ERASMUS UNIVERSITEIT ROTTERDAM

Erasmus School of Economics

Master Thesis Business Analytics & Quantitative Marketing

Robust Probabilistic Forecasting of Binary Outcomes with Regularisation

Author, student number: G.J. (Brian) Leenen, 476908

Supervisor: Dr. M. (Mikhail) Zhelonkin
Second assessor: Dr. A. (Andreas) Alfons

March 23, 2021

Abstract

In many classification problems we are not only interested in the class to which an event belongs, but also in the probability that the event belongs to the class. This paper studies whether logistic regression can produce calibrated class probability estimates in a high-dimensional setup, both in the presence and absence of (rowwise or cellwise) contamination. Using simulated data as well as real data, we compare multiple robust and non-robust logistic regression estimators and investigate whether they can produce calibrated forecasts. Concurrently, we investigate whether there is a preferable variable selection method, as well as how logistic regression performs compared to two popular machine learning classifiers. We find that, in the absence of contamination, logistic regression can produce calibrated forecasts in high dimensions and outperform the machine learning methods, as long as variable selection is applied. However, when contamination is introduced, the non-robust methods break as expected. We show that the robust and regularised alternatives currently available in the literature have unexpected behaviours, and provide simulation evidence explaining the behaviour of one of them.

Contents

1 Introduction

Prediction of binary outcomes is a commonly recurring problem in a plethora of disciplines in *inter alia* the biomedical, economic and natural sciences. In many instances we are not only interested in the class to which an event belongs, but also in the probability that the event belongs to the class. In other words, we want to get an idea of how certain we are about our predictions; if we think that an event will happen with probability 99%, then we are much more certain about our prediction than if we assign a probability of 51%. As our decision making may depend on how confident we are, being able to accurately predict these probabilities is of paramount importance.

Logistic regression is among the most frequently used models to compute such probabilities, if not the most used model (Hastie et al., 2009). In logistic regression, we assume response y_i follows a Bernoulli distribution with probability

$$\mathbb{P}(y_i = 1 | X_i = x) \equiv \sigma(x'\beta) = (1 + e^{-x'\beta})^{-1},$$

where x is a p-dimensional vector of predictors and β is the corresponding vector of regression coefficients. β is then estimated by minimising the negative log-likelihood

$$\ell(\beta) = \sum_{i=1}^{n} \left(-y_i(X_i'\beta) + \log\left(1 + e^{X_i'\beta}\right) \right). \tag{1}$$

By well-established results in classical statistics, it holds under suitable distributional assumptions that for fixed p and sample size $n \to \infty$ the resulting maximum likelihood estimate $\hat{\beta}$ is asymptotically normal with mean β and variance $\frac{1}{n}\mathcal{I}^{-1}(\beta)$, where $\mathcal{I}$ is the information matrix. Further, the maximum likelihood estimator is efficient, meaning it is optimal for our probabilistic forecasting purposes; we can obtain approximately unbiased class probabilities with the smallest variance.

Two key assumptions underlying the preceding results are that Equation (1) is the correct specification of the log-likelihood and that we have a large sample size n, which is much larger than the dimension of the problem p. The former condition pertains to potential model misspecification, which is an issue with any statistical model. The latter point, however, concerns the data that we have available and is an issue which has risen to prominence with the emergence of large, modern datasets. When the $n \gg p$ regime is replaced with one where p/n does not tend to zero as $n \to \infty$, classical results regarding the behaviour of the maximum likelihood estimator fail (Sur and Candès, 2019).

The setting where the dimensionality is so large that we have $p > n$ has received much attention in the literature, see e.g. Bühlmann and van de Geer (2011) for an extensive overview. In this high-dimensional setting, the maximum likelihood estimate $\hat{\beta}$ is undefined. To resolve this, one generally assumes that among the entire set of p predictors only a subset only belongs in the model, in which case the goal becomes to select the informative predictors as well as accurately estimate the associated parameters. In other words, the coefficient vector β is assumed to be sparse according to some structure. To obtain accurate estimates of the class probabilities $\mathbb{P}(y_i = 1 | X_i = x)$ we must uncover that structure.

In this research, we are interested in our ability to accurately estimate class probabilities using logistic regression when p/n is larger than classically assumed and in the high-dimensional setting. We investigate this problem in the setting where all data are generated by the same model distribution, as well as in the setting where the data is contaminated with outliers. In doing so, we attempt to answer the following research question and subquestions:

Can logistic regression produce accurate class probability estimates when data is high-dimensional?

- How does this result change when we contaminate the data with extreme predictor values and with misclassified observations?
- Is there a variable selection method that is preferable for probabilistic forecasting?
- How does the performance of logistic regression compare to that of popular machine learning classifiers?

To investigate these questions, we employ a simulation study and apply our results to real-life data.

The rest of this paper is structured as follows. Section 2 provides an overview of sparse logistic regression theory and prior work on robust regularised regression. Subsequently, Section 3 introduces the methodology used in the simulation study of Section 4 and real data analyses of Section 5. In Section 6 we discuss the results and their limitations, and further provide some suggestions for future research. Section 7 then concludes.

2 Theoretical Framework

In this section, we first formalise the notion of sparsity introduced in Section 1 and discuss sparse logistic regression. Subsequently, we provide an overview of prior work on robust regression with sparsity. Our discussion primarily focusses on the rowwise contamination paradigm, as to the best of our knowledge no work has been published on logistic regression methods that are robust against cellwise outliers. Instead, we provide an overview of developments pertaining to cellwise robust methods and how they may be applied to (sparse) logistic regression.

2.1 Review of Sparse Logistic Regression

The following is an overview of developments in sparse regression and how they may be used in logistic regression. We only discuss the lasso, the elastic net and best subset selection, but several other (non-convex) variable selection methods have been proposed. See for example Tibshirani (2011) for a summary of other sparse regression techniques and other extensions of the lasso.

2.1.1 Sparsity and the Lasso

In the literature, sparsity is generally understood as parameter vector β consisting of some set $\beta_{\mathcal{A}}$ of non-zero coefficients and a remainder $\beta_{\mathcal{A}^c}$ of all-zero coefficients. $\mathcal{A}$ is called the active set and recovery of the active set is referred to as consistency in variable selection (e.g. Zou, 2006) or support recovery (e.g. Salehi et al., 2019). Clearly, if we knew $\mathcal{A}$ in advance and the cardinality of $\mathcal{A}$ is less than n, the sparse regression problem reduces to an ordinary regression problem. This hypothetical ideal leads to the notion of what is called an oracle in the literature (Fan and Li, 2001; Fan and Peng, 2004). The oracle estimates $\beta_{\mathcal{A}}$ as accurately as possible, if $\mathcal{A}$ is known in advance. In our logistic regression setting, the oracle is given by the maximum likelihood estimator using only the variables in $\mathcal{A}$.

In practice, one does not know $\mathcal{A}$ and needs a selection procedure to choose the predictors to include in the model. One commonly used selection procedure, the Least Absolute Shrinkage and Selection Operator, abbreviated to lasso, was developed by Tibshirani (1996). The lasso achieves variable selection by penalising the l_1-norm of β, which is known to produce sparse solutions (Hastie et al., 2009). Though the author primarily considered the linear regression problem, the lasso is also broadly used for logistic regression. To this end, we extend the objective function of Equation (1) with the lasso penalty to form a penalised log-likelihood

$$\ell_{lasso}(\beta) = \ell(\beta) + n\lambda \sum_{j=1}^{p} |\beta_j|, \tag{2}$$

where $\ell(\beta)$ is the negative log-likelihood of Equation (1), $|\beta_j|$ denotes the absolute value of the j-th element of β, and λ is a tuning parameter to be chosen by cross-validation (CV) or an information criterion (e.g. Zou, 2018). The objective in Equation (2) is convex and there exist fast algorithms to minimise it. Further, under suitable regularity conditions on the design matrix and the size of the non-zero coefficients, lasso recovers the support, which helps explain the algorithm's popularity. See Bühlmann and van de Geer (2011) for an elaborate explanation of the conditions under which consistency in variable selection holds.

A drawback of the lasso is that it produces biased parameter estimates by design, which may in turn affect model forecasts. This is because, even if lasso achieves support recovery, the l_1-norm penalty shrinks all p components of estimate $\hat{\beta}_{lasso}$, including our estimates of coefficients $\beta_{\mathcal{A}}$. Since our class probability forecasts are given by $\sigma(X\hat{\beta}_{lasso})$, the lasso produces biased forecasts.

2.1.2 The Adaptive Lasso

In addition to probabilistic forecasting, the bias of lasso severely complicates its use in statistical inference. Zou (2006) showed that although the lasso estimator is consistent in variable selection, it is not consistent in estimation of coefficients $\beta_{\mathcal{A}}$. For inference, one requires an estimator that is consistent in both senses. An estimator that has this feature is said to possess the oracle property. Zou (2006) proposed the adaptive lasso (Adalasso) estimator. Adalasso replaces the lasso penalty in Equation (2) by a reweighted version, which begets a log-likelihood of the form

$$\ell_{adalasso}(\beta) = \ell(\beta) + n\lambda \sum_{j=1}^{p} w_j|\beta_j|. \tag{3}$$

Here, $\ell(\beta)$ is again the negative log-likelihood of Equation (1). Zou (2006) proved that for suitable w_j, the adalasso has the oracle property asymptotically ($n \to \infty$). The author suggested weights of the form $w_j = |\hat{\beta}_{init,j}|^{-1}$, where $\hat{\beta}_{init,j}$ is an initial estimate of β_j that is $\sqrt{n}$-consistent in variable selection. The idea of this choice of weights is as follows. If we find near-zero initial estimates for some coefficients, then we suspect that those coefficients are true zeros and we want to fix them at zero. By using the reciprocal of a near-zero initial estimate as a penalty factor, we assign large penalties to those coefficients, inducing adalasso to shrink the coefficients towards zero much more quickly than the lasso does. Similarly, if we find larger initial estimates for the some parameters, we suspect they are part of $\beta_{\mathcal{A}}$ and belong in the model. We therefore do not wish to penalise them as heavily and assign the coefficients a smaller penalty. The effect of shrinkage is less pronounced for these parameters as a result. In this way, we may be able to eliminate the irrelevant coefficients before the penalty meaningfully shrinks the relevant coefficients (Hastie et al., 2015).

Zou (2006) originally proposed to use the maximum likelihood estimator to compute the w_j for the generalised linear model, but when $p > n$ the estimator is undefined. Huang et al. (2008) instead suggest using the lasso estimator which minimises Equation (2), setting $\beta_j = 0$ if $\beta_{lasso,j} = 0$. The authors call this approach the *iterated lasso*. The iterated lasso is also advocated by Bühlmann and van de Geer (2011).

2.1.3 The Elastic Net

The adalasso was introduced as an alternative to the lasso that possesses the oracle property. However, the lasso has another prominent shortcoming, whereby its coefficient shrinkage paths are unstable when variables are highly correlated. Zou and Hastie (2005) proposed an extension of the lasso which replaces the penalty in Equation (2) by a convex combination of the l_1 and (squared) l_2 norms. For logistic regression, the objective function is then given by

$$\ell_{enet}(\beta) = \ell(\beta) + n\lambda \left(\alpha \sum_{j=1}^{p} |\beta_j| + \frac{(1-\alpha)}{2} \sum_{j=1}^{p} \beta_j^2 \right), \quad 0 \leq \alpha \leq 1, \tag{4}$$

which is still convex. Zou and Hastie (2005) called this the elastic net (EN). Through the incorporation of the l_2-norm used in ridge regression, the formulation in Equation (4) encourages sharing of coefficients among highly correlated variables. This stabilises the coefficient paths and further avoids lasso's tendency to arbitrarily set one of multiple correlated variables to zero, while keeping another in the model. See Hastie et al. (2015) for an explanation of these phenomena.

Note that the EN is essentially a generalisation of the unweighted lasso penalty shown in Equation (2), hence logistic regression with the EN does not exhibit the oracle property. Zou and Zhang (2009) introduced a natural extension of the EN that addresses this point analogously to Adalasso, which they suitably named the adaptive elastic net (AdaEN). The AdaEN makes a simple adjustment to Equation (4) in which each $|\beta_j|$ (but not the β_j^2) is multiplied by a weighting factor w_j as in Equation (3). Weights are again chosen as $w_j = |\hat{\beta}_{init,j}|^{-1}$. In the high-dimensional setting the EN is used as initial estimate. Zou and Zhang (2009) show that adaEN asymptotically possesses the oracle property under weak regularity conditions.

2.1.4 New Developments in Best Subset Selection

The lasso is sometimes understood as a convex relaxation of best subset selection (Hastie et al., 2017), which aims to find the subset of k regressors that best fits the data, e.g. in the sense of smallest loss. In the context of logistic regression, the best subset selection (BSS) operator minimises the log-likelihood with 'l_0' penalty

$$\ell_{\mathcal{S}^*}(\beta) = \ell(\beta) + n\lambda \sum_{j=1}^{p} 1\{\beta_j \neq 0\}, \tag{5}$$

where $1\{\cdot\}$ is the indicator function, which equals unity when the argument is true and zero otherwise. This approach directly controls the sparsity of the model by forcing a fit that only uses k predictors and holds the benefit of not shrinking the non-zero parameters, possibly resolving issues caused by the bias that the lasso penalty introduces.

Until recently, the practicality of best subset selection was severely inhibited by the fact that minimising Equation (5) is NP-hard. For problems with $p > 30$, best subset selection may entail unacceptably long computation times. New work in optimisation has resolved this drawback to a great extent. An important result in this regard was derived by Bertsimas et al. (2016), who show that minimising Equation (5) may be tackled as a mixed integer programming problem. This means that advancements in mixed integer optimisation can be used to fit best subset selection, making the estimator tractable in dimensions previously considered infeasible. Research interest in the mixed integer optimisation approach has grown quickly as a result, see for example Dedieu et al. (2020) and the references therein.

An extensive comparison of the predictive performance of best subset selection and the lasso in linear regression is made in Hastie et al. (2017). The authors argue that the lasso cannot simply be understood as a heuristic for best subset selection and that which procedure is superior depends on the data. They compare simulations where the signal-to-noise ratio (indirectly given by the R^2 in linear regression) is high or low. In the high signal-to-noise regime, best subset selection outperforms lasso, whereas in the more noisy regime best subset selection overfits and lasso has greater predictive accuracy[1]. This behaviour leads Hazimeh and Mazumder (2019) to consider an extended version of the l_0-regularised estimator in Equation (5), which uses a continuous shrinkage penalty to prevent overfitting. Dedieu et al. (2020) subsequently introduce an equivalent estimator for the logistic regression case:

$$\ell_{\mathcal{S}_p^*}(\beta) = \ell(\beta) + n\left(\lambda_0\|\beta\|_0 + \lambda_q\|\beta\|_q^q\right), \quad q \in \{1, 2\}. \tag{6}$$

[1] Hastie et al. (2017) actually show that their version of the relaxed lasso (comparable to Adalasso) does best overall, but we do not discuss that estimator here.

2.2 (Rowwise) Robust Regularised Regression

As in the low-dimensional setting, the regression objective functions introduced in Section 2.1 may break when the data are contaminated with outlying observations. To develop estimators that are applicable when $p > n$ and are robust to such rowwise outliers, a new branch of literature has emerged. In the following, we provide an overview of work on robust regularised regression and discuss how this paper contributes to the literature.

2.2.1 The Linear Regression Case

Most work on robust regularised regression so far focusses on the linear regression case, see Section 5.10 of Maronna et al. (2019). Early work on robust sparse linear regression includes that of Khan et al. (2007), who develop a robust version of least angle regression (Efron et al., 2004) for variable selection. Subsequent proposals replaced the quadratic loss of lasso least squares by other convex functions (Wang et al., 2007; Li et al., 2011), but the resulting estimators have unbounded influence functions (Maronna et al., 2019). Alfons et al. (2013) take a different approach. Instead of 'robustifying' a regularised estimator, they start from a robust regression estimator and extend it with a regularisation penalty. Specifically, the authors consider the least trimmed squares (LTS) estimator of Rousseeuw (1984) with a lasso penalty. Being an extension of LTS, this method is inefficient at the model distribution (i.e. normal errors), however.

Research pertaining to robust regularised linear regression is ongoing. See Gijbels and Vrinssen (2015), Smucler and Yohai (2017), Kong et al. (2018) and Amato et al. (2020) and their references for some recent developments. However, these works have limited relevance for this research, e.g. Gijbels and Vrinssen (2015) and Smucler and Yohai (2017) use the MM-estimator of Yohai (1987), which is not applicable in logistic regression. We therefore do not elaborate further on these papers.

2.2.2 The Logistic Regression Case

Research on robust regularised logistic regression is novel. An early study that addresses robust maximum likelihood methods with regularisation in a high-dimensional context is that of Neykov et al. (2014), but they only consider multiple linear regression and Poisson regression. Two proposals explicitly addressing the case of logistic regression are made in Kurnaz et al. (2018b) and Avella-Medina and Ronchetti (2017). In this section we briefly discuss how their estimators fit in the robust logistic regression literature, but full implementation details are deferred to Section 3.

Kurnaz et al. (2018b) essentially extend the work of Alfons et al. (2013) on sparse least trimmed squares in two ways. First, the authors generalise the use of the lasso penalty to the EN. Second, and more importantly for this paper, they consider logistic regression as well as linear regression. Starting from the estimator proposed by Bianco and Yohai (1996), which was subsequently improved by Croux and Haesbroeck (2003), the authors develop a robust and regularised logistic regression estimator which is fitted by minimising a robust deviance measure over a trimmed subset of the training data. Analogous to Alfons et al. (2013)'s treatment of sparse LTS for the linear regression case, a reweighting step is used to improve the estimator's efficiency. The authors provide an approximating algorithm to construct the optimal subset of training data, which is outlined in Section 3. In a simulation study where 5% of training examples are contaminated, they show that the predictive performance of their estimator is superior to that of classical logistic regression with elastic net regularisation. When the data are not contaminated, the authors show that performance of the classical and the reweighted robust estimators is comparable, indicating that the trimming procedure does not lead to a large loss of efficiency so long as the reweighting step is used.

