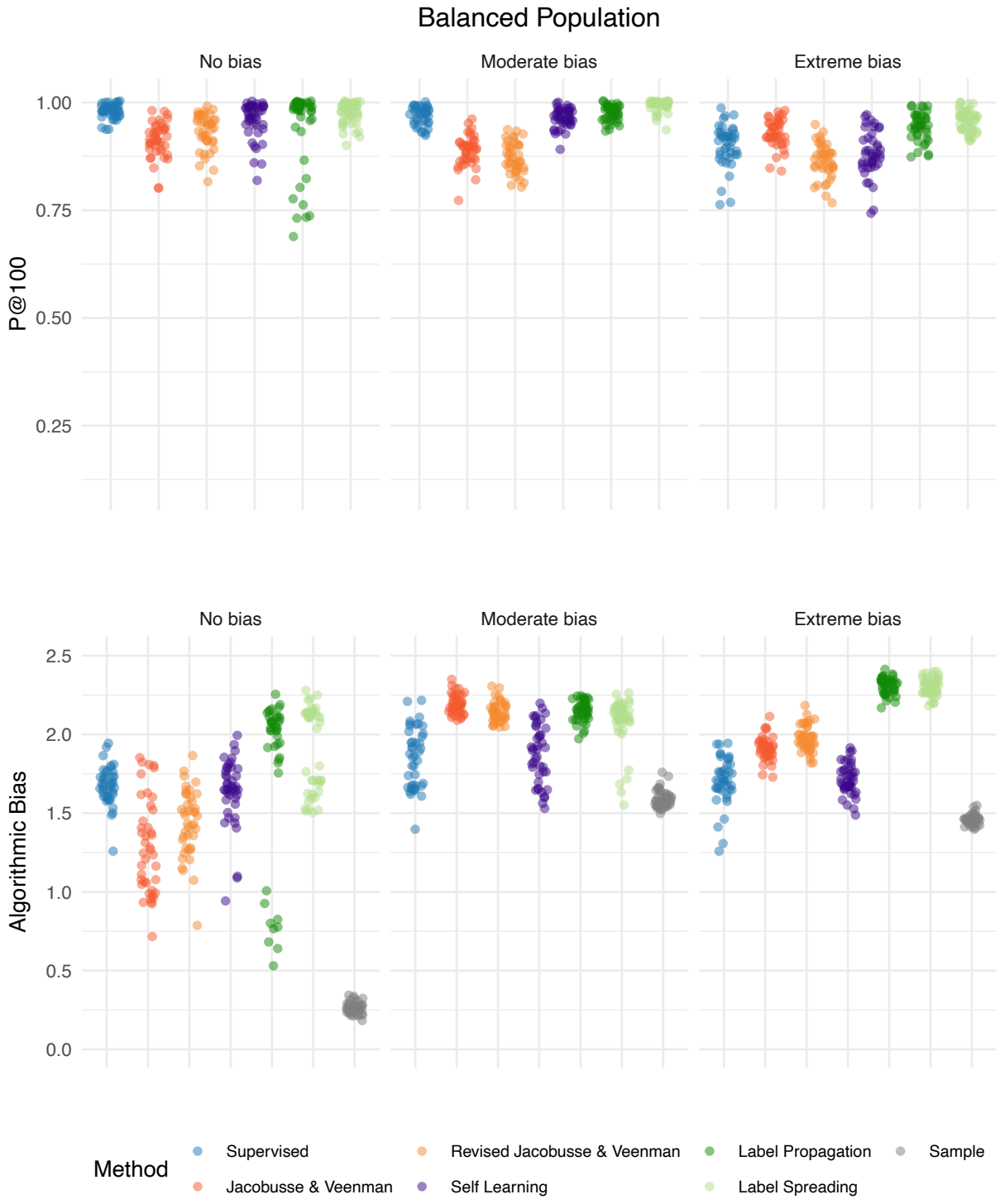

Figure 8. Performance of methods on balanced classes. Dots represent individual simulations. The grey bars in the bottom plots indicate the average distance between the true class 1 centroids in the sample and test set, distinct from the method-specific algorithmic bias shown by colored bars.

