Statistical Model Selection and Prediction for Non-standard Data: Insights and Applications in Economics and Finance

Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

von der KIT-Fakultät für
Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

M.Sc. Konstantin Görgen

Tag der mündlichen Prüfung: 28.07.2022

Referentin: Prof. Dr. Melanie Schienle
Korreferent: Prof. Dr. Christian Conrad

Karlsruhe, 29.07.2022

Acknowledgements

First of all, I would like to express my deepest appreciation to Prof. Dr. Melanie Schienle for her guidance and support during my time as a PhD-student. You motivated me to pursue a PhD in the first place and provided an excellent work environment. I could always rely on you to discuss my research as well as other issues in my PhD and you constantly provided me with motivation and advice. I am also thankful to Prof. Dr. Christian Conrad for serving as the co-advisor in my defense, and supporting my research in the previous years. Furthermore, I would like to thank Prof. Dr. Fabian Krüger for giving me valuable feedback and the interesting research discussions, and to Prof. Dr. Martin Klarmann for chairing my thesis defense. In addition, I would like to extend my sincere thanks to Prof. Dr. Kyusang Yu for supporting my research visit in Seoul as well as the Karlsruhe House of Young Scientists (KHYS) for sponsoring my time there.

I could not have undertaken this journey without the growing number of all my colleagues and friends who made my time at the Chair of Statistical Methods and Econometrics invaluable. Specifically, I would like to thank Dr. Rebekka Buse for guidance in the beginning of my thesis, Nils Koster for proof-reading and for interesting discussions both in and outside of research, Lotta Rüter for providing a great atmosphere to our daily work environment, Lora Pavlova for making the Mannheim graduate classes much easier and enjoyable, Dr. Johannes Bracher for supporting my research and for our fruitful cooperation on the Covid-19 Forecast Hub, and Dr. Sebastian Lerch, Jieyu Chen, Nina Horat, and Daniel Wolffram. Additionally, I would like to thank Jannik Deuschel, Jonas Meirer and Davide Hailer for excellent research assistance. Thanks should also go to Dr. Marc Schmidt for interesting talks about PhD-life and for making the daily "home-office" life so pleasant, and to Tobias Wetterich for the interesting talks about statistical methods from an industry perspective.

Finally, I am deeply indebted to my family for their unconditional support and motivation throughout my studies: To my sister Ellinor Görgen, my father Dr. Gisbert Görgen, and my grandmother Ursula Wyneken for motivating me and giving me stability and emotional support. I am extremely grateful to my mother, Dr. Alexandra Wyneken-Görgen, who always believed in me, thus giving me strength and the confidence to pursue a PhD in the first place. Words cannot express my gratitude to Eleanor McSweeney. Thank you for always being there for me, especially during difficult times, and for your unequivocal support in such moments.

Contents

List of Figures

List of Tables

List of Abbreviations

ADF (test)	Augmented Dickey-Fuller (test)
AoE	Actual over Expected Exceedances
ASDP	Approximate Semi-Definite Program Algorithm
CART	Classification and Regression Trees
CAV	Conditional Autoregressive Value at Risk Method
CNN	Convolutional Neural Network
CPA	Conditional Predictive Ability
EDF	Effective Degrees of Freedom
ESB	Empire State Building
FC	Fully Connected
FDR	False Discovery Rate
FDP	False Discovery Proportion
FPR	False Positive Rate
FRED	Federal Reserve Economic Data
fREML	fast Restricted Maximum Likelihood
GAM	Generalized Additive Model
GRF	Generalized Random Forest
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
JFK	John F. Kennedy International Airport
KPSS (test)	Kwiatkowski-Phillips-Schmidt-Shin (test)
LCD	Lasso Coefficient Difference
LGD	Loss Given Default
LSM	Lasso Signed Max
MAD	Mean Absolute Deviation
MAE	Mean Absolute Error
ML	Machine Learning
MSD	Mean Squared Deviation
MSE	Mean Squared Error
NYC	New York City
NYPD	New York City Police Department
OLS	Ordinary Least Squares
PC	Principal Component
PCA	Principal Component Analysis
PIRLS	Penalized Iterative Reweighted Least Squares
QR	Quantile Regression
QRF	Quantile Regression Forest
RMSE	Root Mean Squared Error
SAV	Symmetric Absolute Value
SD	Standard Deviation
SE	Standard Error
SOEP	Socio-Economic Panel
TPR	True Positive Rate
TRACE	Trade Reporting and Compliance Engine
USD	US-Dollar

1 Introduction

In an increasingly digital world, data has become abundant and research about leveraging this amount of data is on the rise (Blei and Smyth, 2017; Dhar, 2013). Such data can be used for measuring success or failure in business, for targeting customers, to influence public policies, or more generally, to inform any decision-maker (cp. Athey, 2017). Researchers, however, face additional challenges when trying to extract information from such often unstructured, noisy, and incomplete data, for example, to understand and identify driving factors of an economic indicator or policy (Bareinboim and Pearl, 2016). This includes highly correlated, time-dependent data, combinations of unstructured data, and even high-dimensional situations, where we have very few data points and many potentially relevant factors. While organizations are beginning to see the value of this new, non-standard data, the methods to analyze and draw conclusions from it have not yet been fully adapted (see e.g. Einav and Levin, 2014).