Avella-Medina and Ronchetti (2017) take a different approach, building on Cantoni and Ronchetti (2001)'s robust quasi-likelihood estimator for the generalised linear model. The authors use the same quasi-likelihood criterion as Cantoni and Ronchetti (2001), but penalise it with a class of suitable functions to create an estimator that asymptotically possesses oracle properties as well as being robust against leverage points and outliers in the response. This class of penalty functions includes the lasso penalty, but also non-convex penalties such as that proposed in Fan and Li (2001). Throughout their paper, they focus on the adaptive lasso penalty, with the robust lasso estimator being used as an initial estimate. The authors develop a coordinate descent algorithm to compute the solution path of the initial estimate. In a simulation study that focusses on Poisson regression, the authors show that their estimator outperforms its classical counterpart in all contamination scenarios, though it becomes unstable when contamination exceeds 5% of the training examples. This reflects the fact that the robustness of the quasi-likelihood estimator is only local, in the neighbourhood of the model distribution.

Other work pertaining to robust regularised logistic regression is that of Sun et al. (2020). The authors connect recent theoretical results due to Ali and Tibshirani (2019) regarding the existence of the lasso-penalised maximum likelihood estimator to the notion of a breakdown point for logistic regression, arguing that earlier work on this topic (specifically that of Croux et al., 2002) does not extend to the case of lasso regularisation. We do not elaborate further on this discussion here since it has no implications for the current research. Sun et al. (2020) do this in the context of a lasso-penalised trimmed maximum likelihood estimator for logistic regression, which has an objective function is given by

$$\ell_{lasso}^{MTL}(\beta) = \sum_{l=1}^{h} d_l(\beta) + h\lambda \sum_{j=1}^{p} |\beta_j|, \tag{7}$$

where $d_1(\beta) \leq d_2(\beta) \leq \cdots \leq d_h(\beta)$ are the h smallest ordered deviances. The authors then derive the breakdown point of their estimator, which is defined in accordance with the results of Ali and Tibshirani (2019) and is increasing in the proportion of trimmed observations. An important drawback of the estimator proposed by Sun et al. (2020) is that it relies on ordinary likelihood which does not downweight outliers, such that it has an unbounded influence function. We do not include this estimator in our study, because it is essentially a less robust reformulation of the approach taken by Kurnaz et al. (2018b).

2.3 Cellwise Contamination and Robustness

An assumption made in the literature on robustness discussed thus far is that outliers are as specified by the Tukey-Huber contamination model (Maronna et al., 2019); the data is a mixture of ‘good’ observations generated from the model distribution and outlying observations which are generated from an arbitrary contamination distribution. Archetypically, the model characterises observations as being either completely outlying or not outlying at all. This abstraction may be unreasonable in the context of modern datasets, which tend to have many features and are often created by combining multiple data sources, not all of which may be reliable.

2.3.1 Cellwise Contamination

For the reasons mentioned above, it is likely that modern datasets have several observations that contain mostly ‘good’ cells, but also some outlying cells. This may not be because the observation as a whole does not match up with the rest of the data, but simply because the observations combine information from many different sources of varying quality. For example, in biomedical settings, one might imagine that an observation represents measurements of thousands of sensors or markers, each of which is imperfect and may make faulty measurements at random. Then, even if sensors have a small probability of making faulty measurements, a large proportion of observations will qualify as outliers because of the sheer number of variables[2]. This proportion may be greater than the breakdown point of even

[2] If cells have a probability ε of being outliers, a proportion $1-(1-\varepsilon)^p$ of observations will be outlying in expectation.

robust estimators, which is bound by 50% under the Tukey-Huber contamination model. New methods are thus needed to deal with cellwise outliers.

2.3.2 Cellwise Robust Two-Step Methods

The concept of cellwise outliers is formalised in the cellwise contamination model due to Alqallaf et al. (2009). However, because of the novelty of the cellwise contamination paradigm, research on estimators robust against cellwise outliers is nascent. In particular, estimators should be robust against cellwise as well as rowwise outliers, which has proven to be challenging (Öllerer et al., 2016). To overcome these difficulties, some authors suggest handling cellwise and rowwise outliers as separate issues, which gives rise to two-step procedures that address the two sequentially (Leung et al., 2016).

One such two-step procedure for handling the case where there are both cellwise and rowwise outliers was suggested by Rousseeuw and Van Den Bossche (2018). The authors introduce the DetectDeviatingCells (DDC) algorithm, which uses pairwise regressions among the predictors to come up with an expected value for each cell in the data matrix. When a cell deviates too much from its expected value, it is flagged as an outlier and a 'corrected' value must be imputed. Pairwise regressions are used because they avoid problems caused by the dimensionality of the dataset (which inspired the cellwise contamination paradigm), while simultaneously preserving part of the correlations among variables to more accurately determine what exactly constitutes an outlier. The theoretical justification of this first step is given by Rousseeuw and Van Den Bossche (2018). After outlying cells have been corrected, the data should satisfy the rowwise contamination model. The second step therefore entails using existing robust regression methods on the corrected dataset.

2.3.3 Rowwise and Cellwise Robust Regression

An argument against two-step procedures such as that of Rousseeuw and Van Den Bossche (2018) is made by Filzmoser et al. (2020). They argue that whether a cell is outlying is determined by the model used. In that case, the pre-processing of data, as in two-step procedures, is likely to be inconsistent with the model. Instead, they advocate one-step procedures which tackle cellwise and rowwise in a single, model-consistent way. To this end, the authors propose the cellwise robust M-regression estimator for the low-dimensional ($n \gg p$) linear regression case, which generalises the MM-estimator of Yohai (1987) to the setting with cellwise and rowwise outliers. They develop an iteratively reweighted least squares (IRLS) algorithm which detects and imputes outlying cells as part of the fitting process. To detect outlying cells, they use the Sparse Directions of Maximal Outlyingness algorithm of Debruyne et al. (2019). Starting from a (rowwise) robust regression estimator such as the MM-estimator, the IRLS algorithm iterates between detecting and imputing outlying cells of 'mostly good' observations, downweighting outlying observations, and updating the regression parameter estimates. This procedure is repeated until the regression estimates converge.

The estimator of Filzmoser et al. (2020) could theoretically be extended to the high-dimensional and potentially even logistic regression setting, but developing and implementing such methods is beyond the scope of this paper. Instead, we consider a more broadly applicable cellwise and rowwise robust regression procedure that was suggested by Machkour et al. (2020). The authors develop an adaptive lasso estimator for high-dimensional linear regression which robustifies the MM-estimator against cellwise outliers by making the adaptive weights a function of predictor outlyingness. In this way, the adaptive weights w_j seen in Equation (3) are replaced by weights of the form $w_j = |z_j \times \hat{\beta}_{init,j}|^{-1}$, where z_j measures the outlyingness of predictor j. This approach is conceptually different from that of Filzmoser et al. (2020). Whereas the procedure of Filzmoser et al. (2020) explicitly addresses the outlyingness of each cell and corrects individual cells where necessary, Machkour et al. (2020) effectively treat cellwise outlyingness as a characteristic of the entire predictor and downweights predictors accordingly. This is because the sole goal of the former authors is to robustify the MM-estimator against cellwise contamination, whereas the latter seek to simultaneously robustify the variable selection of lasso and extend the MM-estimator to the case of cellwise contamination. Machkour et al. (2020) propose measuring predictor outlyingness using the Adjusted Stahel-Donoho Outlyingness of Van Aelst et al. (2011). They first create an outlyingness matrix, which measures the outlyingness $r_{i,j}$ of each cell $X_{i,j}$ as a weighted average of the outlyingness of observation i and predictor j. For a given predictor j, outlyingness measure z_j is then computed by aggregating the $r_{i,j}$ across all i.

An advantage of the approach of Machkour et al. (2020) is that the predictor weights used to robustify the lasso regression estimator are not dependent on the choice of regression estimator. The authors apply their methodology to the MM-estimator because of its theoretical and empirical performance in low and high dimensions (Smucler and Yohai, 2017), but the computation of the adaptive weights is not intrinsically linked to the estimator. Imperatively, the procedure is not even bound to the linear regression setting and can be used in any estimator that is regularised using the adaptive lasso or elastic net. For this reason, we include the procedure in our analysis.

2.4 Contribution of this Research

Logistic regression often serves as a benchmark against which classifiers are compared (Sur and Candès, 2019). Yet, Sections 2.2 and 2.3 indicate that literature on regularised robust regression is nascent, especially outside of the linear setting. Avella-Medina and Ronchetti (2017) and Kurnaz et al. (2018b) develop estimators that exist in high dimensions and have bounded influence functions. The authors compare the performance of their estimators to their classical counterparts, but not to each other's estimators. Further, their simulations studies are completely focussed on out-of-sample classification accuracy and consistency in variable selection and do not address probabilistic forecasting performance. To the best of our knowledge, the literature generally provides no evidence on the impact of variable selection and shrinkage on the forecasting performance of logistic regression, either in the absence or in the presence of contamination. Similarly, while Hastie et al. (2017) compare the predictive accuracy of least squares linear regression with best subset selection and lasso, there are, to the best of our knowledge, no comparable studies investigating which feature selection method is preferable for probabilistic forecasting.

This study contributes to the literature by answering the questions posited above, which are especially interesting in the context of contaminated and high-dimensional data. She and Owen (2011) and Donoho and Montanari (2016) investigate the connection between M-estimators and penalised least squares estimators in the context of linear regression. They show that if the objective functions are suitably formulated, the two approaches yield the same solution. Avella-Medina and Ronchetti (2015) note that this finding has important implications, as it means that developments in sparse modelling regarding non-asymptotic theory and optimisation algorithms might be directly applicable to robust statistics. Pertaining to the study at hand, it is worthwhile to investigate whether connections between sparse regression and robust regression methods are reflected in the methods' forecasting performance when the data are contaminated.

3 Methodology

In this section we provide implementation details of the methods used in the simulation and real data studies. We discuss the regularised and robust logistic regression (RRLR) estimators introduced in the previous section in detail and contrast these with the non-robust benchmark estimators. Further, we give brief overviews of the machine learning methods that are included in the study for comparison.

3.1 Robust and Sparse Logistic Regression

As outlined in Section 2.2, Kurnaz et al. (2018b) and Avella-Medina and Ronchetti (2017) each propose a RRLR estimator. In this section, we discuss their estimators in turn and compare them, starting with the approach of the former authors. We begin with the deviances

$$d_i(\beta) = -y_i(X_i'\beta) + log\,(1 + e^{X_i'\beta}), \tag{8}$$

which are just components of the log-likelihood in Equation (1). If we consider the terms $[y_i - \sigma(X_i'\beta)]$, $y_i \in \{0,1\}$ as residuals, then the $d_i(\beta)$ are essentially logarithms of these residuals. Analogously to LTS for the linear regression setting, we may thus obtain a robust version of logistic regression by trimming these deviances to construct a subset of observations of size h.

A problem with the deviances in Equation (8) is that the function is highly influenced by bad leverage points, which in this context are observations for which we observe a strongly positive score $(X_i'\beta)$ but $y_i = 0$, or a strongly negative score but $y_i = 1$. If we make the predictors arbitrarily anomalous, then $d_i(\beta) \to +\infty$ unless $\beta = 0$. Robust logistic regression therefore requires a robust deviance function. Pregibon (1981) proposed a class of estimators in which $d_i(\beta)$ are replaced by $\rho(d_i(\beta))$, for a function $\rho(\cdot)$ which increases slower than the identity function. Bianco and Yohai (1996) showed that Pregibon (1981)'s proposal was not Fisher consistent and suggested an adjusted, consistent alternative estimator based on an alternative deviance measure $\varphi_{BY}(X_i'\beta; y_i)$. Croux and Haesbroeck (2003) improved this work further, defining a choice for $\rho(\cdot)$ that ensures that the robust estimator exists whenever the maximum likelihood estimator exists. They also proposed a fast algorithm to compute the resulting estimator. For our purposes, it is sufficient to state that the equation is zero in expectation, such that the estimator is still Fisher consistent.

Croux and Haesbroeck (2003) further derive the influence function of their estimator and show that it is unbounded, meaning the bad leverage points discussed earlier may still have an undue effect on the estimator. The authors propose to overcome this issue by downweighting high-leverage observations, where leverage is determined using the squared Mahalanobis distance

$$M_i = (X_i - \bar{X})' S^{-1} (X_i - \bar{X}).$$

Here, $\bar{X}, S$ are robust estimates of location and scatter obtained using the minimum covariance determinant estimator (Rousseeuw, 1985). As a weighting scheme, Croux and Haesbroeck (2003) use a hard rejection rule of the form

$$W(M_i) = \begin{cases} 1, & \text{if } M_i \leq \chi_p^2(.975), \\ 0, & \text{else,} \end{cases}$$

where $\chi_p^2(\cdot)$ denotes the corresponding quantile of the χ^2 distribution with p degrees of freedom. Using this approach, 2.5% of observations are expected to be flagged as outliers under the model distribution.

Kurnaz et al. (2018b) use the deviance measure $\varphi_{BY}(X_i'\beta; y_i)$ to define a robust objective function with regularisation

$$Q(H, \beta) = \sum_{i \in H} \varphi_{BY}(X_i'\beta; y_i) + h\lambda \left(\alpha \sum_{j=1}^{p} |\beta_j| + \frac{(1-\alpha)}{2} \sum_{j=1}^{p} \beta_j^2 \right). \tag{9}$$

In Equation (9), H denotes the subset of $h \leq n$ observations to be used in the fitting procedure. For given choice of h, we wish to use the optimal subset $H_{h,opt}$ which minimises the sum of deviances. Finding $H_{h,opt}$ is not trivial, however, as it entails a combinatorial problem. Kurnaz et al. (2018b) propose an approximating algorithm much like that used in Alfons et al. (2013) for linear regression, but altered to account for the fact that binary logistic regression uses two distinct classes with a given class balance. α and λ are tuned using a grid search, where the parameters are chosen such that the mean deviance

$$\bar{d}(\alpha, \lambda) = \frac{1}{h} \sum_{i \in H_{h,\alpha,\lambda}} \varphi_{BY}(X_i'\hat{\beta}_{\alpha,\lambda}; y_i)$$

is minimised, with $H_{h,\alpha,\lambda}$ the best subset of size h obtained with given values for α, λ. $\bar{d}(\alpha, \lambda)$ is computed by 5-fold CV, a procedure that is carried out twice for stability in case $H_{h,\alpha,\lambda}$ is still contaminated. See Kurnaz et al. (2018b) for an outline of the full procedure.

As in Alfons et al. (2013), the authors use a reweighting step to improve efficiency of the estimator. In doing so, they deviate from the rejection rule based on Mahalanobis distance as proposed by Croux and Haesbroeck (2003) and instead use Pearson residuals

$$r_i = \frac{y_i - \pi_i}{\sqrt{\pi_i(1-\pi_i)}}, \tag{10}$$

with $\pi_i = \sigma(X_i'\beta)$. The weighting scheme is then based on the hard rejection rule

$$W(r_i) = \begin{cases} 1, & \text{if } |r_i| \leq \Phi^{-1}(.9875), \\ 0, & \text{else}, \end{cases} \tag{11}$$

where $|r_i|$ is the absolute value of r_i and $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution.

The authors have implemented their estimator in the **R** package `enetLTS` (Kurnaz et al., 2018a). We do not make any changes to their code and call the `enetLTS` fitting method with default parameters. We exclusively consider the reweighted estimator in our analysis, as it performs uniformly better in the authors' simulations. The authors call their estimator enetLTS (even in the context of logistic regression), but we refer to it as the Bianco-Yohai estimator with elastic net regularisation (BY-EN)[3] throughout this paper to avoid confusing the estimator with the package that implements it. As a non-robust baseline against which we can evaluate the performance of BY-EN, we use classical logistic regression with an elastic net penalty.

The approach of Avella-Medina and Ronchetti (2017) extends the work of Cantoni and Ronchetti (2001) on robust quasi-likelihood to the high-dimensional setting. We first summarise the robust quasi-likelihood approach, starting from the quasi-likelihood estimators for the generalised linear model of Wedderburn (1974). These estimators are defined implicitly by the estimating equations

$$\sum_{i=1}^{n} \frac{y_i - \mu_i}{\sqrt{V(\mu_i)}} \mu_i x_i = 0, \tag{12}$$

where μ_i and $V(\mu_i)$ denote the expectation and variance of $y_i|x_i$. Outliers in the response and leverage points affect the estimators through y_i and x_i respectively, so to control the influence of contaminated data Cantoni and Ronchetti (2001) apply weighting functions. The estimating equations become

$$\sum_{i=1}^{n} \left[\frac{\psi(r_i)}{\sqrt{V(\mu_i)}} \frac{\partial \mu_i}{\partial \beta} w(x_i) - a(\beta) \right] = 0, \tag{13}$$

where r_i are Pearson residuals, which in are given by Equation (10) in case of logistic regres-

[3] For the remainder of the paper, BY-EN denotes the reweighted version of the estimator unless explicitly stated otherwise.

sion. Further, $a(\beta)$ is shorthand for a term that ensures the equality holds in expectation, i.e. Fisher consistency. The weighting functions are given by $\psi(\cdot)$ and $w(\cdot)$, where the former is a bounded function such as the Huber function

$$\psi(r_i) = \begin{cases} r_i, & \text{if } |r_i| \leq c, \\ c\,\mathrm{sgn}(r_i), & \text{else,} \end{cases} \tag{14}$$

for some constant $c > 0$.

For a full specification of the quasi-likelihood function from which these estimating equations are derived we refer to Cantoni and Ronchetti (2001) or Avella-Medina and Ronchetti (2017). Here, we will simply denote the function by $\rho(\beta)$, whose robustness properties are governed by $\psi(\cdot)$ and $w(\cdot)$. The authors extend $\rho(\beta)$ with the adaptive lasso penalty introduced by Zou (2006). As before, coefficient weights are determined as $w_j = |\hat{\beta}_{init,j}|^{-1}$. The initial estimate $\hat{\beta}_{init,j}$ is the lasso-penalised robust quasi-likelihood solution. To compute the solution path, the authors develop a coordinate descent algorithm, which represents a robust alternative to the IRLS algorithm used for the generalised linear model. The robustness of the algorithm is ensured by the use of 'pseudo-data' vector z which replaces the response, as well as observation weighting matrix W. For a complete specification of these weights and pseudo-data we refer to Avella-Medina and Ronchetti (2017). The algorithm first selects the optimal value of tuning parameter λ based on a robust information criterion. Subsequently, the algorithm updates the pseudo-data and observations weights, after which it finally fits parameters β based on the updated penalty term, weights and data. This procedure is repeated until convergence. The initial value of tuning parameter λ is proportional to the max-norm of the product $WX'z$.