Discussion

For the NLA and other institutions aiming to use machine learning for predictive insights, it would be very handy to make use of all available information to improve their models, not just the limited labelled cases. After all, the NLA has extensive access to data about employers across the Netherlands, which can be translated into rich and useful information, potentially helping to find and address situations of unfair, unhealthy, or unsafe working conditions. This research aims to support this goal by comparing different semi-supervised methods in terms of ranking precision and algorithmic bias. These methods were applied in six different simulated scenarios, varying the composition of the classes in the population and the level of labelling bias in the sample. As expected, the results suggest that different methods are optimal under varying conditions. Supervised methods are preferred for balanced classes with no labelling bias, for their simplicity and precision. In contrast, semi-supervised methods, despite higher algorithmic bias, may be more suitable for unbalanced classes with labelling bias, as they generally achieved higher precision@100.

Scenarios with class imbalance

The first hypothesis of this research was aligned with the findings of Jacobusse and Veenman (2016), aiming to confirm that, in situations with class imbalance and selection bias of the labelled sample, better predictions could be achieved by including unlabelled cases in the training set. This set would then consist of real cases of class 1 and all unlabelled cases, which would receive a pseudo-class 0. The simulations in this study, which conceptually replicate the conditions of the experiment by Jacobusse and Veenman (2016), support this hypothesis. As seen in Figure 7 and Table A1, for a population with unbalanced classes, the "Jacobusse & Veenman Method" has a superior precision@100 than the supervised method in both the simulations of moderate labelling and extreme labelling

bias. This is due to the fact that only the labelled cases were used for the supervised approach, limiting the amount of information available to the model and consequently reducing its predictive accuracy. The presence of class imbalance exacerbates this issue by making the minority class even more underrepresented. In this sense, labelling bias can be somewhat beneficial for the supervised method, as it introduces more examples of class 1 cases into the sample, helping to capture the characteristics of this class. Because of that, for a population with unbalanced classes, the best performance of the supervised method occurs when labelling bias is extreme, resulting in the largest proportion of class 1 cases in the training set (see an example of distribution of classes in Table 1). In this scenario, the labelled class 1 sample closely matches the class 1 cases in the test set, as illustrated by the grey bar in Figure 7, leading to improved precision compared to when there is no labelling bias. However, because the labelling bias also skews the data in the direction of the most informative features, the generalisability of the supervised model remains limited. By incorporating unlabelled data, the "Jacobusse & Veenman Method" mitigates these issues, providing a more comprehensive view of the data distribution and improving the precision@100. Additionally, this method decreases algorithmic bias by using a more diverse training set which includes unlabelled cases and thus better represents the true distribution of the population across all features, leading to fairer and more balanced predictions. So, when applied to an unbalanced population with labelling bias, as outlined in its underlying principles, this method provides a straightforward solution to achieve a balanced outcome with robust performance and reduced algorithmic bias, compared to the supervised learning method (See Table A1).

As proposed in the second hypothesis, the semi-supervised methods, by dynamically using the unlabelled data, reached even superior levels of precision@100 compared to the "Jacobusse & Veenman Method", except for the self-learning method. Label spreading smooths the label information across the data graph by considering general similarity between data points (Zhou et al., 2003). This smoothing is intended to prevent the model

from fitting too closely to noisy or outlier data points, thereby enhancing its ability to generalize. However, this uniform spreading can sometimes reduce the influence of the minority class labels. In scenarios without outliers, such as in this study, this cautious approach may have slightly compromised its performance compared to label propagation. This other method not only spreads labels based on data similarity but also preserves the initial labelled information throughout its iterative process, more easily allowing the propagation of labels of rare cases (Bengio et al., 2006). This preservation helps maintain the integrity of the initial labelled data, ensuring that the model learns from these examples effectively. As a result, label propagation can achieve the highest precision@100 in simulations involving populations with unbalanced classes, even when faced with extreme labelling bias. Nevertheless, this increase in precision comes with the cost of an increase in algorithmic bias, although this effect is less accentuated for label propagation compared to label spreading (see Figure 7 and Table A1). This increase is because semi-supervised methods depend on the characteristics of the labelled data to disseminate pseudo-labels to other instances. Even if this propagation is smoothed or anchored on the original labels, these models will give more weight to cases that are more similar to the initial labelled sample. Because of this, the cases with higher predictive probability of belonging to class 1 will have a closer relation to the features that were more informative in the selection of the labelled sample. Thus, because precision@100 is the main metric used in this research, the results indicate that while semi-supervised methods can enhance precision in top-ranked predictions, they also introduce some level of algorithmic bias that needs to be carefully managed.

When it comes to the self-learning method, its effectiveness is dependent on the performance of a base classifier trained on the labelled data (O. Chapelle et al., 2009). In this study, this method was notably affected by the low representation of class 1 cases, leading to precision@100 values that were very close to or even lower than those achieved by the supervised method (see Figure 7 and Table A1). As a result, this method did not

bring considerable gain that justifies its adoption.

Finally, it is worth noting that the performance of label spreading and label propagation methods heavily relies on the accurate calibration of their hyper-parameters. In this experiment, Mean Reciprocal Rank (MRR) was used as a metric for model comparison, focusing on optimizing accuracy for the top 100 instances with the highest predicted probability of belonging to class 1. This approach involved setting the hyper-parameters alpha and gamma to low values, enhancing sensitivity to the underlying data structure and improving accuracy for cases closely resembling true class 1 samples. However, this setting led to lower global precision as higher-ranked cases beyond the top 100 were not prioritized in the optimization process (Table A1). This disparity highlights that, while label propagation excels at ranking top instances accurately, its overall performance across the entire dataset may vary with these specific hyper-parameters. Adjusting hyper-parameters for improved global precision could potentially lead to the use of higher values for alpha and gamma, which means that the models would be less dependent on the initial labelled data to assign pseudo-labels. This could increase the overall precision but at the cost of the accuracy of the lower-ranked predictions. Thus, although the semi-supervised methods based on point similarity demonstrate their usefulness in predicting lists of cases with the highest probability of being violators, their utility for predictions with high global precision needs to be further investigated.

Balanced Population

The supervised method demonstrates strong performance in scenarios with balanced classes and no labelling bias (see Figure 8 and Table A2). This result was anticipated, as the labelled sample, containing an equal distribution of observations for each class, ensures a representative dataset, leading to high precision and low algorithmic bias. Even when moderate labelling bias was introduced, model performance remained robust due to the consistent proportion of class 1 cases. With extreme labelling bias, the performance of the

supervised method showed a slight decline ($M = 0.9, SD = 0.049$). Only in these scenarios the semi-supervised methods, including the "Jacobusse & Veenman Method," achieved higher precision@100, though this came with increased algorithmic bias, particularly for label propagation and label spreading. So, for populations with balanced classes, the supervised method provides good precision with controlled levels of algorithmic bias, making it a reliable choice for such datasets.

Ad Hoc - "Revised Jacobusse & Veenman Method"

The "Revised Jacobusse & Veenman Method" is a semi-supervised approach that builds on the original "Jacobusse & Veenman Method" to create a hybrid training set. Initially, it divides the unlabelled data into 5 folds. Over 5 rounds, it combines 4 of these folds with all labelled cases and uses this combined set to predict the class probability of the unlabelled cases in the remaining fold, thereby creating a different training set for each round. In these training sets, labelled class 1 data is kept, labelled class 0 data is removed, and unlabelled data is included with a pseudo-label '0'. After the 5 rounds, a final training sample is formed, comprising labelled cases of class 1, unlabelled cases with over 50% probability of being class 1, and the remaining unlabelled cases with a pseudo-label '0'. labelled cases of class 0 are excluded. Finally, a random forest classifier is trained on this final sample.