The authors provide an **R** implementation of their estimator for the cases of Poisson and logistic regression in the Supplementary Material of their paper, which we use in our analysis. The authors give the user the option to choose let weighting function $\psi(\cdot)$ be given by the Huber function, Tukey's biweight function or the (non-robust) identity function. In the latter case, 'ordinary' likelihood is used and the algorithm of Avella-Medina and Ronchetti (2017) reduces to the cyclical coordinate descent algorithm of Friedman et al. (2010). Throughout our analysis, we use the Huber function in line with Avella-Medina and Ronchetti (2017)[4]. We make some changes to the code to prevent bugs that the source code produced during

[4] Another reason to choose the Huber function is that the fitting procedure consistently fails to converge for logistic regression when using Tukey's biweight. Though the authors only consider Poisson regression in their paper, they ostensibly encountered similar numerical problems. In the Supplementary Material, the authors note that their algorithm was 'not satisfactory' for Tukey's biweight function and yielded 'a very erratic solution path'.

simulations, but note that the numerical stability of the code is still not guaranteed[5]. As a non-robust baseline against which to compare the performance of the robust estimator, we replace the Huber function with the identity function.

3.2 Non-robust Sparse Logistic Regression

As explained in Section 2, we wish to investigate how the elastic net penalty compares to best subset selection in terms of probabilistic forecasting performance. To this end, we include four regularised classical logistic regression (RCLR) estimators in the analysis. The first two are classical logistic regression with elastic net penalty (CLR-EN) and with the adaptive elastic net (CLR-AdaEN). We implement the two using the glmnet package. Hyperparameters λ and α are chosen by cross-validation each time the models are trained, with deviance (Equation (8)) as loss function. Candidate values for α are the same as those used by enetLTS to ensure BY-EN can be compared with CLR-EN fairly. Candidate values for λ are automatically chosen by glmnet, which is also the procedure that enetLTS follows for the reweighted estimator (i.e. enetLTS imports and calls the `cv.glmnet()` function for this step). The adaptive elastic net uses the elastic net solution as initial estimate.

The other two methods are variants of best subset selection. The first method, classical regression with BSS (CLR-BSS), uses only the 'ℓ_0-norm' penalty as in Equation (5). The second, classical regression with BSS and lasso penalty (CLR-BSSL), further adds shrinkage and effectively implements Equation (6) with $q = 1$. These methods are implemented in the L0Learn package (Hazimeh and Mazumder, 2021). Tuning parameters λ_0 and λ_q are automatically selected by fitting method `L0Learn.cvfit()`, which we call with default parameters. One exception is that we set the largest permitted model size k equal to p to avoid biasing the fitting procedures with our knowledge of the true sparsity in the simulations.

3.3 Machine Learning Methods

3.3.1 Support Vector Machines

We add support vector machines (SVM) with a radial basis function (RBF) kernel $K(x, x') = \exp(-\gamma\|x - x'\|^2)$ as a classifier. We choose SVM with RBF kernel on account of the method's ability to deal with high-dimensional data (Hastie et al., 2009). Further, the kernel is commonly employed in practice (Meyer et al., 2021). Though the SVM's hinge loss implies

[5] The code primarily suffers from issues when evaluating derivative $\partial\mu_i/\partial\eta_i$, which is used to compute z and W. The derivative entails multiplying and dividing by fitted probabilities π_i, some of which become numerically zero or one during the fitting process. To resolve some of the issues, we used logarithmic transformations and constrained π_i to be in the interval $[10^{-8}, 1 - 10^{-8}]$. This reduced the number of errors that occurs, but the estimator still performs poorly.

that it has an unbounded influence function (Yang et al., 2010), the regularisation term of the soft-margin SVM should improve its robustness similarly to the improved robustness of penalised least squares.

We optimise the cost of constraint violation C and the RBF kernel parameter γ by cross-validation each time we train the model. Herein, we consider values $\{0.01, 1, 10, 25, 50, 100\}$ for C and $\{\frac{1}{2p}, \frac{1}{p}, \frac{2}{p}\}$ for γ. This choice of values is mostly as suggested by the authors of the e1071 package (Meyer et al., 2021) that implements SVM in **R**.

It must be noted that the SVM is not naturally a probabilistic classifier and is therefore unsuited for the task of probabilistic forecasting if implemented naively. A method to transform the output of a support vector into a posterior distribution over the classes was introduced by Platt (1999) and is known as Platt scaling. We prefer Platt scaling over the isotonic regression method of Zadrozny and Elkan (2002) per the study of Niculescu-Mizil and Caruana (2005), who conclude that Platt scaling performs better when the posterior distribution is sigmoidal and in small sample sizes, both of which apply in our study. As explained in Section 4.1, we use small sample sizes throughout this paper to control the computational burden of the robust methods.

Platt scaling entails fitting a logistic regression model on the output of the SVM. We reserve part of the training data as a validation set to be used for this procedure, such that the logistic regression is not trained on the same data as the SVM. The reason for this is that fitting the logistic regression model on the same data that was used to train the SVM leads to overfitting (Platt, 1999). Note that fitting the logistic regression model is feasible regardless of p, as the model contains only two parameters, an intercept and a slope for the output of the unscaled SVM. Nonetheless, the number of observations to be reserved for scaling is a precarious choice when the training set is small. If too few observations are used, the logistic regression model fit will be poor and the posterior probabilities will be inaccurate, even if the SVM could have produced accurate forecasts. If too many are used, few observations are left for training the SVM. However, this is an inherent drawback of using an SVM for probabilistic forecasting.

To make the fitting procedure of the SVM as favourable as possible, we create a single validation set from the training data that is used for hyperparameter tuning as well as Platt scaling. This maximises the number of observations used for hyperparameter tuning without leaking information of the validation set to the model. For each set of hyperparameters, we train the SVM on the training set and estimate predictive accuracy based on the validation set. We then select the optimal hyperparameter configuration and perform Platt scaling on the same validation set. Although the SVM uses the validation set for prediction prior to Platt scaling, it is never trained on it.

3.3.2 Neural Networks

In addition to SVM we also include a feed-forward neural network (multilayer perceptron) in our analysis. Unlike SVM, a neural network with sigmoid non-linearity at the output layer is naturally a probabilistic classifier and therefore does not require calibration. Niculescu-Mizil and Caruana (2005) show that neural networks without calibration produce some of the most accurate posterior probabilities and are competitive with calibrated SVM.

We create a network with a single hidden layer using the RSNNS package (Bergmeir and Benítez, 2019). The number of units in the hidden layer is a hyperparameter and is chosen by cross validation. We consider $\frac{1}{3}p$, $\frac{2}{3}p$ and p units, which is mostly in line with the rules of thumb outlined in Heaton (2008). The sigmoid function is chosen as activation function for all hidden units. Weights are initialised randomly and are updated by backpropagation, using cross-entropy as a loss function. To prevent overfitting we apply weight decay, which is equivalent to ridge regression for linear models (Hastie et al., 2009). As in ridge regression, a tuning parameter λ must be chosen which determines how much shrinkage is applied. We choose the optimal value among $0, 0.001, 0.01, 0.1$ per the guidelines in Kuhn and Johnson (2013).

3.4 Evaluation of Probabilistic Forecasting Performance

We are primarily interested in probabilistic forecasting performance, and therefore consider the calibration (consistency) and sharpness (efficiency) of the forecasts. Generally, the goal of probabilistic forecasting is to maximise sharpness subject to calibration (Gneiting and Katzfuss, 2014). In our setting with a binary outcome, calibration may simply be investigated by means of a calibration diagram. In a calibration diagram, we divide posterior probabilities (usually for class $y = 1$) into q bins of width $1/q$. We then plot the observed fraction of observations with $y = 1$ in a bin against the predicted probabilities of those observations. If predictions are calibrated, then they should be indistinguishable from random draws from the true posterior distribution (Gneiting and Katzfuss, 2014). Visually, this means the graph should follow the 45-degree line; predicted probabilities should coincide with empirical probabilities. For example, if we set $q = 20$ and our forecasts are accurate, approximately half of the observations with predicted class probabilities in the 50-54% bin should have $y_i = 1$. We set $q = 20$ throughout the paper, balancing the granularity of the bins with the number of test observations available to accurately estimate the fraction of forecasts assigned to each bin.

Sharpness of forecasts refers to the concentration of the predictive distribution. When choosing between competing methods that are all calibrated, we prefer the method that produces the sharpest (most concentrated) forecasts, as these are the forecasts in which we are most confident. Sharpness is evaluated by means of metrics called scoring rules, which may be understood as being equivalent to loss functions in regression and classification tasks. As with loss functions, an infinite number of scoring rules exists. To choose between them, the literature encourages the use of *proper* scoring rules. Proper scoring rules ensure that quoting the true posterior distribution as the forecast distribution is optimal (in terms of 'smallest loss') in expectation (Gneiting and Raftery, 2007). Essentially, proper scoring rules encourage truth-telling when making forecasts (Gneiting and Katzfuss, 2014). Among a class of proper scoring rules (with certain 'loss', e.g. quadratic or entropy), a scoring rule is further said to be *strictly proper* if it assigns the best possible score only to a forecast which exactly coincides with the realised value. Formally, if $S(F, G)$ denotes the expected score obtained with given forecast F when the true value is G, then S is proper if

$$S(G, G) \leq S(F, G), \tag{15}$$

where we implicitly assume that the objective is to minimise the score. Further, S is strictly proper if the inequality in Equation (15) is strict whenever $F \neq G$ (Gneiting and Katzfuss, 2014).

We employ the Brier score, a scoring rule which is proper for predictions of categorical variables (Gneiting and Raftery, 2007). Other proper (strictly) scoring rules for categorical variables are available (e.g. logarithmic score), but the main findings of this paper do not depend on the scoring rule used. For the sake of brevity, we therefore only consider the Brier score. Given a set of forecasted probabilities $\hat{p}$ corresponding to m test set outcomes y, the Brier score is computed as

$$BS(y, \hat{p}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{p}_i)^2. \tag{16}$$

The Brier score ranges from 0 to 1, with a lower score indicating greater sharpness. In particular, the Brier score is 0 whenever $y_i = \hat{p}_i = 0$ or $y_i = \hat{p}_i = 1$. This means the rule is strictly proper in our setting, as no other forecast than 0 or 1 can attain a score of 0.

If class $y = 1$ occurs with frequency f in the test data, guessing $y = 1$ with probability f has a Brier score of $f - f^2$ in expectation. Equation (16) further shows that the Brier score is symmetric, in the sense that an observation j with $y_j = 1$ and $\hat{p}_j = 0.3$ inflates the Brier score by the same amount as an observation k with $y_k = 0$ and $\hat{p}_k = 0.7$. For some forecasting settings this property may be undesirable. For example, in medical applications one may prefer an asymmetric scoring rule that penalises poor forecasts of positive tests more heavily than poor forecasts of negative tests.

4 Simulation Study

In this section we study the probabilistic forecasting performance of the methods discussed in Section 3 in a controlled setting. We first discuss the simulation setup, after which we sequentially present the results of the scenarios studied.

4.1 Setup

We consider primarily two scenarios. In the first scenario, the dimensionality $p = 50$ is relatively large compared to, but not larger than the training sample size $n = 150$. The scenario investigates the importance of regularisation for probabilistic forecasting, when we step outside of the classical paradigm where we have many observations of a few variables. In the second scenario, the data is high-dimensional and the dimensionality $p = 100$ is larger than the training sample size $n = 50$. This scenario is arguably more challenging and allows us to inspect if calibrated probabilistic forecasting is feasible with high-dimensional data. The two scenarios are comparable to the simulation studies in e.g. Maronna (2011) and Kurnaz et al. (2018b).

In both scenarios, β is highly sparse, with only 10% of its entries being non-zero. For simplicity, we set set all non-zero entries to 0.3, i.e. $\beta_{\mathcal{A}} = [0.3, 0.3, \ldots, 0.3]'$. A value of 0.3 helps to control the distribution of the class probabilities, especially in the scenario where $p = 100$. For values greater than 0.3, most probabilities $\sigma(X\beta)$ become numerically close to zero or one, even if the predictors in $\mathcal{A}$ have unit variance. Such an extreme distribution where is arguably uninteresting for probabilistic forecasting and hence we wish to avoid it.

Design matrix X consists of two distinct submatrices $X_{\mathcal{A}}$ and $X_{\mathcal{A}^c}$, which respectively correspond to the non-zero and zero elements of β. $X_{\mathcal{A}}$ is drawn from a zero-mean multivariate distribution with covariance matrix $\Sigma_{\mathcal{A}}$, whose (i, j)-th entry $\Sigma_{\mathcal{A},(i,j)}$ is given by $0.9^{|i-j|}$. The informative variables are therefore highly correlated, warranting the use of the elastic net penalty described in Section 2.1.3. The uninformative variables in $X_{\mathcal{A}^c}$ are drawn from another zero-mean multvariate normal distribution with covariance matrix $\Sigma_{\mathcal{A}^c}$, which has entries $\Sigma_{\mathcal{A}^c,(i,j)} = 0.5^{|i-j|}$[6]. The informative variables are drawn independently of the uninformative variables. We then construct $X = [X_{\mathcal{A}}, X_{\mathcal{A}^c}]$ and sample the outcomes y_i from Bernoulli distributions with probability $\mathbb{P}(y_i = 1|X_i = x) = (1 + e^{-x'\beta})^{-1}$. X_i is symmetric around zero, so the classes are balanced in expectation.

For both choices of p, we consider a setting where there is no contamination as well as a setting where design X is contaminated. Here, we follow the procedure of Kurnaz et al.

[6] Further testing with $\Sigma_{\mathcal{A}^c,(i,j)} = 0.0^{|i-j|}$ showed the correlation between uninformative variables is not consequential for the performance of the estimators.

(2018b), who select the first $\lfloor 0.1n \rfloor$ observations for which $y = 0$ and shift the mean of the corresponding informative variables (but not the uninformative variables) by a value of 20. Effectively, this means that the $X_{\mathcal{A}}$ of the contaminated observations are drawn from a $N(\mathbf{20}, \Sigma_{\mathcal{A}})$ distribution, where we denote $\mathbf{20} = [20, 20, \ldots, 20]'$. The contamination procedure implies that the data contains approximately 5% bad leverage points, since the contaminated observations have a large, positive score but $y = 0$. We repeat each scenario $R = 100$ times, balancing computational burden of the robust estimators with accuracy. The runtime of a scenario is 10-12 hours on an Intel Xeon @ 2.00 GHz × 8 processors and, based on experimentation, we are fully confident that the results do not change when $R > 100$.

4.2 Results when $p < n$, but p/n is large

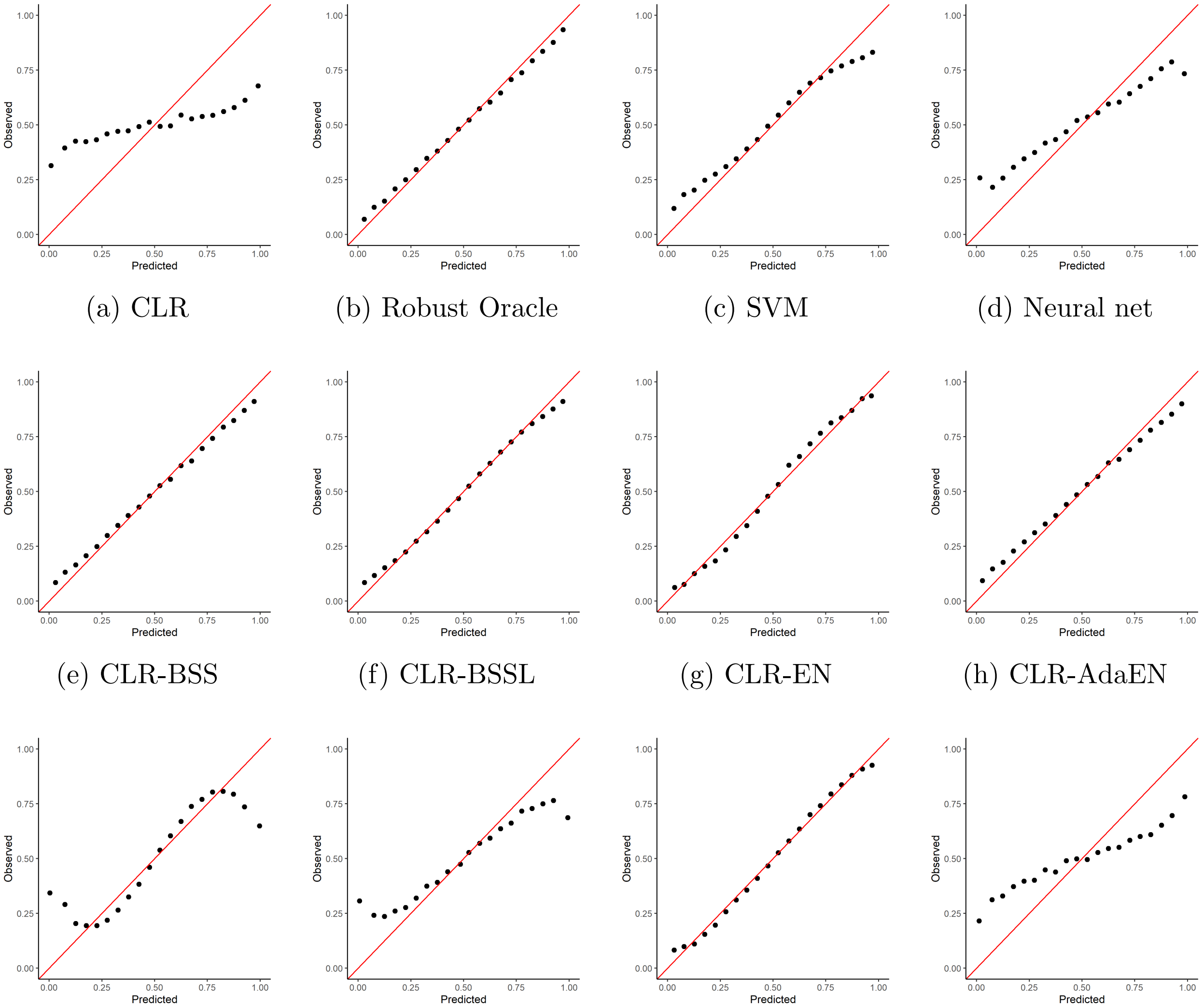
4.2.1 No contamination

We start with the setting $p = 50$ and $n = 150$. Figure 1 shows that when there is no contamination, several methods produce calibrated forecasts. Among the CLR estimators, the non-regularised estimator is the only method that is distinctly uncalibrated. The data is too noisy, and the non-regularised estimator overfits. Variable selection as performed by best subset selection or the elastic net mostly resolves this problem, though in the case of the non-adaptive elastic net this comes at the cost of an attenuation bias due to shrinkage.