Across the simulations, the performance of this method was similar to the "Jacobusse & Veenman Method" in terms of precision@100 and algorithmic bias, with a few exceptions. For example, in populations with unbalanced classes and no labelling bias, there are very few cases of class 1 in the labelled sample, making it difficult to capture the characteristics of this group and generalize to the entire population using supervised methods. In this scenario, the "Jacobusse & Veenman Method" amplifies the issue by further oversampling class 0, assigning pseudo-labels '0' to all the unlabelled cases. The "Revised Jacobusse & Veenman Method" mitigates this problem by assigning some

additional pseudo-labels '1' to the training set. Despite this improvement, the performance of the other tested methods remains superior in this scenario.

The main drawback of the "Revised Jacobusse & Veenman Method" is its reliance on a threshold to assign pseudo-labels '1' to the unlabelled sample. By default, it assigns pseudo-labels '1' to every case with more than a 50% probability of belonging to class 1. However, these thresholds are not universally applicable and depend heavily on the distribution of classes in the population. Future research could explore ways to optimize this threshold for different scenarios, improving the method's robustness and generalisability.

Limitations and Future Directions

The methods, measures, and design employed in this study demonstrated strengths and weaknesses by exploring various methods tailored to different types of populations and samples. The selection of Mean Reciprocal Rank (MRR) as an optimization metric proved particularly effective in evaluating precision@100 for identifying instances with the highest predicted probability of belonging to class 1, highlighting its relevance for practical applications in similar contexts.

However, a notable limitation of this study stems from its simulation-based approach, which relies on predefined characteristics to simulate the population. Simulations, by their nature, are constrained by the assumptions and parameters set during their design. In this study, simulations were used both to emulate the characteristics of a population and to reproduce a situation of labelling bias.

For the creation of the population, 20 features were constructed to reflect real-world scenarios, yet these features may not capture the full spectrum of complexities and variability present in actual data. Real-world populations exhibit a broader range of characteristics and interactions that simulations might oversimplify or fail to represent

accurately. For instance, while the simulation uses predefined distributions and relationships, real-world data often involves more complex, non-linear interactions and a wider variety of feature types, including binary or categorical variables, which may influence outcomes in nuanced ways. Consequently, the predictive models developed in this study might not fully generalize to real-world scenarios where the data may deviate significantly from these predefined patterns. Regarding the labelling process, the choice of using a linear combination of the two most informative features aimed to replicate a situation where inspectors prioritize segments of employers based on a few key characteristics. However, this approach may not fully capture the intricate decision-making processes of real inspectors, who might consider a broader and more complex set of criteria. Future research could benefit from either using real data or developing more complex simulations to validate and extend the findings of this study. Incorporating a more diverse set of feature types and interaction terms, or validating models with real-world data, would enhance the robustness of the findings and improve the applicability of the methods to a wider range of real-world situations.

Another constraint is the relatively modest sample sizes in certain scenarios, such as random sampling with only 25 cases, which may limit the generalisability of findings. Given the substantial size of the employer population in the Netherlands—approximately 2,000,000 active establishments (CBS, 2022)—it is crucial to scale methods effectively to handle large-scale datasets. Future studies should prioritize testing these methods on larger and more diverse datasets to validate and extend results effectively. Furthermore, applying these methods to real data or generating synthetic datasets based on real-world data will be essential for validating their efficacy in practical settings, thereby bridging the gap between theoretical simulations and real-world applicability.

Finally, future research should explore various levels of class imbalance and labelling bias to uncover insights for better model implementation. Investigating the percentage of pseudo-labelled instances in the training data will reveal its impact on performance.

Similarly, examining the effects of different proportions of biased-selected versus random instances in the sample could refine methods and improve accuracy. Additionally, understanding the impact of outliers on label spreading will help assess its robustness and reliability in practical applications.

Conclusion

The findings from this study suggest that the choice of method should be contingent upon the specific characteristics of the dataset and the objectives of the analysis. For scenarios with severe class imbalance and pronounced labelling bias, methods that effectively handle these challenges, such as the "Jacobusse & Veenman Method," may be particularly suitable. On the other hand, for more balanced datasets or where the focus is on minimizing algorithmic bias, traditional supervised methods can work sufficiently well.