	β_1	β_2	β_3	β_4	β_5	FPR($\mathcal{A}^c$)
CLR with elastic net	0.20	0.20	0.23	0.24	0.18	0.20
CLR with adaptive elastic net	0.30	0.28	0.31	0.35	0.25	0.20
CLR with best subset selection	0.16	0.33	0.46	0.36	0.11	0.01
CLR with best subset selection and lasso	0.18	0.29	0.34	0.30	0.15	0.04
Robust oracle	0.36	0.29	0.27	0.32	0.37	
Ground truth	0.30	0.30	0.30	0.30	0.30	

Table 1: Mean estimates of parameters corresponding to the active set and mean false positive rate for including uninformative variables in the model. CLR is classical logistic regression. $p = 50$, $n = 150$ and there is no contamination.

Table 1 exhibits mean parameter estimates produced by the RCLR estimators. The CLR-EN estimator visibly pulls the parameter estimates $\hat{\beta}$ to zero, which pulls forecasts $\sigma(X\hat{\beta})$ to $\sigma(0) = 50\%$. This explains the subtle sigmoidal pattern in the corresponding calibration plot in Figure 1. The calibration plot stays slightly below the 45-degree line for predicted probabilities below 50%, because the forecasting distribution of the elastic net has thin tails compared to the true posterior distribution. As a result, few observations are assigned to bins of predicted probabilities such as [0, 5%) and [5%, 10%). For predicted probabilities greater than 50%, the calibration plot stays above the 45-degree line because the forecasting distribution is more dense around the mean of 50% than the true posterior distribution. Too many observations are assigned to bins such as [45%, 50%) and [50%, 55%).



(a) CLR

(b) Robust Oracle

(c) SVM

(d) Neural net

(e) CLR-BSS

(f) CLR-BSSL

(g) CLR-EN

(h) CLR-AdaEN

(i) CR-Lasso (Huber)

(j) CR-Adalasso (Huber)

(k) CR-Lasso (Identity)

(l) BY-EN

Figure 1: Calibration plots for the $p < n$ setting with 5% bad leverage points, all in the $y = 0$ class. Results are obtained across 100 simulation runs with 1000 test observations each. CLR, CR and BY respectively denote classical logistic regression, the Cantoni-Ronchetti estimator and the Bianco-Yohai estimator. (Ada-) EN and BSS (-L) stand for (adaptive) elastic net penalty and best subset selection (with lasso penalty), while Huber and Identity respectively indicate that robust quasi-likelihood with the Huber function or non-robust likelihood was used.

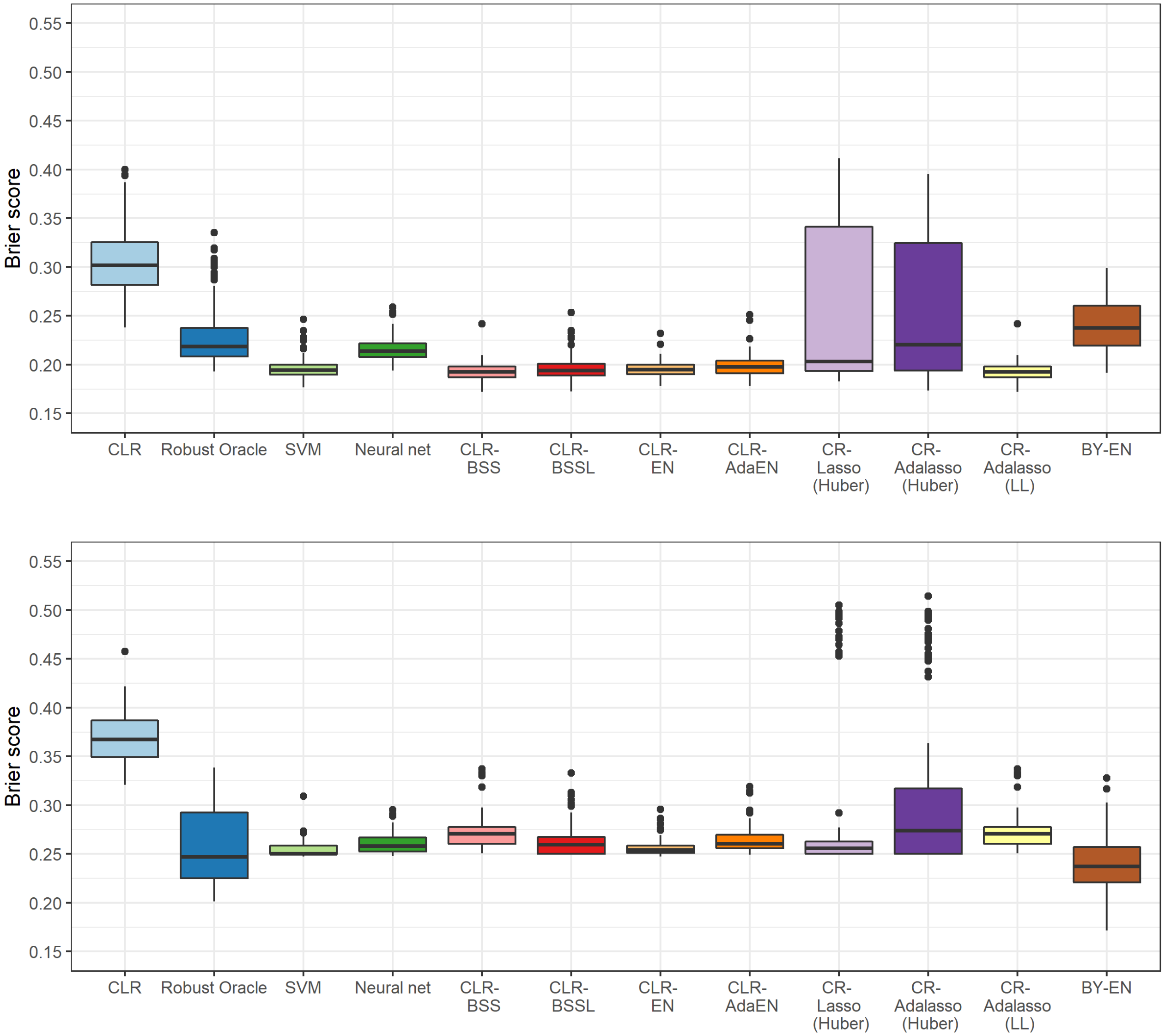
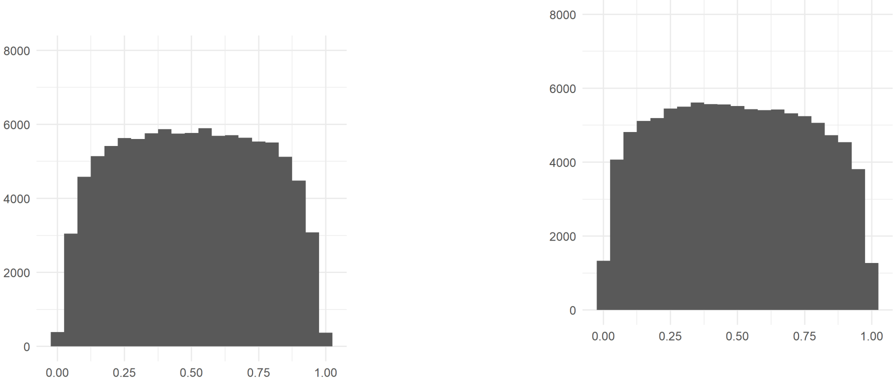


Figure 2: Brier scores for $p = 50, n = 150$. Top: without contamination. Bottom: with 5% bad leverage points, all in class $y = 0$. Results are obtained across 100 simulation runs with 1000 test observations each. In both plots, the vertical axis is truncated at 0.35, thereby removing the largest scores for the logit. Except for the support vector machine (SVM), neural net and the 3 rightmost estimators, all estimators are variants of classical logistic regression (CLR). (Lasso-) S* and (Ada-) EN respectively denote CLR with best subset selection (and lasso penalty) and CLR with (adaptive) elastic net penalty. The 3 rightmost estimators are the Cantoni-Ronchetti (CR) estimator with lasso penalty, CR estimator with adaptive lasso penalty and the Bianco-Yianco estimator with elastic net penalty.

Table 1 further shows that the parameter bias is mostly overcome by CLR-AdaEN, which produces mean parameter estimates that are more accurate those of the robust oracle in terms of l_2 loss $\|\beta - \hat{\beta}\|_2$. This is desirable from the viewpoint of statistical inference on those variables, though the calibration plot of CLR-AdaEN actually exhibits slightly greater deviations from the 45-degree line than the robust oracle. As can be seen in Figure 3, this is because the posterior distribution as estimated by CLR-AdaEN penalty is too dense at the tails. When the forecasting distribution has fat tails, too many observations are assigned to the smallest and largest predicted probability bins. We speculate this is a result of the high false positive rate with which CLR-EN and CLR-AdaEN include uninformative variables in the model, as shown by Table 1. If too many uninformative variables are assigned non-zero coefficients, estimated class probabilities may become inflated. Best subset selection performs much better in this regard and producer sparser models with a near-zero false positive rate for the uninformative variables. However, this comes at the cost of far less accurate parameter estimates for the informative variables.



(a) True posterior

(b) Fitted posterior CLR-AdaEN

Figure 3: Comparison of the true posterior distribution in the uncontaminated $p < n$ setting and the estimates produced by classical logistic regression with the adaptive elastic net (CLR-AdaEN). The horizontal axis shows the posterior probabilities. The vertical axis shows the number of times a probability occurs and is measured on the same scale in the graphs.

Overall, all RCLR estimators perform comparably. All methods result in reasonably calibrated forecasts and are competitive with the robust oracle in this regard. In terms of forecast sharpness, the top panel of Figure 2 shows that again none of the methods is preferable, with all achieving a median Brier score of just under 0.20. This makes the estimators more efficient that the robust oracle, which has a median Brier score is approximately 0.24.

The RRLR methods perform poorly compared to their non-robust counterparts. We expected a loss of efficiency was expected at the model distribution, but not such such a large difference in calibration. All three versions of the Cantoni-Ronchetti (CR) estimators as well as BY-EN are completely uncalibrated. It appears the robust methods do not apply enough shrinkage, though in the case of the CR estimators we note that numerical convergence issues occurred during the simulations. This may indicate that its poor performance is attributable to numerical properties, as opposed to theoretical properties of the estimator. Nonetheless, all RRLR methods produce poor parameter estimates for the active set and have high false positive rates for the uninformative variables. Though the adaptive step improves the performance of the CR estimator in both aspects, the estimator still applies insufficient shrinkage.

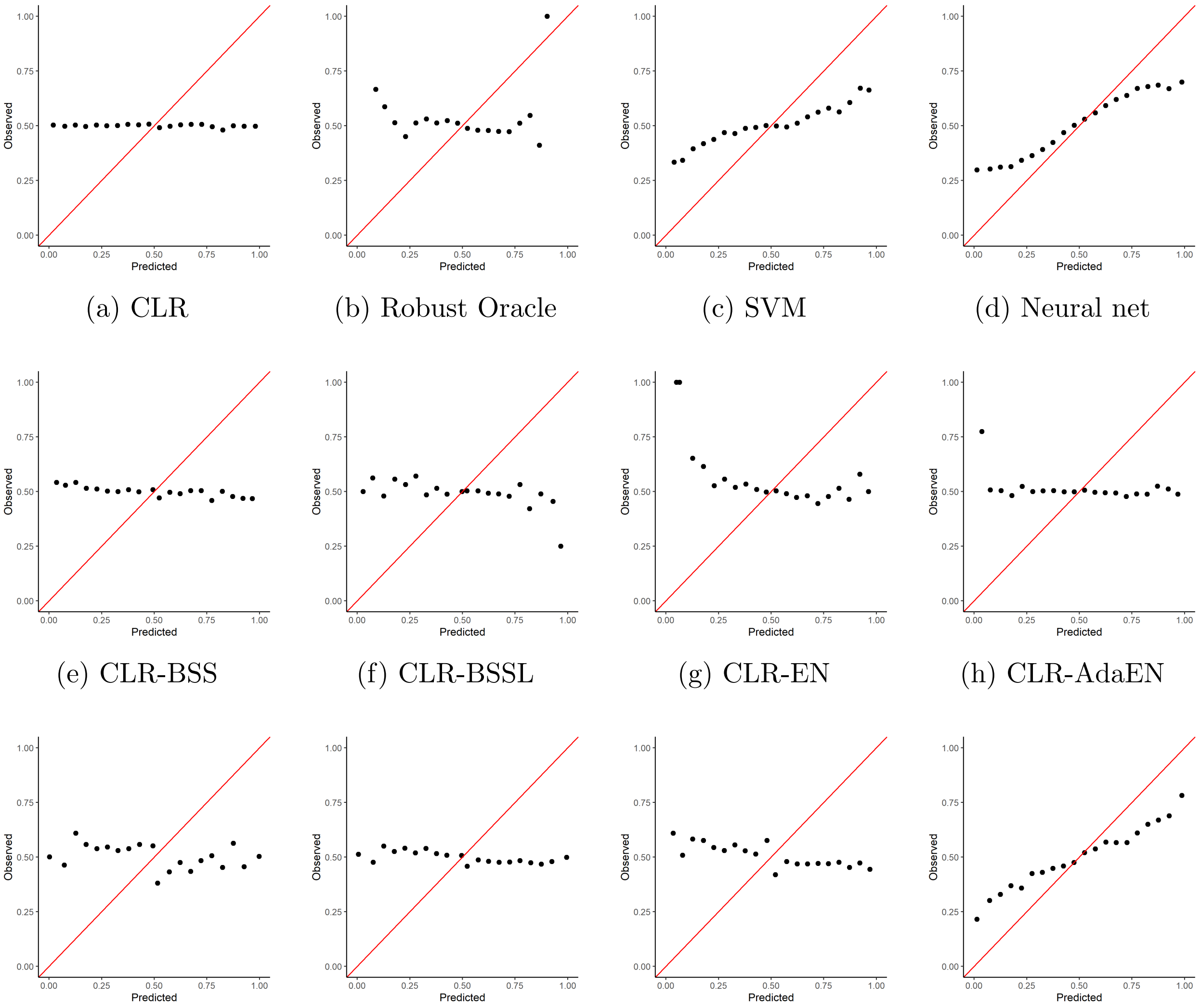
Although the robust methods perform poorly, our results are not contradictory with Avella-Medina and Ronchetti (2017) and Kurnaz et al. (2018b). The former paper only studies Poisson regression, so we cannot compare our results. The later paper focusses on classification and finds that BY-EN performs comparably to CLR-EN in this regard in the absence of contamination. These findings are corroborated by our simulation study. We find that BY-EN attains a misclassification rate of 31.6%, compared to 29.6% for the classical baseline[7].

The machine learning methods perform poorly. The SVM is arguably better than the neural net, though neither produces calibrated forecasts. Out of 150 training examples, 40 were reserved for Platt scaling. Experimentation shows that reserving a larger share of observations for calibration is actually harmful to the performance of the SVM. Platt scaling entails fitting a logistic regression with only an intercept and a single slope, for which 40 observations is ostensibly sufficient. The benefit of increasing the size of the validation set does not outweigh the impact a further reduction of the number of training examples has on the SVM.

4.2.2 Adding 5% bad leverage points

Figure 4 shows that all classical estimators break when the data are contaminated by 5% bad leverage points. The relatively high degree of contamination and severity of the outliers causes the estimators to produce extremely poor coefficient estimates in this setting, which makes their forecasts completely uncalibrated.

[7] These misclassification rates are much higher than those observed in Kurnaz et al. (2018b), but that is because the authors use $\beta_{\mathcal{A}} = [1, 1, \dots, 1]'$, which implies the test data in Kurnaz et al. (2018b) are much closer to being perfectly separable. This improves the classification performance of all methods.



(a) CLR

(b) Robust Oracle

(c) SVM

(d) Neural net

(e) CLR-BSS

(f) CLR-BSSL

(g) CLR-EN

(h) CLR-AdaEN

(i) CR-Lasso (Huber)

(j) CR-Adalasso (Huber)

(k) CR-Lasso (Identity)

(l) BY-EN

Figure 4: Calibration plots for the $p < n$ setting with 5% bad leverage points, all in the $y = 0$ class. Results are obtained across 100 simulation runs with 1000 test observations each. CLR, CR and BY respectively denote classical logistic regression, the Cantoni-Ronchetti estimator and the Bianco-Yohai estimator. (Ada-) EN and BSS (-L) stand for (adaptive) elastic net penalty and best subset selection (with lasso penalty), while Huber and Identity respectively indicate that robust quasi-likelihood with the Huber function or non-robust likelihood was used.