The implications of this study extend beyond methodological advancements to practical applications in fields reliant on predictive modeling. By addressing inherent challenges like class imbalance and labelling bias, researchers and practitioners can enhance the reliability and fairness of predictive models in real-world settings. These findings contribute to ongoing efforts to improve the accuracy and ethical considerations of machine learning applications in various domains. In the context of the NLA, this study can help bring insights for the development and future applications of techniques that help to assure good working conditions across the Netherlands.

References

Baker, R. S., & Hawn, A. (2022). Algorithmic bias in education. *International Journal of Artificial Intelligence in Education*, 1–41.

Bengio, Y., Delalleau, O., & Le Roux, N. (2006). Label propagation and quadratic criterion. Semi-supervised learning. *Semi-Supervised Learning*.

Brownlee, J. (2011). Clever algorithms. *Nature-Inspired Programming Recipes*, *436*, 454.

Bruggeman, V., Milieu Consulting SRL, Xenidis, R., & van Giffen, B. (2023). *Artificial intelligence and algorithms in risk assessment: Addressing bias, discrimination and other legal and ethical issues*. https://www.ela.europa.eu/sites/default/files/2023-08/ELA-Handbook-AI-training.pdf

CBS. (2022). *How many companies in the Netherlands? - The Netherlands in numbers 2021*. https://longreads.cbs.nl/the-netherlands-in-numbers-2021/how-many-companies-in-the-netherlands/

Centraal Bureau voor de Statistiek. (2023). *Werkenden*. https://www.cbs.nl/nl-nl/visualisaties/dashboard-arbeidsmarkt/werkenden

Chapelle, Olivier., Scholkopf, Bernhard., & Zien, Alexander. (2006). *Semi-supervised learning*. MIT Press.

Chapelle, O., Scholkopf, B., & Zien, A. (2009). Semi-supervised learning (chapelle, o. Et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, *20*(3), 542–542.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Efimov, V. (2023). *Comprehensive Guide to Ranking Evaluation Metrics - towards Data Science*. https://towardsdatascience.com/comprehensive-guide-to-ranking-evaluation-metrics-7d10382c1025

Elkan, C. (2001). The foundations of cost-sensitive learning. *International Joint Conference on Artificial Intelligence*, *17*, 973–978.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

GASearchCV — sklearn genetic opt 0.10.1 documentation. (2023). https://sklearn-genetic-opt.readthedocs.io/en/stable/api/gasearchcv.html

Iscen, A., Tolias, G., Avrithis, Y., & Chum, O. (2019). Label propagation for deep semi-supervised learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5070–5079.

Jacobusse, G., & Veenman, C. (2016). On selection bias with imbalanced classes. *Discovery Science: 19th International Conference, DS 2016, Bari, Italy, October 19–21, 2016, Proceedings 19*, 325–340.

James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

LabelPropagation. (2024). https://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.LabelPropagation.html#sklearn.semi_supervised.LabelPropagation

LabelSpreading. (2024). https://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.LabelSpreading.html#sklearn.semi_supervised.LabelSpreading

Lum, K., & Isaac, W. (2016). To predict and serve? *Significance*, *13*(5), 14–19.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, *54*(6), 1–35.

Ministerie van Sociale Zaken en Werkgelegenheid. (2024, March 27). *Jaarverslag 2023*. https://www.nlarbeidsinspectie.nl/publicaties/jaarverslagen/2024/03/27/jaarverslag-2023

Mugari, I., & Obioha, E. E. (2021). Predictive policing and crime control in the united states of america and europe: Trends in a decade of research and the future of predictive policing. *Social Sciences*, *10*(6), 234.

Oliveira, W. D. G. de, & Berton, L. (2023). A systematic review for class-imbalance in

semi-supervised learning. *Artificial Intelligence Review*, *56*(Suppl 2), 2349–2382.

Olson, R. S. (2024). *TPOT API - TPOT*. http://epistasislab.github.io/tpot/api/

Rizve, M. N., Duarte, K., Rawat, Y. S., & Shah, M. (2021). In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv Preprint arXiv:2101.06329*.

SelfTrainingClassifier. (2024). https://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.SelfTrainingClassifier.html#sklearn.semi_supervised.SelfTrainingClassifier

Shah, N., Bhagat, N., & Shah, M. (2021). Crime forecasting: A machine learning and computer vision approach to crime prediction and prevention. *Visual Computing for Industry, Biomedicine, and Art*, *4*(1), 9.

Sociale Zaken en Werkgelegenheid, M. van. (2022). *What does the Netherlands Labour Authority do*. https://www.nllabourauthority.nl/publications/publications/2022/01/24/what-does-the-netherlands-labour-authority-do

Triguero, I., Garcia, S., & Herrera, F. (2015). Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*, *42*, 245–284.

Van Giffen, B., Herhausen, D., & Fahse, T. (2022). Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods. *Journal of Business Research*, *144*, 93–106.

Van Rossum, G., & Drake Jr, F. L. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Williamson, E. J., & Forbes, A. (2014). Introduction to propensity scores. *Respirology*, *19*(5), 625–635.

Zhou, D., Bousquet, O., Lal, T., Weston, J., & Schölkopf, B. (2003). Learning with local and global consistency. *Advances in Neural Information Processing Systems*, *16*.

Zhu, X., & Ghahramanih, Z. (2002). Learning from labeled and unlabeled data with label propagation. *ProQuest Number: Information to All Users*.

Appendix A

Performance measures

Table A1

Performance Summary of Methods for Populations with Unbalanced Classes: Mean and Standard Deviation

		Precision		AUPRC		P@50		P@100		Alg. Bias	
Labelling Bias	Method	M	SD	M	SD	M	SD	M	SD	M	SD
No bias	Supervised	0.508	0.214	0.224	0.037	0.477	0.085	**0.412**	0.066	**1.244**	0.177
No bias	Jacobusse & Veenman	0.268	0.317	0.185	0.043	0.395	0.120	**0.347**	0.095	**1.186**	0.229
No bias	Revised Jacobusse & Veenman	0.389	0.272	0.197	0.043	0.406	0.112	**0.371**	0.096	**1.217**	0.235
No bias	Self Learning	0.342	0.274	0.213	0.048	0.438	0.107	**0.390**	0.084	**1.234**	0.186
No bias	Label Propagation	0.015	0.068	0.301	0.080	0.561	0.158	**0.512**	0.128	**1.407**	0.172
No bias	Label Spreading	0.048	0.216	0.310	0.085	0.571	0.174	**0.511**	0.144	**1.407**	0.217
Moderate bias	Supervised	0.119	0.014	0.227	0.033	0.471	0.083	**0.414**	0.066	**1.151**	0.226
Moderate bias	Jacobusse & Veenman	0.621	0.211	0.315	0.040	0.595	0.095	**0.538**	0.069	**0.986**	0.088
Moderate bias	Revised Jacobusse & Veenman	0.552	0.155	0.306	0.037	0.581	0.084	**0.528**	0.065	**0.962**	0.102
Moderate bias	Self Learning	0.240	0.162	0.272	0.045	0.526	0.104	**0.460**	0.078	**0.982**	0.146
Moderate bias	Label Propagation	0.210	0.256	0.360	0.049	0.720	0.084	**0.632**	0.075	**1.204**	0.110
Moderate bias	Label Spreading	0.115	0.053	0.300	0.046	0.607	0.097	**0.537**	0.074	**1.337**	0.272
Extreme bias	Supervised	0.073	0.008	0.309	0.042	0.581	0.088	**0.525**	0.071	**0.986**	0.127
Extreme bias	Jacobusse & Veenman	0.630	0.112	0.371	0.041	0.662	0.095	**0.600**	0.077	**0.921**	0.071
Extreme bias	Revised Jacobusse & Veenman	0.594	0.126	0.367	0.041	0.658	0.085	**0.591**	0.068	**0.915**	0.087

Table A1 continued

		Precision		AUPRC		P@50		P@100		Alg. Bias	
Labelling Bias	Method	M	SD	M	SD	M	SD	M	SD	M	SD
Extreme bias	Self Learning	0.074	0.006	0.311	0.040	0.569	0.089	**0.522**	0.070	**0.985**	0.120
Extreme bias	Label Propagation	0.050	0.000	0.424	0.051	0.765	0.078	**0.686**	0.069	**1.233**	0.097
Extreme bias	Label Spreading	0.050	0.000	0.406	0.051	0.744	0.080	**0.655**	0.073	**1.321**	0.114

Note. Main performance metrics are in bold. Additional performance metrics were also included for a more comprehensive overview.