Among the robust estimators, two results are noteworthy. First, the robust oracle breaks. At 5% contamination the breakdown point of the unregularised Cantoni-Ronchetti estimator is exceeded[8], which reflects the local character of the estimator's robustness. Although we cannot rule out that the results of the regularised CR estimators are at least partly driven by

[8] Further testing (not shown) confirmed that the estimator does not break when the contamination level is reduced to 1-2%.

numerical issues, it seems reasonable to assume that the regularised version of the estimator also struggles with this degree of contamination. Avella-Medina and Ronchetti (2017) provide some evidence for this in the Poisson regression setting, as they show that parameter inaccuracy of the estimator (measured by l_2 loss) shoots up when the contamination level exceeds 5%. Second, BY-EN is completely unaffected by the contamination. Though this is in line with Kurnaz et al. (2018b), who showed that the classification performance of the estimator is unimpeded in the same simulation setup, it means that the forecasts of the estimator remain uncalibrated.

The machine learning methods outperform the RCLR estimators in this setting. Whereas the classical estimators all break completely, the machine learning algorithms perform reasonably for this level of contamination. The neural net now outperforms the SVM, which is most easily explained by the fact that the SVM must be calibrated by Platt scaling. This procedure entails fitting an ordinary logistic regression on a validation set and, in practice, it is extremely hard to ensure that the validation set is uncontaminated. Though the SVM may possess some inherent robustness through its implicit regularisation, its forecasts will generally be highly uncalibrated if the validation set contains outliers. The neural net does not require explicit calibration and therefore does not have this exposure.

4.2.3 Discussion

The results clearly indicate when p/n is larger than assumed in classical settings, regularisation is necessary if we wish to obtain calibrated forecasts. In the absence of contamination, the RCLR estimators are vastly superior in terms of calibration, with the bias introduced by regularisation having a negligible impact on calibration. The differences in calibration between best subset selection and the elastic net are marginal. The estimators that perform variable selection by best subset selection produce sparser models, but exhibit a much greater variance than the estimators that use the elastic net. For statistical inference, one would clearly prefer the adaptive elastic net over best subset selection, despite the theoretical benefits of the latter approach.

The RRLR estimator perform poorly, though in the case of the regularised CR estimators numerical issues ostensibly play an important role. If these issues are resolved, better results would probably be obtained. BY-EN produces uncalibrated forecasts, but has the benefit of exhibiting a greater robustness. Whereas the robust oracle, which is just the unregularised CR estimator, breaks at 5% bad leverage points, BY-EN remains unaffected. If the calibration of BY-EN could be improved in the uncontaminated setting, the estimator could serve as an excellent forecasting tool, though computational speedups are required to make it viable for larger datasets. We investigate the behaviour of the estimator in more detail in

Section 4.4.

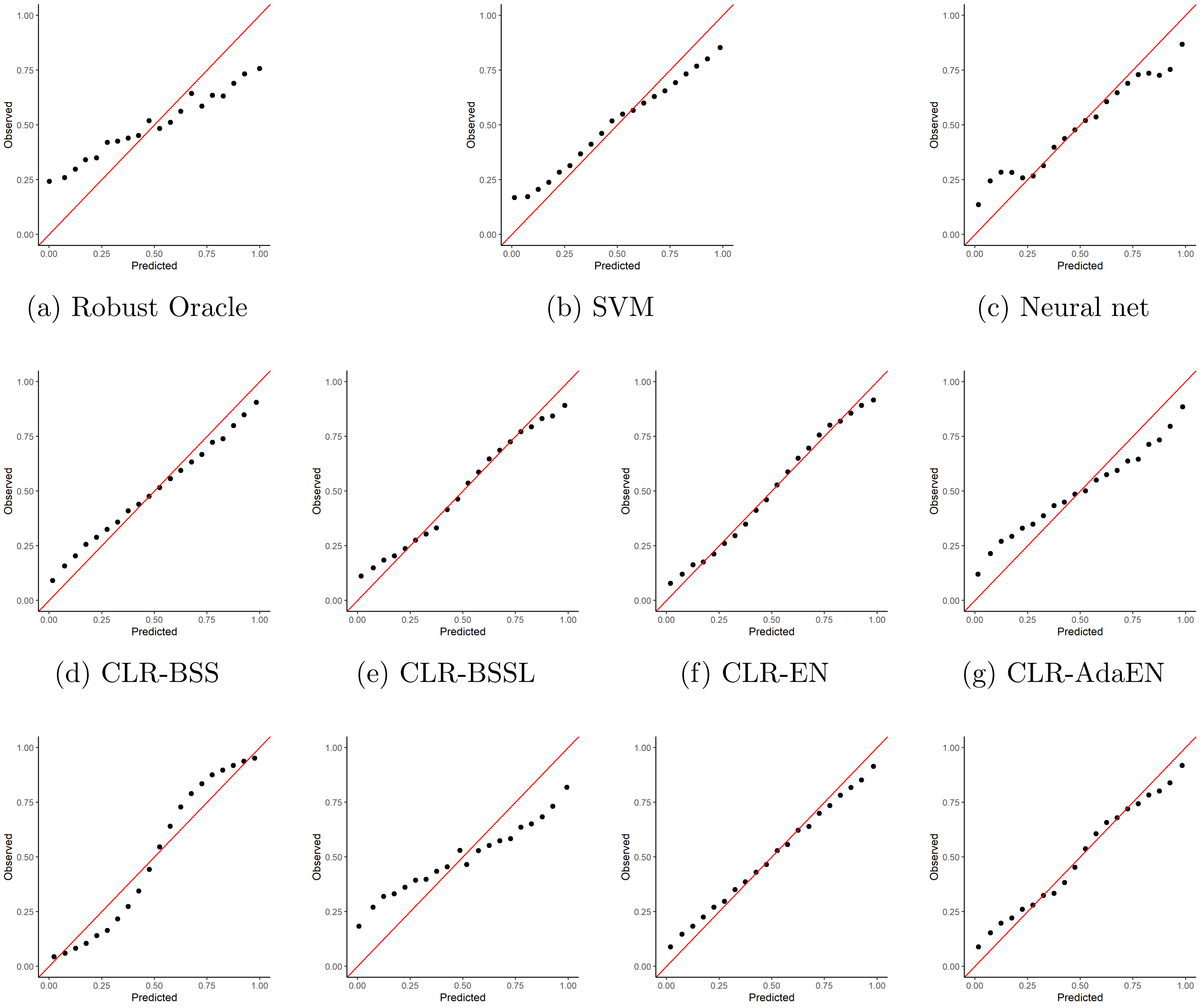
4.3 Results when $p > n$

4.3.1 No contamination

We present calibration plots for the uncalibrated high-dimensional setting in Figure 5. A few things stand out. First, the robust oracle now produces uncalibrated forecasts. With $p = 100$, $n = 50$ and 90% sparsity, the robust oracle now entails fitting a robust estimator on 50 observations of 10 variables, such that $p/n = 0.2$. This is comparable to the performance of classical logistic regression in Section 4.2.1, where $p/n = 0.33$. The non-regularised estimators are unable to deal with such a large dimensionality and produce poor fits.

It is highly surprising that the calibration of BY-EN and the CR estimator with unweighted lasso penalty (CR-Lasso) is better in this scenario than when $p = 50$ and $n = 150$. Distortions to the calibration plot of BY-EN are still greater than those observed for its classical counterpart (CLR-EN), but deviations from the 45-degree line are much less grave than in Section 4.2.1 and calibration is reasonably good. CR-Lasso now exhibits a distinct sigmoidal distortion, but the calibration is undoubtedly better than in Section 4.2.1. The sigmoidal pattern heavily pronounced, but could potentially be overcome by Platt scaling, which is designed to resolve this type of distortion. Although Platt scaling is ordinarily susceptible to outliers, we could use the robustness of the CR estimator to flag outliers. We could, for example, use an initial robust estimate to downweight or remove outliers, and then use the cleaned data to create a train/validation split that is free of outliers.

Calibration of the machine learning methods is comparable to the $p < n$ scenario. Both the SVM and the neural net produce a posterior distribution that is too heavy at the tails, with the latter exhibiting greater distortions. Overall, the machine learning methods struggle with the large p/n ratios used in our simulations and underperform compared to the study of Niculescu-Mizil and Caruana (2005), who use 8 datasets with 4000 training examples each and dimensionality ranging from 16 to 200. Although Niculescu-Mizil and Caruana (2005) use real datasets, which tend to be much more complex than multivariate normal data, our simulations indicate that the SVM and neural net struggle with the dimensionality considered here.



(a) Robust Oracle

(b) SVM

(c) Neural net

(d) CLR-BSS

(e) CLR-BSSL

(f) CLR-EN

(g) CLR-AdaEN

(h) CR-Lasso (Huber)

(i) CR-Adalasso (Huber)

(j) CR-Lasso (Identity)

(k) BY-EN

Figure 5: Calibration plots for the $p > n$ setting without contamination. Results are obtained across 100 simulation runs with 1000 test observations each. Results are obtained across 100 simulation runs with 1000 test observations each. CLR, CR and BY respectively denote classical logistic regression, the Cantoni-Ronchetti estimator and the Bianco-Yohai estimator. (Ada-) EN and BSS (-L) stand for (adaptive) elastic net penalty and best subset selection (with lasso penalty), while Huber and Identity respectively indicate that robust quasi-likelihood with the Huber function or non-robust likelihood was used.

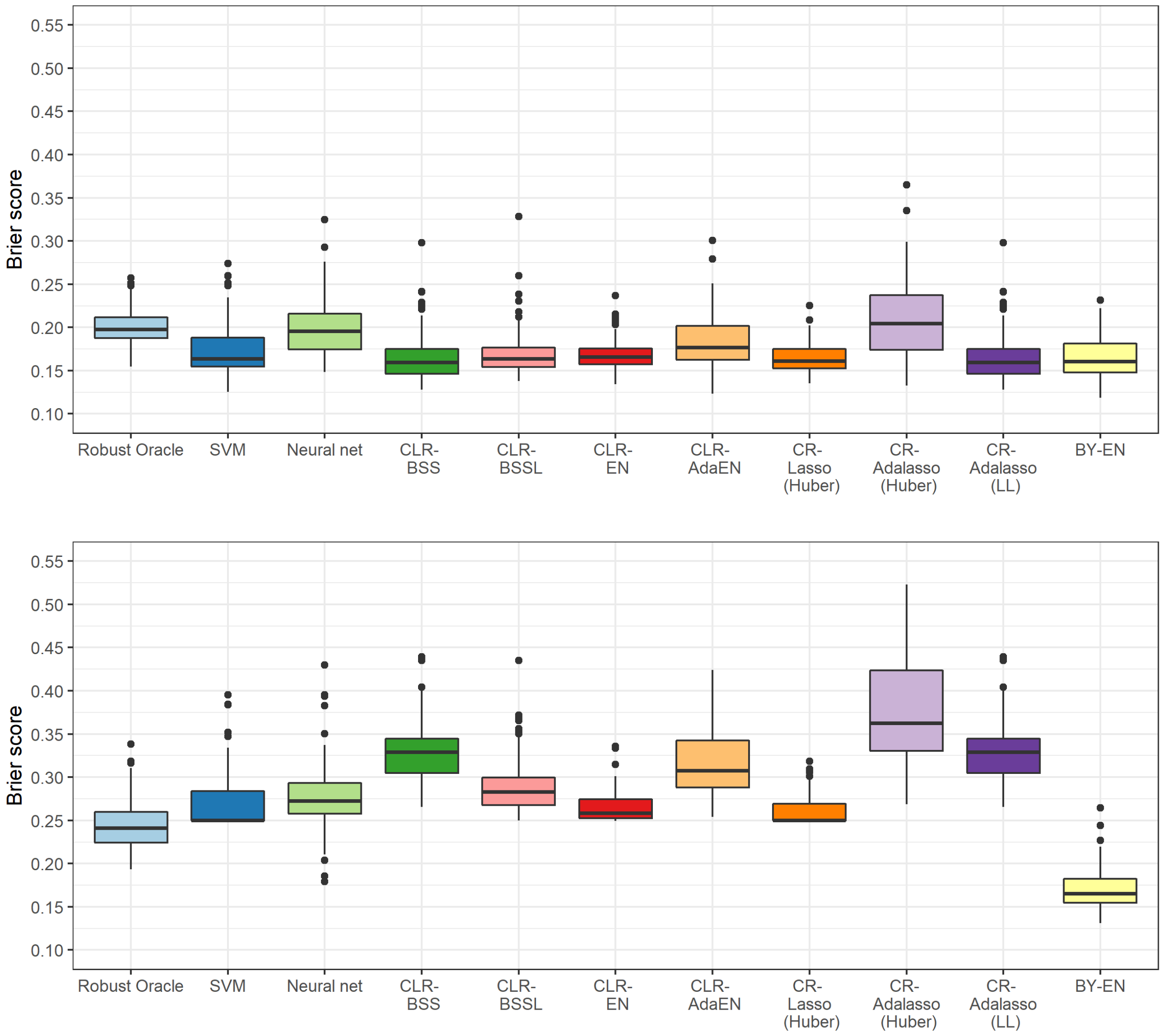
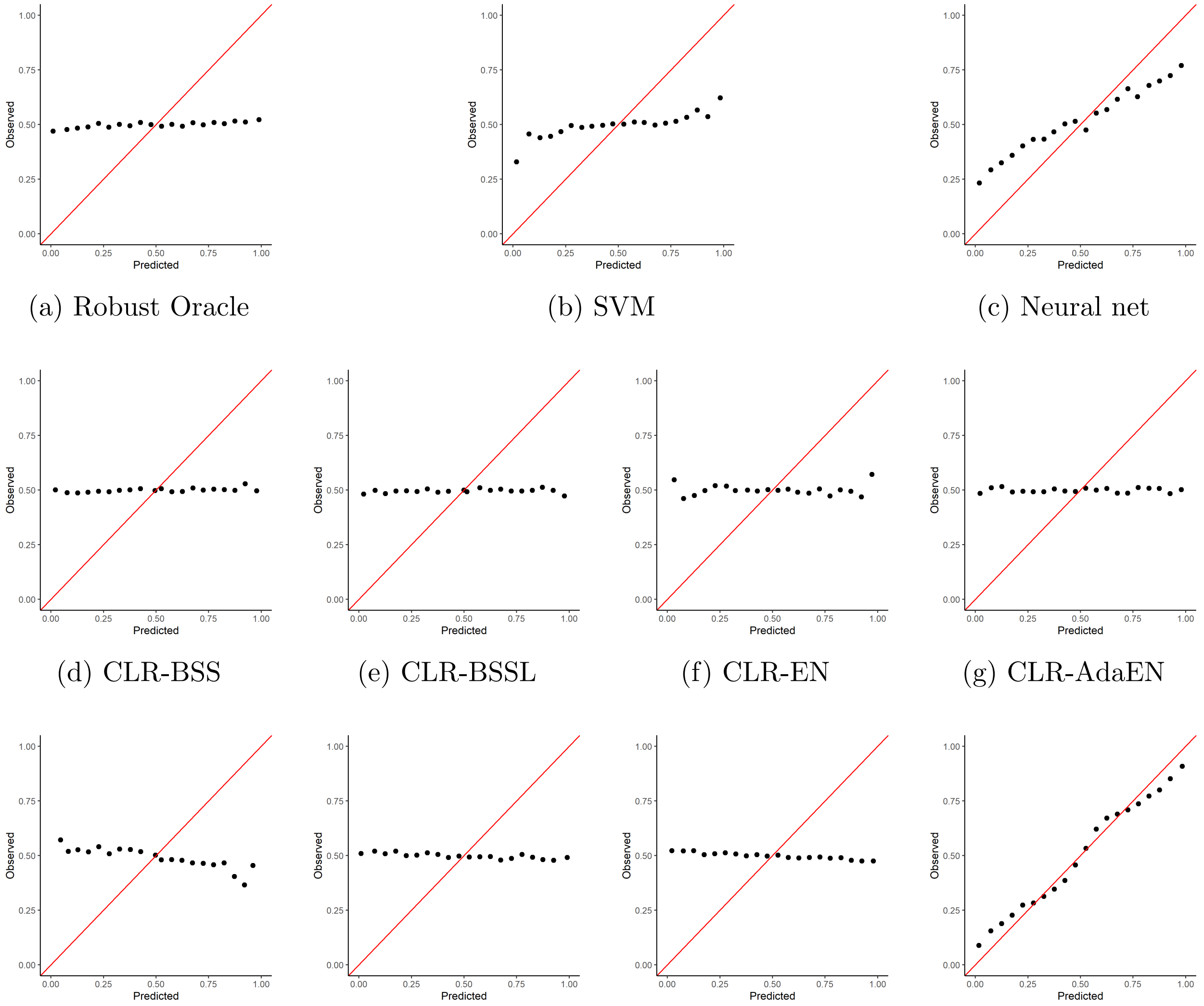


Figure 6: Brier scores for $p = 100, n = 50$. Top: without contamination. Bottom: with 5% bad leverage points, all in class $y = 0$. Results are obtained across 100 simulation runs with 1000 test observations each. In both plots, the vertical axis is measured on a $\log_2$ scale. Except for the support vector machine (SVM), neural net and the 3 rightmost estimators, all estimators are variants of classical logistic regression (CLR). (Lasso-) S* and (Ada-) EN respectively denote CLR with best subset selection (and lasso penalty) and CLR with (adaptive) elastic net penalty. The 3 rightmost estimators are the Cantoni-Ronchetti (CR) estimator with lasso penalty, CR estimator with adaptive lasso penalty and the Bianco-Yianco estimator with elastic net penalty.

4.3.2 Adding 5% bad leverage points

When we introduce bad leverage points, all methods break, except for one. The same robustness that BY-EN exhibited in the $p < n$ setting is shown here and the estimator is almost completely unaffected by the outliers. Figure 6 shows that the worst-case Brier score is slightly higher after contamination, though median Brier scores of 0.161 (no contamination) and 0.165 (contamination) are close to identical.



(a) Robust Oracle

(b) SVM

(c) Neural net

(d) CLR-BSS

(e) CLR-BSSL

(f) CLR-EN

(g) CLR-AdaEN

(h) CR-Lasso (Huber)

(i) CR-Adalasso (Huber)

(j) CR-Lasso (Likelihood)

(k) BY-EN

Figure 7: Calibration plots for the $p > n$ setting with 5% bad leverage points, all in the $y = 0$ class. Results are obtained across 100 simulation runs with 1000 test observations each. CLR, CR and BY respectively denote classical logistic regression, the Cantoni-Ronchetti estimator and the Bianco-Yohai estimator. (Ada-) EN and BSS (-L) stand for (adaptive) elastic net penalty and best subset selection (with lasso penalty), while Huber and Identity respectively indicate that robust quasi-likelihood with the Huber function or non-robust likelihood was used.

Among the remaining methods, the neural net is closest to being calibrated. Though the algorithm's performance is worse in the contaminated setting than when there is no contamination, the neural net is less affected than the other methods. In general, the machine learning methods exhibit a greater degree of robustness than the RCLR methods, though the SVM is inhibited by its reliance on Platt scaling. Given sufficient training data, the SVM could benefit from replacing classical logistic regression by a robust alternative. In the settings considered in this simulation study however, the relative inefficiency of robust methods may instead inhibit the SVM's performance. When there are only 50 training examples, the size of the validation set for calibration is likely too small to effectively use robust regression.

4.3.3 Discussion

The most promising method when $p > n$ is BY-EN. The estimator is approximately calibrated in the uncontaminated setting and is unaffected by the relatively high degree of contamination considered in this study. The fact that the estimator performs much better when $p > n$ than when $p < n$ makes us question what determines its behaviour, which is what we investigate in the next section.

4.4 What governs the behaviour of the Bianco-Yohai estimator?

The simulations show that the robust methods produce uncalibrated forecasts, but do not provide insight into what governs their behaviour. It is surprising that the robust estimators do not produce calibrated forecasts at the model distribution, when their classical counterparts are calibrated. In this section, we try to explain this phenomenon for BY-EN. We focus on this estimator for two reasons. First, among the two robust estimators it is closest to being calibrated, with particularly strong performance in the $p > n$ setting. Second, experience with the enetLTS package lets us rule out that BY-EN's behaviour governed by numerical properties. This makes the estimator easier to investigate than the CR estimator, which exhibits numerical erraticism during our simulations.

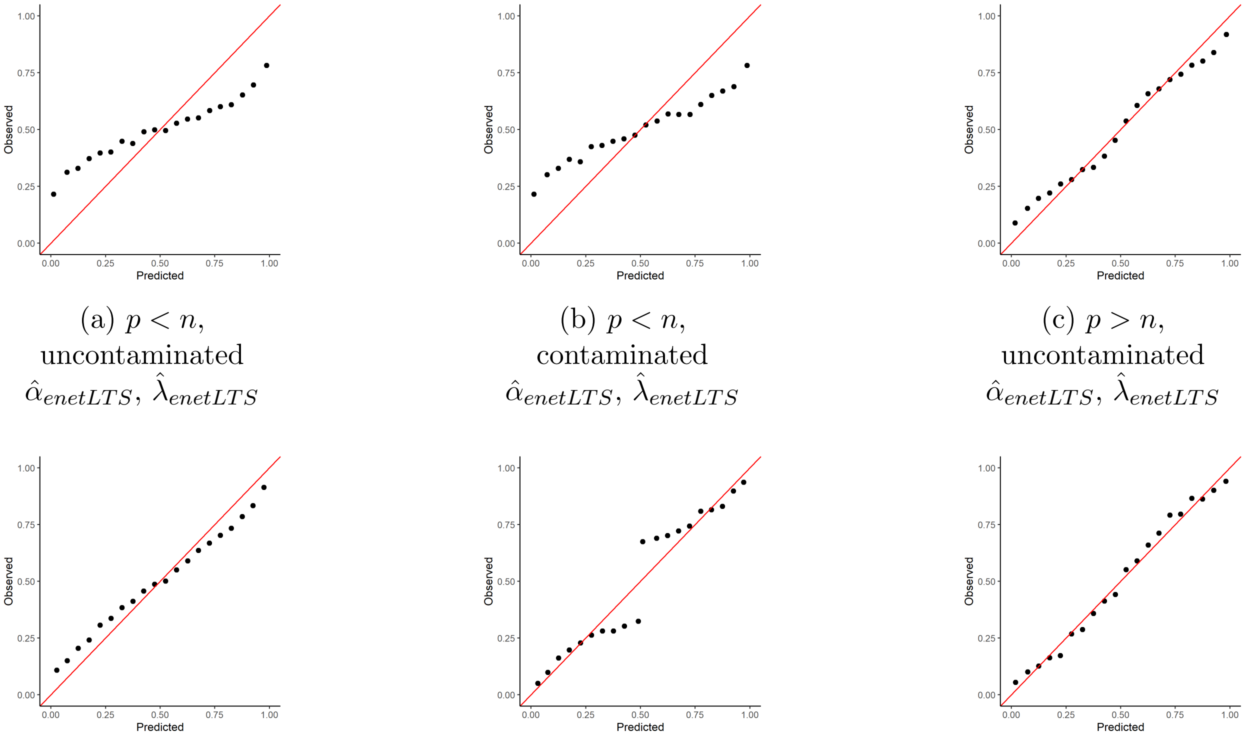
A reasonable guess as to why the robust methods are less calibrated than their classical counterparts is the influence of shrinkage. We know that both the quasi-likelihood approach of Cantoni and Ronchetti (2001) and the bounded deviance approach of Croux and Haesbroeck (2003) are Fisher consistent and lead to calibrated forecasts when $p \ll n$. This is also reflected in the performance of the robust oracle in Section 4.2.1, where it must estimate 5 coefficients (intercept is zero) based on 150 observations. Whether we construct the robust oracle using the quasi-likelihood approach or the bounded deviance approach, it produces calibrated forecasts. However, when the robust oracle must be fitted based on data where

$p = 10$ and $n = 50$ in Section 4.3.1, it is no longer calibrated. This suggests BY-EN may be uncalibrated simply because it applies insufficient shrinkage.

To test this hypothesis, we first compare the regularisation paths of the robust estimators to their classical counterparts. For the classical estimators as well as the robust estimators, the regularisation path (given α) entails a sequence of values λ in $[0, \lambda_0]$, or $(0, \lambda_0]$ when $p > n$ since the unpenalised estimator is undefined in the latter case. For the classical estimators, λ_0 is chosen such that $\alpha\lambda_0 = \frac{1}{N}\max_{j\in\{1,\ldots,p\}}|\text{Cor}(x_j, y)|$ and the interval $[0, \lambda_0]$ is subsequently divided into a sequence of values on the log scale (Friedman et al., 2010). By tuning λ over the corresponding interval for all relevant values of α, we obtain the optimal combination of hyperparameters $\hat{\alpha}$ and $\hat{\lambda}$. Kurnaz et al. (2018a) follow this approach almost exactly for the reweighted version of BY-EN, as they compute its regularisation path using glmnet[9]. An important difference with the classical estimator is that outliers, as determined by the raw estimator, are excluded when computing the regularisation path. This implies a different choice for λ_0, which inevitably results in a different estimate of the optimal hyperparameter $\hat{\lambda}$. Further, to reduce the computational burden, the mixing parameter α is taken from the raw estimator, as opposed to tuning it for the reweighted estimator. There are thus two main differences between the tuning procedures for CLR-EN and BY-EN. First, λ is tuned for the reweighted estimator using a data subset. Second, α is only tuned for the raw estimator, which in fact relies on an even smaller subset of the data.

To inspect how tuning α and λ based on a subset of the data affects calibration, we revisit our simulations. We inspect the calibration of BY-EN based using the same training and test data, but override the tuning procedures of enetLTS. Instead of using the values $\hat{\alpha}_{\text{enetLTS}}$ and $\hat{\lambda}_{\text{enetLTS}}$ tuned by enetLTS, we force the reweighted BY-EN estimator to be fitted using values $\hat{\alpha}_{\text{glmnet}}$ and $\hat{\lambda}_{\text{glmnet}}$ tuned by glmnet for CLR-EN.

[9] For the raw estimator a completely different regularisation path is computed based on a robust correlation measure.



(a) $p < n$, uncontaminated $\hat{\alpha}_{enetLTS}$, $\hat{\lambda}_{enetLTS}$

(b) $p < n$, contaminated $\hat{\alpha}_{enetLTS}$, $\hat{\lambda}_{enetLTS}$

(c) $p > n$, uncontaminated $\hat{\alpha}_{enetLTS}$, $\hat{\lambda}_{enetLTS}$

(d) $p < n$, uncontaminated $\hat{\alpha}_{glmnet}$, $\hat{\lambda}_{glmnet}$

(e) $p < n$, contaminated $\hat{\alpha}_{glmnet}$, $\hat{\lambda}_{glmnet}$

(f) $p > n$, uncontaminated $\hat{\alpha}_{glmnet}$, $\hat{\lambda}_{glmnet}$

Figure 8: Calibration plots for the Bianco-Yohai estimator with elastic net penalty in multiple scenarios. Results are obtained across 100 simulation runs with 1000 test observations each. Subscript $enetLTS$ indicates that a hyperparameter was tuned by the enetLTS package, while subscript $glmnet$ indicates that the glmnet package was used.

The results in Figure 8 show that when using $\hat{\alpha}_{\text{glmnet}}$ and $\hat{\lambda}_{\text{glmnet}}$, BY-EN produces nearly calibrated forecasts in the uncontaminated setting. The main drawback is a loss of robustness, because the hyperparameters are no longer tuned based on cleaned training data. The main question that arises from Figure 8 is why the use of hyperparameters selected by glmnet improves calibration. Indeed, $\hat{\alpha}_{\text{glmnet}}$ and $\hat{\lambda}_{\text{glmnet}}$ are not even optimised for BY-EN, yet improve calibration compared to $\hat{\alpha}_{\text{enetLTS}}$ and $\hat{\lambda}_{\text{enetLTS}}$ which were explicitly tuned for the (raw or reweighted) BY-EN estimator. As the two cross-validation procedures seek to minimise the same loss criterion, this result is highly counterintuitive.

Upon closer inspection of the objective functions based on which α and λ are tuned by glmnet and enetLTS, it becomes clear that there is one considerable difference between the two. If we abbreviate the elastic net penalty as $P(\beta)$, the glmnet criterion may be written as

$$\ell_{glmnet}(\beta) = \sum_{i=1}^{n} \left(-y_i(X_i'\beta) + \log\left(1 + e^{X_i'\beta}\right) \right) + n\lambda P(\beta), \tag{17}$$

which is of course equivalent to Equation (4). For the reweighted BY-EN estimator, on the other hand, λ is tuned for an objective

$$\ell_{enetLTS}(\beta) = \sum_{i=1}^{n} w_i \left(-y_i(X_i'\beta) + \log\left(1 + e^{X_i'\beta}\right) \right) + \sum_{i=1}^{n} w_i \lambda P(\beta), \tag{18}$$

where w_i are weights that remove flagged outliers, which are determined by the rejection rule in Equation (11) using the raw coefficient estimates. According to Kurnaz et al. (2018b), the objective function in Equation (18) uses 97.5% of data in expectation under the model distribution. If this claim is true, the criteria are virtually identical and should lead to nearly identical solutions.

When the unregularised BY estimator is used in the low-dimensional setting and there is no sparsity, this claim indeed holds true. In that case, we know the BY estimator is Fisher consistent and the share of false positives is controlled under the model. However, for the regularised estimator considered here the claim is false. The large dimensionality and elastic net make the raw estimator biased. Though we have no formal proof, as Kurnaz et al. (2018b) do not provide theory regarding the (asymptotic) consistency of their estimator, we find strong simulation evidence that the raw estimator does not recover true class probabilities π_i in expectation under the model distribution. Then Pearson residuals r_i from Equation (10) are also distorted, which means that the expected fraction of data used in Equation (18) is not 97.5%. Depending on how the raw parameter estimates are distorted, too many observations may be excluded from Equation (18). enetLTS may thus produce worse estimates of α and λ than glmnet, at least in the absence of contamination, because it discards too much 'good' data and is thus much less efficient than suggested.

This is exactly what happens in our simulations. Across 100 repetitions, the raw BY-EN estimator flags 17.1% of observations as outliers on average in the uncontaminated $p < n$ setting. In the uncontaminated $p > n$ setting, this is 10.0% on average, which helps explain why the two hyperparameter tuning methods lead to more similar results in this case than when $p < n$. Even though much more than 2.5% of the training data is excluded on average, less of the variation of X is removed from the data when $p > n$ than when $p < n$. When too much variation is removed from the data, as in our $p < n$ setting, our estimate of the optimal degree of shrinkage is biased towards zero. Much as Hastie et al. (2017) found that the best subset selection operator (which essentially applies zero shrinkage) dominates lasso in the high signal-to-noise regime because less shrinkage is required in this setup than in the noisy regime, the ‘clean’ data causes enetLTS to produce a solution with less shrinkage than what is required for the more noisy data generating process. In other words, the outlier rejection rule in Equation (11) removes so much of the variability in the training data that it is no longer representative of the data generating process. BY-EN overfits the ‘clean’ data and has poor out-of-sample performance, because the test data is drawn from the (more variable) data generating process. To resolve this issue, one needs to either improve the raw estimate or change the rejection rule. The current rule is valid if the Pearson residuals are (asymptotically) normal, but clearly this assumption is unreasonable when the raw estimator is used in its current form.

4.5 Cellwise Contamination

In this section we investigate the behaviour of the estimators in the presence of cellwise contamination. We inspect only the CLR-EN, CLR-AdaEN, CLR-BSS and CLR-BSSL estimators as they produced calibrated forecasts in the uncontaminated setting. We further include the robust oracle, to see if its robustness under the rowwise contamination paradigm has any value in the presence of cellwise contamination. The simulation setup is identical to that of Section 4.2.1, except we randomly replace 5% of cells in training design X by draws from a $N(20, 1)$ distribution.

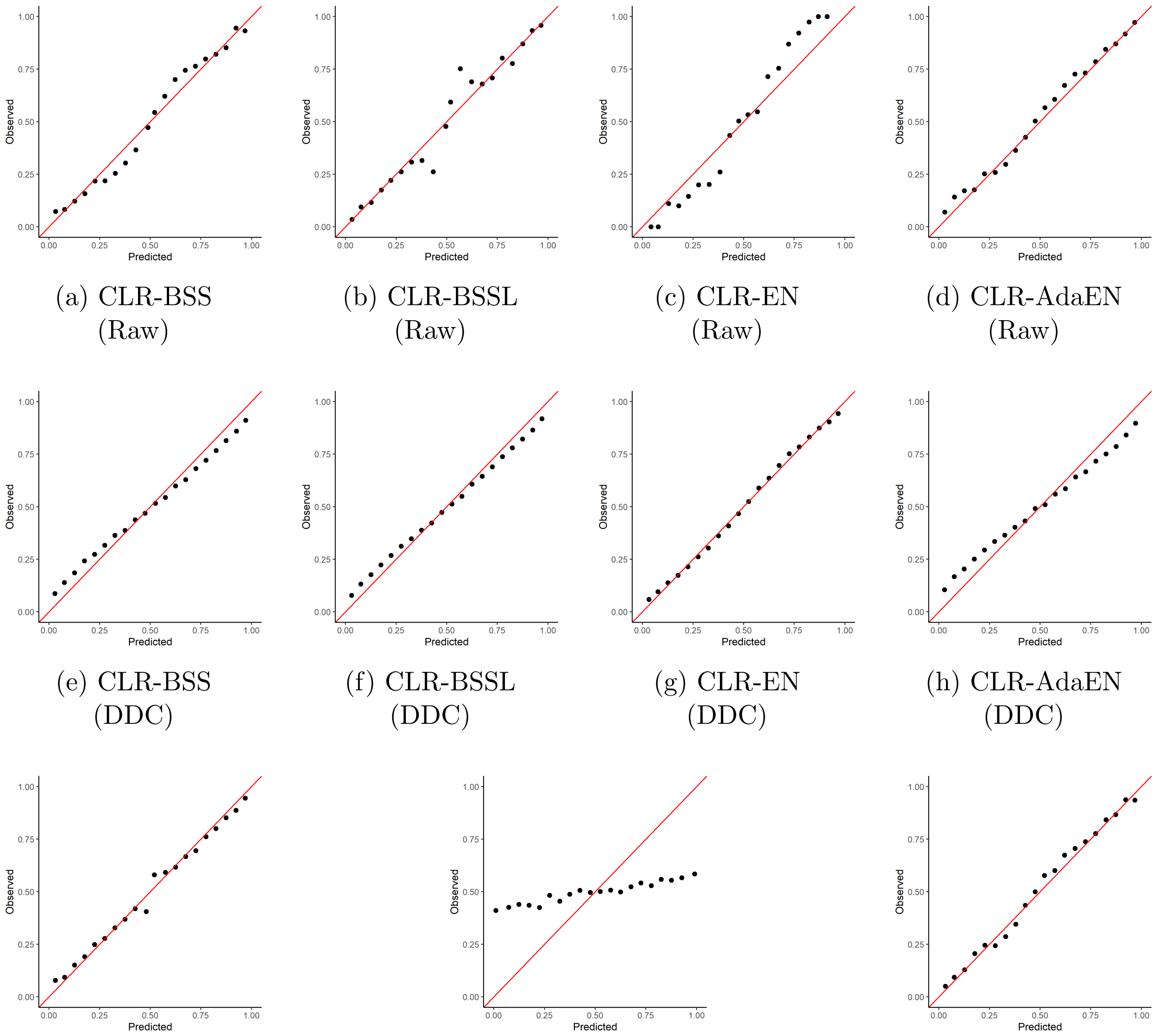
4.5.1 No Robustness

We first inspect what happens when no robustness measures are taken. The results of this scenario may be found in the subfigures of Figure 9 that are labelled *Raw*. The CLR-AdaEN estimator performs best in this setting, but clearly none of the methods are robust. All estimators, including the CLR-AdaEN, can be made uncalibrated by making the outliers sufficiently extreme. We further observe that the robust oracle breaks in this scenario. Although the estimator is able to handle 5% rowwise outliers, the propagation of outliers under the cellwise contamination paradigm means that the breakdown point of the estimator is exceeded in this scenario.

4.5.2 DDC

The subfigures of Figure 9 that are labelled *DDC* exhibit the results that are obtained when the DDC algorithm is applied to the training data before the estimators are fitted. Improvements can be observed in all regularised CLR methods, which are now roughly as well calibrated as they were in Section 4.2.1. A notable exception to this improvement is the robust oracle, which becomes completely uncalibrated when the DDC algorithm is applied.