Table A2

Performance Summary of Methods for Populations with Balanced Classes: Mean and Standard Deviation

		Precision		AUPRC		P@50		P@100		Alg. Bias	
Labelling Bias	Method	M	SD	M	SD	M	SD	M	SD	M	SD
No bias	Supervised	0.718	0.021	0.791	0.021	0.987	0.018	**0.980**	0.016	**1.677**	0.120
No bias	Jacobusse & Veenman	0.638	0.396	0.720	0.035	0.921	0.055	**0.913**	0.041	**1.290**	0.301
No bias	Revised Jacobusse & Veenman	0.798	0.200	0.736	0.031	0.944	0.045	**0.932**	0.041	**1.440**	0.210
No bias	Self Learning	0.700	0.038	0.768	0.045	0.967	0.042	**0.959**	0.043	**1.636**	0.214
No bias	Label Propagation	0.651	0.251	0.805	0.057	0.944	0.096	**0.940**	0.095	**1.770**	0.540
No bias	Label Spreading	0.736	0.080	0.818	0.028	0.974	0.031	**0.974**	0.026	**1.903**	0.268
Moderate bias	Supervised	0.795	0.040	0.761	0.030	0.977	0.020	**0.969**	0.021	**1.852**	0.189
Moderate bias	Jacobusse & Veenman	0.836	0.024	0.642	0.013	0.897	0.044	**0.888**	0.035	**2.192**	0.062
Moderate bias	Revised Jacobusse & Veenman	0.820	0.026	0.630	0.016	0.880	0.059	**0.871**	0.036	**2.144**	0.063
Moderate bias	Self Learning	0.779	0.040	0.745	0.032	0.974	0.028	**0.965**	0.022	**1.875**	0.185
Moderate bias	Label Propagation	0.499	0.505	0.819	0.023	0.986	0.019	**0.976**	0.018	**2.149**	0.067
Moderate bias	Label Spreading	0.955	0.156	0.839	0.023	0.996	0.013	**0.993**	0.013	**2.078**	0.164
Extreme bias	Supervised	0.624	0.054	0.709	0.019	0.908	0.061	**0.900**	0.049	**1.705**	0.151
Extreme bias	Jacobusse & Veenman	0.817	0.025	0.656	0.019	0.937	0.043	**0.929**	0.033	**1.911**	0.075
Extreme bias	Revised Jacobusse & Veenman	0.788	0.021	0.645	0.017	0.876	0.041	**0.865**	0.040	**1.975**	0.083

Table A2 continued

		Precision		AUPRC		P@50		P@100		Alg. Bias	
Labelling Bias	Method	M	SD	M	SD	M	SD	M	SD	M	SD
Extreme bias	Self Learning	0.661	0.030	0.697	0.014	0.889	0.066	**0.881**	0.053	**1.715**	0.098
Extreme bias	Label Propagation	0.500	0.000	0.750	0.020	0.959	0.034	**0.944**	0.033	**2.307**	0.054
Extreme bias	Label Spreading	0.500	0.000	0.760	0.023	0.969	0.029	**0.958**	0.025	**2.302**	0.057

Note. Main performance metrics are in bold. Additional performance metrics were also included for a more comprehensive overview.