We speculate that this result is due to model inconsistency that arises when cellwise and rowwise outliers are treated separately, as discussed by Filzmoser et al. (2020). If pre-processing of the training data by DDC is inconsistent with the model assumed by the robust oracle, then we may observe a worsening of results as seen here.



(a) CLR-BSS (Raw)

(b) CLR-BSSL (Raw)

(c) CLR-EN (Raw)

(d) CLR-AdaEN (Raw)

(e) CLR-BSS (DDC)

(f) CLR-BSSL (DDC)

(g) CLR-EN (DDC)

(h) CLR-AdaEN (DDC)

(i) Robust Oracle (Raw)

(j) Robust Oracle (DDC)

(k) CLR-AdaEN (Robust weights)

Figure 9: Calibration plots for the $p < n$ setting with 5% cellwise contamination. Results are obtained across 100 simulation runs with 1000 test observations each. CLR denotes classical logistic regression. (Ada-) EN and BSS (-L) stand for (adaptive) elastic net penalty and best subset selection (with lasso penalty). Raw indicates no robustness measures were taken, while in plots labelled DDC the training data were pre-processed with the DetectDeviatingCells algorithm. Robust weights indicates robust adaptive weights were used.

4.5.3 Robust Adaptive Weights

The poor calibration of the robust oracle in the preceding section demonstrates the need for methods that can simultaneously handle rowwise and cellwise outliers. As a first venture into this topic, we inspect the behaviour of the CLR-AdaEN estimator when we compute robust adaptive penalties as in Machkour et al. (2020). The calibration plot of this estimator may be found in the last subfigure of Figure 9. Although the graph trails the 45-degree line fairly closely, we still observe some deviations. The calibration plot is highly similar to that of the CLR-AdaEN without robust adaptive weights (Section 4.5.1), though this may simply be a matter of our choice of simulation parameters. If the fraction of contamination or the gravity of the outliers was increased, we expect that the CLR-AdaEN with robust adaptive weights outperforms the performs its non-robust equivalent. The relatively strong performance of the non-robust CLR-AdaEN indicates that the adaptive elastic net penalty inherently possesses some robustness. If outlying variables are penalised heavily by the initial estimator, the non-robust adaptive weights should resemble their robust counterparts to some degree, even if outlyingness is not explicitly incorporated into the adaptive weights.

5 Real Datasets

In this section we investigate if the calibration results of the simulation study extend to real datasets. We use two datasets, which we inspect in turn. For each dataset, we first describe the data used and subsequently inspect calibration of the methods. As in Section 4, we use calibration plots. If multiple methods are calibrated, we again use the Brier score to determine which is most efficient.

5.1 Dataset 1: Molecule Classification

5.1.1 Data Description and Processing

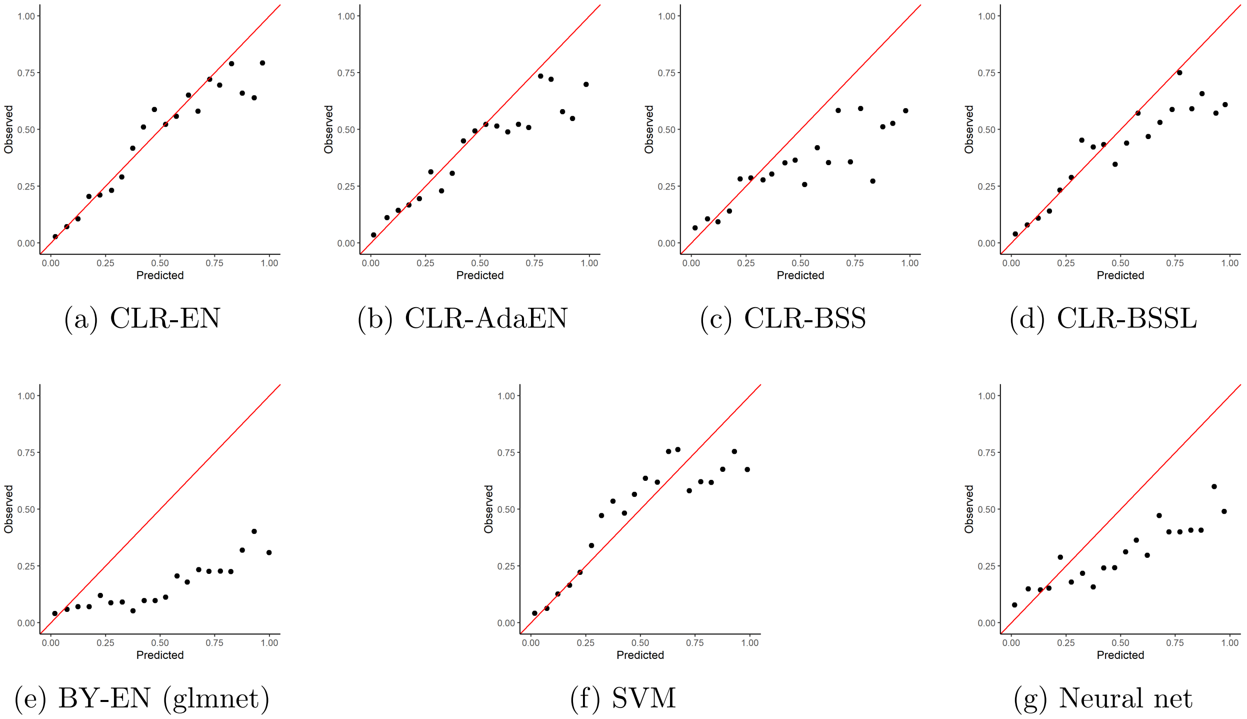
We first consider a dataset from the field of chemistry, which was originally analysed in Dietterich et al. (1994). The data describes the conformations ('shapes') of a series of 102 molecules, which must be classified as either being a musk or not being a musk. These conformations are determined by measuring the position of each component of a molecule (i.e. an atom) relative to the position of a 'baseline' molecule, which was chosen arbitrarily. Predictors are therefore the distances between an atom's positioning and the position of the corresponding atom of the baseline molecule. Distances are measured in hundredths of Angstroms (10^{-12} metres). Although this means the predictors are integer-valued, we treat the data as being continuous on an arbitrary scale, per the recommendations of the authors.

The ground truth classification is determined by human experts, who have established that 39 of the 102 molecules are musks and the remainder are non-musks. The dataset consists of a total of 6,598 observations of 166 predictors and a class indicator. We remove 3 predictors that have a median absolute deviation of zero. The data are imbalanced, with 15.4% of observations being classified as musks. Further, the predictors exhibit a relatively large variability for both classes. Though this is expected for the non-musk molecules, which are likely more diverse in conformation, predictor values are highly variable even among the class of musks. A simple check using robust Mahalanobis distances[10] flags 2,383 observations as suspect, of which 110 are in the class of musk molecules. This accounts for approximately 10% of all musk observations, which one might intuitively expect to be more alike. Overall, the dataset appears to contain multiple observations whose location in the design space may make them influential.

Although we are interested in probabilistic forecasting with high-dimensional data, the fact that $p < n$ in this dataset is actually an advantage. As there are considerably more

[10] We compute Mahalanobis distances using the median as location estimate and the MCD matrix as scatter estimate. The 97.5th percentile of the $\chi^2(163)$ distribution is used as cutoff.

observations than predictors, we shuffle the data and split it into 10 datasets of 659-660 observations each. The split takes into account that the classes are imbalanced, approximately replicating this imbalance in each dataset. We then create a training/test split in each smaller dataset, again taking into account class imbalance. This approach begets us 10 high-dimensional datasets which each have 163 predictors, 75 training observations and 584-585 test observations. Compared to a single train/test split with a large test set, our approach has the advantage of fitting each estimator 10 times on independent datasets. This helps alleviate the randomness of parameter estimates. A drawback of the approach is that fewer observations are available for forecasting, as we need 10 times more training examples than if we used a single split. However, we believe that the benefits of fitting the estimator on more than one training set outweighs this cost.



(a) CLR-EN

(b) CLR-AdaEN

(c) CLR-BSS

(d) CLR-BSSL

(e) BY-EN (glmnet)

(f) SVM

(g) Neural net

Figure 10: Calibration plots for the molecule classification data. Results are obtained by splitting the original data into 10 smaller datasets, which are subsequently split into 75 training observations and 584-585 test observations each. CLR denotes classical logistic regression, whereas (Ada)EN and BSS(L) respectively indicate an (adaptive) elastic net penalty or best subset selection (with lasso penalisation) was used for variable selection. The BY-EN estimator is the Bianco-Yohai estimator with elastic net penalisation, using hyperparameters optimised for the CLR-EN estimator.

5.1.2 Results

The calibration plots in Figure 10 show that CLR-EN attains the best results on this dataset. The greatest deviations from the 45-degree line occur in the largest probability bins, where the fitted forecasting distribution is less dense than the empirical distribution. This appears to primarily be a result of too many test observations being assigned forecasted probabilities of approximately 50%, a phenomenon that was already observed in the simulations in Section 4 as a result of shrinkage. The main difference with the simulation results is that deviations from the 45-degree line are now concentrated in a single tail, such that the distortions to the calibration plots are no longer sigmoidal. This is attributable to class imbalance, with the estimator only observing an average of $\lfloor 0.154 * 75 \rfloor = 11$ musk molecules in each training set. The difficulty that this imbalance presents is well-established in the literature (e.g. Weiss, 2004) and is common to all RCLR estimators. Overall, results regarding the RCLR estimator are in line with those found in the simulations of Section 4.3.1. As in the simulations, the elastic net outperforms its adaptive counterpart. When the number of training examples is small, the theoretical benefits of the adaptive penalty need not lead to any improvement in forecasting performance. Among the methods based on best subset selection, the penalised estimator clearly outperforms, illustrating the benefits of shrinkage.

Neither of the robust regularised estimators managed to converge on any of the datasets. The BY-EN failed to achieve convergence within the time limit of 30 minutes, whereas all versions of the CR estimator produced numerical errors during the integration step used to compute the robust quasi-likelihood[11]. As a reference method, we instead include BY-EN with hyperparameters selected by glmnet as discussed in Section 4.4. Figure 10 shows that the estimator's forecasts are highly uncalibrated and inferior to those of CLR-EN. Despite the large variability of the predictors, the penalised classical estimators are clearly preferable for this data.

Between the machine learning methods, the SVM's performance stands out, especially considering that the SVM learns based on fewer training examples than the other methods. In each dataset, 30 training examples were reserved for Platt scaling, such that the SVM is effectively trained on 45 observations. Nonetheless, the SVM can compete with the RCLR estimators, in terms of calibration as well as sharpness. Both CLR-EN and the SVM attain a Brier score of approximately 0.093, lower than any other method. The neural net is less effective and is completely uncalibrated.

[11] Similar errors occurred when the code of Avella-Medina and Ronchetti (2017) was first used in the simulation study, but minor changes to the code ensured that the errors never occurred again during the simulations. We have not been able to figure out why the errors re-occur in this application.

5.2 Dataset 2: Predicting Trisomy in Mice

5.2.1 Data Description and Processing

The second dataset is more recent and comes from the field of biology. The data were originally studied by Higuera et al. (2015) and describe the expression levels of 77 proteins that produced measurable signals in the brains of mice. The goal of the study was to identify the subset of proteins that best distinguish between trisomic mice (i.e. mice with Down syndrome) and non-trisomic mice.

The data consists of 15 measurements of 72 mice, for a total of 1,080 observations. Per the recommendations of the authors, we treat these observations as describing 1,080 independent samples. The data originally describes 8 classes of mice, which were based divided on the presence or absence of trisomy, whether the mice were drugged and whether they were stimulated to learn. We reduce this to 2 classes, focussing only on the presence or absence of trisomy. Among the 72 mice, 34 were trisomic, such that the resulting dataset is approximately balanced (47.2% positive labels). We encode the data on whether the mice were drugged and whether they were stimulated to learn as two additional, binary predictors. We consider this approach more meaningful than removing this data. It is probable that trisomic and non-trisomic react differently to these treatments, in which case the dummy variables are informative predictors of trisomy and should be kept in the dataset. The inclusion of the dummy variables begets us a total of 78 predictors, with the remaining 76 consisting of protein expression levels, which are continuous. As the expression of protein *pS6_N* is perfectly collinear with the remaining predictors, we remove it from the data, reducing the total to 77 predictors.

Among the 75 continuous predictors retained in the dataset, approximately 1.7% of cells is missing values. In the absence of further information, we treat these cells as being missing completely at random (Rubin, 1987) and impute them using multiple imputation by chained equations, implemented in **R** using the mice[12] package (Van Buuren and Groothuis-Oudshoorn, 2021). To control the influence of potential outliers, we estimate each missing value 10 times and impute the median.

As in the preceding section, we split the (imputed) data into multiple smaller datasets instead of using a single train/test split. Because the dataset is smaller, we split the data into 3 folds of 360 observations. Each fold is then split into 66 training observations and 294 test observations, such that we again create a high-dimensional problem.

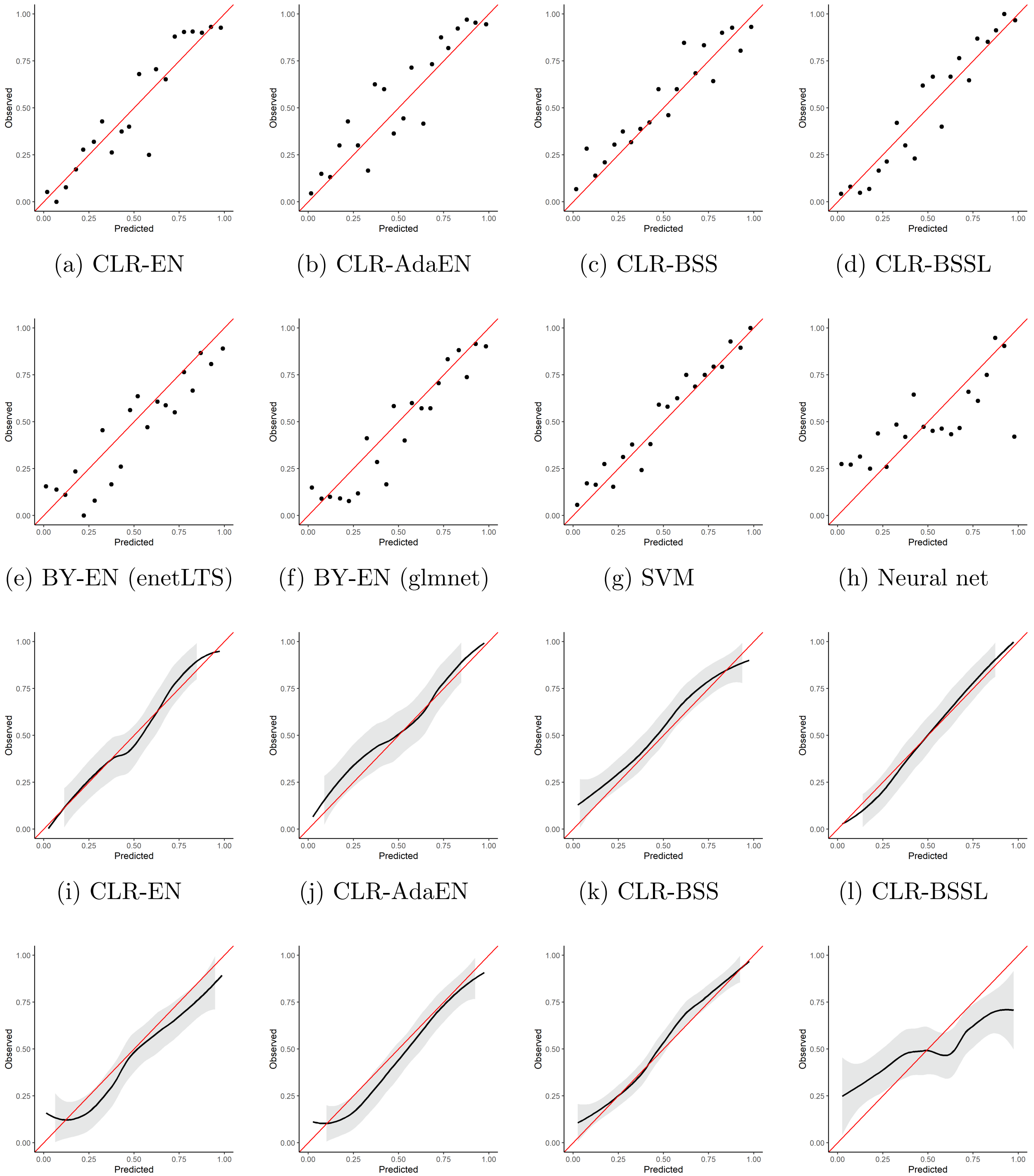
[12] The irony was not lost on us.

5.2.2 Results

The top two rows of Figure 11 show that the smaller number of test observations available in this dataset leads to highly variable calibration plots. For this reason, we include smoothed versions of the plots in the bottom two rows. Smoothing is performed using local (LOESS) regression and we add the 95% confidence intervals of the regressions. The results as shown in the smoothed calibration plots are quite impressive, with multiple methods trailing the 45-degree line closely. CLR-EN, CLR-BSSL and the SVM are arguably the best performers and no longer exhibit large deviations at the right tail of the empirical distribution, now that the classes are more balanced. Overall, the calibration plots suggest that balancing classes may be highly beneficial to forecasting performance. One may exploit this knowledge by e.g. oversampling the minority class, an approach that has already been shown to be effective in practice (see e.g. Cerqueira et al., 2016).

Contrary to Section 5.1, the Bianco-Yohai estimator converges in under 30 minutes on this dataset, as both the number of training examples and the dimensionality are now smaller than in the preceding section. Figure 11 shows that the estimator's calibration is mediocre compared to that of its classical counterpart, with the estimated forecast distribution having fatter tails than its classical counterpart. Nonetheless, the 45-degree line stays within the 95% confidence interval along the entire graph, which is acceptable. When we fit BY-EN using hyperparameters α and λ as selected by glmnet for the classical counterpart, calibration visibly improves. This is in line with our findings in the simulation study, which showed that the estimator may be improved by using a less harsh outlier rejection rule. Too much of the training data is removed using the threshold suggested by Kurnaz et al. (2018b), which results in poorly optimised hyperparameters.

To round up our analysis of this dataset, we compare the sharpness of the calibrated forecasting methods to see which method performs best. In terms of pure calibration performance, one would choose CLR-BSSL based on the smoothed plots in Figure 11. The SVM is also a viable candidate and has the advantage of its calibration being less uncertain. In the scatterplots, the SVM exhibits the least variability, which is also reflected in the comparatively tight 95% confidence interval in the smoothed plots. This is not reflected in the sharpness of the forecasts however, where the SVM (Brier score of 0.141) underperforms the CLR-BSSL (0.091). Overall, one would likely prefer the CLR-BSSL for the forecasting problem that was posed in this section.



(a) CLR-EN (b) CLR-AdaEN (c) CLR-BSS (d) CLR-BSSL

(e) BY-EN (enetLTS) (f) BY-EN (glmnet) (g) SVM (h) Neural net

(i) CLR-EN (j) CLR-AdaEN (k) CLR-BSS (l) CLR-BSSL

(m) BY-EN (enetLTS) (n) BY-EN (glmnet) (o) SVM (p) Neural net

Figure 11: Calibration plots for the Trisomy data. The bottom two rows are smoothed versions of the top two rows, shaded regions are 95% confidence intervals. Results are obtained across 3 datasets, which are split into 66 training observations and 294 test observations each. CLR and BY respectively denote classical logistic regression and the Bianco-Yohai estimator, while (Ada)EN and BSS(L) indicate an (adaptive) elastic net penalty or best subset selection (with lasso penalisation). enetLTS and glmnet refer to the package used to tune hyperparameters BY-EN estimator.

6 Discussion

In this section we summarise the main results of the paper and discuss important limitations to which the results are subject. Subsequently, we provide some recommendations for future research.

6.1 Results and Limitations

This study shows that probabilistic forecasting with high-dimensional data is a viable endeavour. Whether we use best subset selection or an elastic net penalty, the RCLR estimators outperform the machine learning methods and produce calibrated forecasts with greater sharpness in both uncontaminated simulation scenarios. This finding is not just applicable to simulations where the logistic regression methods are correctly specified by design, as the estimators that were viable in the simulations also perform well on the real datasets. Pertaining to the robust estimators, it is important to note that our findings only reflect the theoretical properties of the BY-EN estimator. The regularised Cantoni & Ronchetti estimator was mostly inhibited by implementation issues and could not be thoroughly investigated as a result. Nonetheless, this drawback of the estimator as it is currently available is imperative and implies that, in its current implementation, it cannot be used for logistic regression.

To choose between estimators in practice, one would look at other crucial criteria such as computation time, which we almost completely ignored in this paper. This is an important limitation of our analysis. We deemed the BY-EN competitive with the classical estimators in the high-dimensional setup, but its computation time is so much larger (minutes, even in small datasets, as opposed to seconds for the classical methods) that the estimator is no longer viable if p or n is increased meaningfully. Similarly, we treated best subset selection and the convex elastic net penalty as equivalent because of their forecasting performance, but for practical applications one would virtually always prefer the elastic net. Not only does the convex penalty lead to a reduced computation time, convex optimisation algorithms are implemented and freely available in virtually all popular programming languages. Optimisation algorithms for the best subset selection operator, on the other hand, rely on mixed integer programming solvers. These are typically only implemented in commercial software packages.

Another important limitation of this paper was alluded to in our analysis of the Molecule Classification data. The results clearly show that all methods struggle with class imbalance, which is a feature of many real-world classification datasets. There is a large branch of literature that focusses on classification with class imbalance, also for high-dimensional data, but probabilistic forecasting with such data has not yet been investigated to the best of our knowledge. We expect that oversampling the minority class would prove useful, as it did for classification tasks. Alternatively, one might consider gradient boosting techniques. These can be either combined with a probabilistic classifier or used with classification trees which are subsequently calibrated.

Pertaining to the performance of the machine learning methods, we emphasise that the SVM performed better on the real-life Trisomy data than in our simulations. This could point to an issue of our implementation of the SVM. Specifically, the simulations only test the performance of the SVM with a Gaussian kernel. Though we optimised hyperparameters in each iteration, it could be the case that better results were obtained if we tried multiple kernels, which is what one would do in practice.

6.2 Future Research

Perhaps the most important result of this research pertains to the BY-EN estimator. Our simulations indicate that it can serve as an effective probabilistic forecasting tool for contaminated, high-dimensional data, but that it is inhibited by its reweighting step. We show that the outlier detection rule, which is successfully used by robust estimators such as LTS for the low-dimensional setting, has unexpected effects in high-dimensional settings. In future work, it would be worthwhile to investigate how one could deal with this challenge. A better tuning approach could make the estimator a reliable probabilistic forecasting tool for all settings. Computational speedups are required, but this is less of a challenge. For example, there are hyperparameter tuning heuristics which are much faster than the brute-force grid search used currently.

7 Conclusion

This research investigated whether logistic regression can produce accurate class probability estimates when the data are high-dimensional. Our simulations showed that this is the case. As long as we employed suitable variable selection techniques such as the elastic net or best subset selection, we could produce calibrated and sharp forecasts. This result also held up when we applied the estimators to real datasets.

We further investigated the performance of the estimators in the presence of contamination. We contaminated 5% of our simulated data, which was sufficient to break all estimators based on classical logistic regression. We compared the performance of two different approaches to robust logistic regression with high-dimensional data, which were respectively introduced by Avella-Medina and Ronchetti (2017) and Kurnaz et al. (2018b). Due to numerical issues, we were not able to investigate the performance of the former's proposal in depth, but the latter's estimator proved promising. Surprisingly, our simulations showed that their estimator was able to produce accurate forecasts when the data were high-dimensional and contaminated, but not when the data are low-dimensional and uncontaminated. We investigated this issue in more detail and found that the estimator struggles with poor optimisation of the hyperparameters of its elastic net penalty. This explained why the estimator's forecasting performance differed on a case-by-case basis in our simulations. Future research may attempt to resolve this issue by improving the estimator's reweighting step.

A final question that we sought to answer in this research is how the logistic regression methods compared to that of popular machine learning methods. Based on the work of Niculescu-Mizil and Caruana (2005), we used two machine learning algorithms, neural networks and support vector machines. Neural networks are naturally a probabilistic classifier and therefore did not require any output processing, while support vector machines rely on calibration techniques. In our simulations, neither method produced calibrated forecasts, though neural networks exhibited a greater degree of robustness than support vector machines, which rely on maximum likelihood estimation for calibration. When we applied the machine learning methods to real data, support vector machines proved competitive with the regularised logistic regression estimators. This could indicate that the support vector machines' performance in the simulations was hampered by algorithm choices, specifically the use of a Gaussian kernel. In practice, one should always try multiple kernel functions.

8 References

Alfons, A., Croux, C., and Gelper, S. (2013). Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 7(1):226–248.

Ali, A. and Tibshirani, R. J. (2019). The generalized lasso problem and uniqueness. *Electronic Journal of Statistics*, 13(2):2307–2347.

Alqallaf, F., Van Aelst, S., Yohai, V. J., and Zamar, R. H. (2009). Propagation of outliers in multivariate data. *Annals of Statistics*, 37(1):311–331.

Amato, U., Antoniadis, A., De Feis, I., and Gijbels, I. (2020). Penalised robust estimators for sparse and high-dimensional linear models. *Statistical Methods & Applications*.

Avella-Medina, M. and Ronchetti, E. (2015). Robust statistics: a selective overview and new directions. *WIREs Computational Statistics*, 7(6):372–393.

Avella-Medina, M. and Ronchetti, E. (2017). Robust and consistent variable selection in high-dimensional generalized linear models. *Biometrika*, 105(1):31–44.

Bergmeir, C. and Benítez, J. (2019). RSNNS: Neural networks using the Stuttgart neural network simulator (SNNS). R package version 0.4-12, URL https://cran.r-project.org/web/packages/RSNNS/index.html.

Bertsimas, D., King, A., and Mazumder, R. (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852.

Bianco, A. M. and Yohai, V. J. (1996). Robust estimation in the logistic regression model. In Rieder, H., editor, *Robust Statistics, Data Analysis, and Computer Intensive Methods*, volume 109 of *Lecture Notes in Statistics*, pages 17–34. Springer, New York, NY.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Publishing Company Inc., 1st edition.

Cantoni, E. and Ronchetti, E. (2001). Robust inference for generalized linear models. *Journal of the American Statistical Association*, 96(455):1022–1030.

Cerqueira, V., Pinto, F., Sà, C., and Soares, C. (2016). Combining boosted trees with metafeature engineering for predictive maintenance. In Boström, H., Knobbe, A., Soares, C., and Papapetrou, P., editors, *Advances in Intelligent Data Analysis XV. IDA 2016. Lecture Notes in Computer Science*, volume 9897, page 393–397. Springer, Cham.

Croux, C., Flandre, C., and Haesbroeck, G. (2002). The breakdown behavior of the maximum likelihood estimator in the logistic regression model. *Statistics & Probability Letters*, 60(4):377 – 386.

Croux, C. and Haesbroeck, G. (2003). Implementing the Bianco and Yohai estimator for logistic regression. *Computational Statistics & Data Analysis*, 44(1):273 – 295.

Debruyne, M., Höppner, S., Serneels, S., and Verdonck, T. (2019). Outlyingness: Which variables contribute most? *Statistics and Computing*, 29:707–723.

Dedieu, A., Hazimeh, H., and Mazumder, R. (2020). Learning sparse classifiers: Continuous and mixed integer optimization perspectives. *arXiv preprint arXiv:2001.06471*.

Dietterich, T. G., Jain, A., Lathrop, R., and Lozano-Perez, T. (1994). A comparison of dynamic reposing and tangent distance for drug activity prediction. *Advances in Neural Information Processing Systems*, 6:216–223.

Donoho, D. and Montanari, A. (2016). High dimensional robust M-estimation: asymptotic variance via approximate message passing. *Probability Theory and Related Fields*, 166:935–969.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, J. and Peng, H. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *The Annals of Statistics*, 32(3):928–961.

Filzmoser, P., Höppner, S., Ortner, I., Serneels, S., and Verdonck, T. (2020). Cellwise robust M regression. *Computational Statistics & Data Analysis*, 147:106944.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Gijbels, I. and Vrinssen, I. (2015). Robust nonnegative garrote variable selection in linear regression. *Computational Statistics & Data Analysis*, 85:1 – 22.

Gneiting, T. and Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2nd edition.

Hastie, T., Tibshirani, R., and Tibshirani, R. J. (2017). Best subset, forward stepwise, or lasso? Analysis and recommendations based on extensive comparisons. *Statistical Science (to appear)*.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Number 143 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Hazimeh, H. and Mazumder, R. (2019). Fast best subset selection: Coordinate descent and local combinatioral optimization algorithms. *Operations Research (to appear)*.

Hazimeh, H. and Mazumder, R. (2021). L0Learn: Fast algorithms for best subset selection. R package version 1.2.0, URL https://cran.r-project.org/src/contrib/Archive/L0Learn/.

Heaton, J. (2008). *Introduction to Neural Networks for Java, 2nd Edition*. Heaton Research, Inc., 2nd edition.

Higuera, C., Gardiner, K. J., and Cios, K. J. (2015). Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PLOS One*, 10(6):1–28.

Huang, J., Ma, S., and Zui, C.-H. (2008). The iterated lasso for high-dimensional logistic regression. Technical Report 392, The University of Iowa, Department of Statistics and Actuarial Science.

Khan, J. A., Aelst, S. V., and Zamar, R. H. (2007). Robust linear model selection based on least angle regression. *Journal of the American Statistical Association*, 102(480):1289–1299.

Kong, D., Bondell, H. D., and Wu, Y. (2018). Fully efficient robust estimation, outlier detection and variable selection via penalized regression. *Statistica Sinica*, 28(2):1031–1052.

Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer-Verlag, New York, NY, USA, 1st edition.

Kurnaz, F. S., Hoffmann, I., and Filzmoser, P. (2018a). enetLTS: Robust and sparse methods for high dimensional linear and logistic regression. R package version 0.1.0, URL https://cran.r-project.org/web/packages/enetLTS/index.html.

Kurnaz, F. S., Hoffmann, I., and Filzmoser, P. (2018b). Robust and sparse estimation methods for high-dimensional linear and logistic regression. *Chemometrics and Intelligent Laboratory Systems*, 172:211 – 222.

Leung, A., Zhang, H., and Zamar, R. (2016). Robust regression estimation and inference in the presence of cellwise and casewise contamination. *Computational Statistics & Data Analysis*, 99:1 – 11.

Li, G., Peng, H., and Zhu, L. (2011). Nonconcave penalized m-estimation with a diverging number of parameters. *Statistica Sinica*, 21(1):391–419.

Machkour, J., Muma, M., Alt, B., and Zoubir, A. M. (2020). A robust adaptive lasso estimator for the independent contamination model. *Signal Processing*, 174:107608.

Maronna, R. A. (2011). Robust ridge regression for high-dimensional data. *Technometrics*, 53(1):44–53.

Maronna, R. A., Martin, R. D., Yohai, V. J., and Salibián-Barrera, M. (2019). *Robust Statistics: Theory and Methods (with R)*. Wiley, 2nd edition.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., and Lin, C.-C. (2021). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (formerly: E1071), TU Wien. R package version 1.7-6, URL https://cran.r-project.org/web/packages/e1071/index.html.

Neykov, N., Filzmoser, P., and Neytchev, P. (2014). Ultrahigh dimensional variable selection through the penalized maximum trimmed likelihood estimator. *Statistical Papers*, 55:187–207.

Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 625–632. Association for Computing Machinery.

Öllerer, V., Alfons, A., and Croux, C. (2016). The shooting s-estimator for robust regression. *Computational Statistics*, 31:829–844.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Smola, A. J., Bartlett, P., Schölkopf, B., and Schuurmans, D., editors, *Advances in Large Margin Classifiers*, pages 61–75. MIT Press.

Pregibon, D. (1981). Logistic regression diagnostics. *The Annals of Statistics*, 9(4):705–724.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880.

Rousseeuw, P. J. (1985). Multivariate estimation with high breakdown point. In Grossmann, W., Pflug, G., Vincze, I., and Wertz, W., editors, *Mathematical Statistics and Applications*, volume B, pages 283–297. Reidel Publishing Company, Dordrecht.

Rousseeuw, P. J. and Van Den Bossche, W. (2018). Detecting deviating data cells. *Technometrics*, 60(2):135–145.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys.* John Wiley & Sons.

Salehi, F., Abbasi, E., and Hassibi, B. (2019). The impact of regularization on high-dimensional logistic regression. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 12005–12015. Curran Associates, Inc.

She, Y. and Owen, A. B. (2011). Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*, 106(494):626–639.

Smucler, E. and Yohai, V. J. (2017). Robust and sparse estimators for linear regression models. *Computational Statistics & Data Analysis*, 111:116 – 130.

Sun, H., Cui, Y., Gao, Q., and Wang, T. (2020). Trimmed lasso regression estimator for binary response data. *Statistics & Probability Letters*, 159:108679.

Sur, P. and Candès, E. J. (2019). A modern maximum-likelihood theory for high-dimensional logistic regression. *Proceedings of the National Academy of Sciences*, 116(29):14516–14525.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Methodological)*, 73(3):273–282.

Van Aelst, S., Vandervieren, E., and Willems, G. (2011). Stahel-donoho estimators with cellwise weights. *Journal of Statistical Computation and Simulation*, 81(1):1–27.

Van Buuren, S. and Groothuis-Oudshoorn, K. (2021). mice: Multivariate imputation by chained equations. R package version 3.13.0, URL https://cran.r-project.org/web/packages/mice/index.html.

Wang, H., Li, G., and Jiang, G. (2007). Robust regression shrinkage and consistent variable selection through the lad-lasso. *Journal of Business & Economic Statistics*, 25(3):347–355.

Wedderburn, R. W. M. (1974). Quasi-likelihood functions, generalized linear models, and the Gauss—Newton method. *Biometrika*, 61(3):439–447.

Weiss, G. M. (2004). Mining with rarity: A unifying framework. *SIGKDD Explorations Newsletter*, 6(1):7–19.

Yang, M., Xu, L., White, M., Schuurmans, D., and Yu, Y.-l. (2010). Relaxed clipping: A global training method for robust regression and classification. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 2532–2540. Curran Associates, Inc.

Yohai, V. J. (1987). High breakdown-point and high efficiency robust estimates for regression. *The Annals of Statistics*, 15(2):642–656.

Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 694–699. Association for Computing Machinery.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. (2018). High-dimensional classification. In Härdle, W., Horng-Shing Lu, H., and Shen, X., editors, *Handbook of Big Data Analytics*, pages 225–261. Springer Publishing Company Inc.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Methodological)*, 67(2):301–320.

Zou, H. and Zhang, H. H. (2009). On the adaptive elastic-net with a diverging number of parameters. *The Annals of Statistics*, 37(4):1733–1751.

A Appendix: List of Acronyms

AdaEN Adaptive elastic net. 8

Adalasso Adaptive lasso. 7

BSS Best subset selection. 9

BY-EN Bianco-Yohai estimator with elastic net penalty (enetLTS estimator). 18

CLR-AdaEN Classical logistic regression with adaptive elastic net penalty. 20

CLR-BSS Classical logistic regression with best subset selection. 20

CLR-BSSL Classical logistic regression with best subset selection and lasso penalty. 20

CLR-EN Classical logistic regression with elastic net penalty. 20

CR Cantoni-Ronchetti. 29

CR-Lasso Cantoni-Ronchetti estimator with lasso penalty. 32

CV Cross-validation. 7

DDC DetectDeviatingCells. 13

EN Elastic net. 8

IRLS Iteratively reweighted least squares. 13

LTS Least trimmed squares. 10

RBF Radial basis function. 20

RCLR Regularised classical logistic regression. 20

RRLR Regularised robust logistic regression. 16

SVM Support vector machines. 20