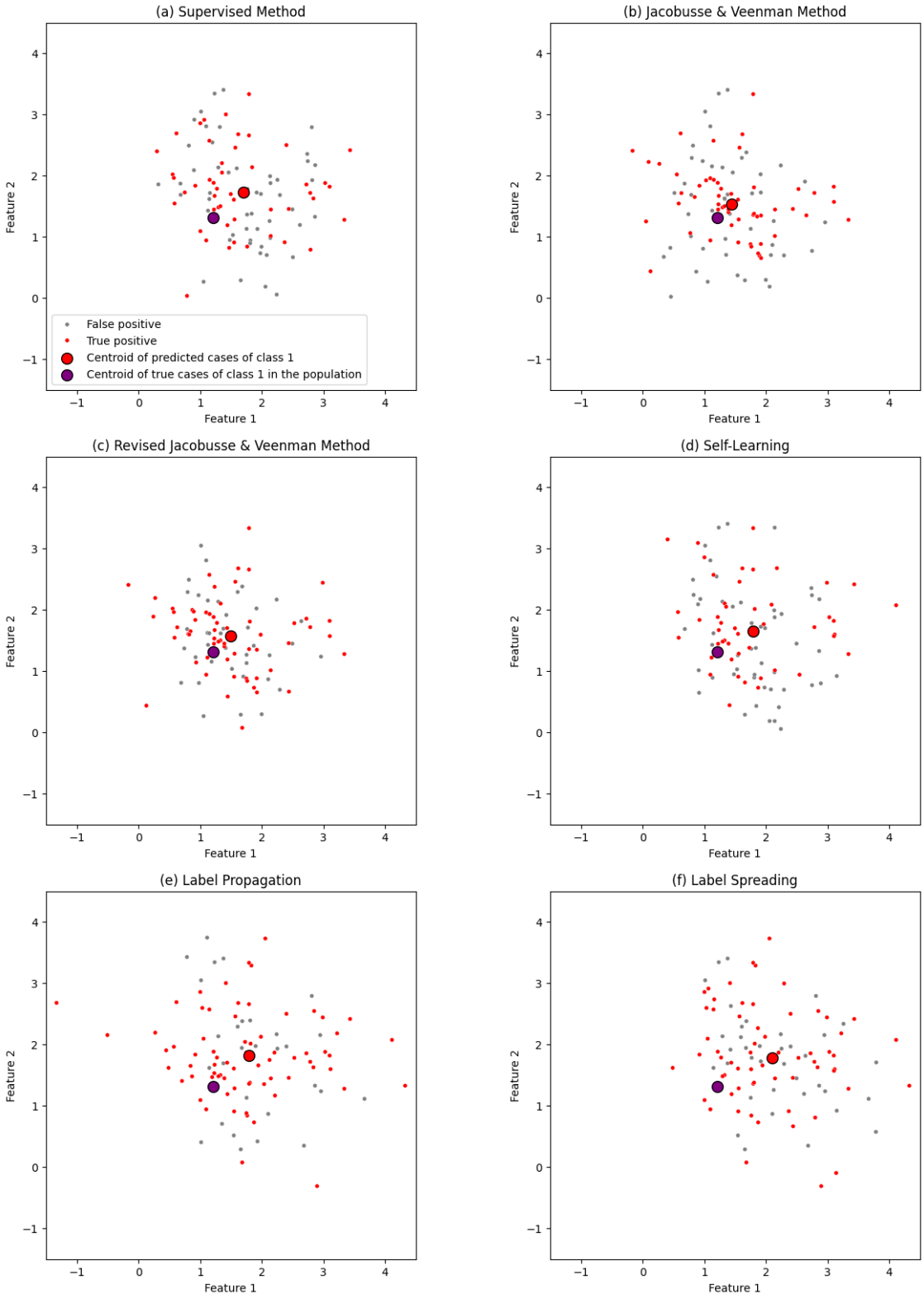

Figure A1. Example of 100 highest prediction probabilities for all 6 methods when applied to a population with unbalanced classes and extreme labelling bias (as in Figure 5, a). Only the 2 most informative features are displayed.

Appendix B

Code

All simulations in this study were created using Python (Van Rossum & Drake Jr, 1995). The code is available via the link:

https://github.com/lexbsb/Classification-Approaches-for-Data-With-Class-Imbalance-and-Biased-Labelled-Sample