In this thesis, I tackle the above challenges by developing interpretable statistical machine learning methods to reveal important effects of public policies, to better assess risks in financial applications, and to quantify market drivers, for example of house prices. I study causal inference, statistical model selection, and prediction in different social and economic contexts. More specifically, I concentrate on uncovering statistical relationships while considering the underlying uncertainty in the data, and on identifying important contributing factors for such relationships. In the first part (Chapters 2 and 3) of my work, I analyze financial risk with cryptocurrencies and corporate bonds. For the former, I identify classes of assets and time periods where flexible machine learning methods, such as random forests employed within an interpretable statistical framework, significantly improve predictability of risk. This is vital given the highly volatile return structure of cryptocurrencies. For corporate bonds, I uncover drivers of the risk of default by developing robustified version of the knockoff framework (Candès et al., 2018), which is

able to correctly handle the underlying, highly correlated time series data. Additionally, focusing on important selected factors improves the predictability of default events while retaining interpretability. In the second part (Chapter 4), focus lies on the evaluation of the causal effect of tuition fees on university student enrollment. I develop methods to deal with the many possible influencing factors given only few observations by combining subsampling-based methods with regularization in a panel setup. I can show that there was a causal effect of the short tuition fee period in Germany by disentangling this effect from other factors and policies. In the third part (Chapters 5 and 6), satellite images are combined with many noisy, observational data sources to show the impact of crime on the housing market of New York City on a spatial grid. To overcome the endogeneity of crime for house prices, I develop a method that leverages satellite data, can be easily extended to other cities, and highlights the non-linearity of crime on a spatial level.

On a more detailed level, the contributions of each chapter are as follows. Chapter 2 studies the estimation and prediction of the risk measure Value at Risk for cryptocurrencies. In contrast to classic assets, their returns are often highly volatile and characterized by large fluctuations occurring at single events. Analyzing 105 major cryptocurrencies, I show that Generalized Random Forests (GRF) (Athey et al., 2019), which can be adapted to specifically fit the framework of quantile prediction, have superior performance over other established methods such as quantile regression and CAViaR. This is particularly visible in unstable times and for classes of highly-volatile cryptocurrencies. Furthermore, I identify important predictors during such times and show their influence on forecasting over time. Finally, the small-sample prediction properties in comparison to standard techniques are investigated in a comprehensive Monte Carlo simulation study.

In Chapter 3, I focus on the recent global financial crisis[1] with the default of large corporate bonds. This highlights the importance of detecting macroeconomic factors that drive recovery rates of such bonds. I propose a purely data-driven method that transparently and robustly identifies such relevant factor groups despite the strong time dependence in the large cross-section of recovery rates. The suggested knockoff-type technique has its focus on detecting interpretable drivers of recovery rates by controlling the proportion of false discoveries. Moreover, I also show that out-of-sample, the resulting

[1]The crisis starting in the USA in 2007

sparse model has similar predictive power to state-of-the-art machine learning models that use the entire set of predictors.

Chapter 4 considers the empirical evaluation of the fixed, flat, short-period German university tuition fee episode. This evaluation is challenging due to different implementation decisions across federal states, few official observations, migration effects and many potentially influential controls. My transparent data-driven model selection approach robustly controls for correlation of the policy decision with state characteristics when the enrollment rate response is only measured with noise. Using this selection approach, I find a significant and substantial negative causal impact of fees on enrollment and substantial migration effects. This is contrary to findings in the literature that are based on ad-hoc covariate choices and no spatial effects.

Chapter 5 and 6 deal with developing and applying a model to assess house prices. The main contribution consists of connecting image data to observational data using deep learning methods to better understand drivers of prices and to predict such prices in New York City. In Chapter 5, I was able to detect from the input images which factors have a strong impact on the final price using a transfer learning approach that extracts information from images linked with very accurate crime data. From this trained neural network model, I could extract features and build a model to predict property prices in the cross-section. With this transfer learning approach, I combine the best of two worlds: the socio-economic link between crime and property values for prediction, and machine learning methods that are able to extract information from images. We show how easily one can scale the trained model to other cities with the example of Philadelphia. In Chapter 6, the focus lies on detecting and measuring the influence of each extracted crime feature for house prices. I therefore extend the model of Chapter 5 and tailor it to allow for the identification of interpretable factors. Applying generalized additive models, I clearly identify important drivers of prices and detect heterogeneities in their influence, depending on the location and the variables themselves. I can visualize the impact of different factors on prices, clarify how the features correspond to crime, and further show that this model maintains excellent forecasting performance as in Chapter 5.

Chapter 2 is joined work with Jonas Meirer and Melanie Schienle. Chapter 3 is joint work with Abdolreza Nazemi and Melanie Schienle and has been submitted to

Management Science. Chapter 4 is joint work with Melanie Schienle, has been presented at the 6th IAAE Conference at the University of Cyprus[2], the Statistical Week 2019 at the University of Trier, and has been submitted to Econometric Reviews with the response to revise and resubmit. Chapter 5 is joint work with Jannik Deuschel and Melanie Schienle and has been presented at the 6th HKMetrics-Workshop at the University of Mannheim. Chapter 6 is joint work with Melanie Schienle and Kyusang Yu.

[2]6th Annual Conference of the International Association for Applied Econometrics

2 Predicting Value at Risk for Cryptocurrencies Using Generalized Random Forests

2.1 Introduction

Cryptocurrencies are an important and rising part of today's digital economy. Currently, the market capitalization of the top 10 cryptocurrencies in the world is close to $2 trillion and growing[1]. The use of cryptocurrencies in terms of daily volume exploded from 2016 to 2018[1], which not only attracts individuals but also business users such as hedge funds, merchants and long-term investors such as crypto-focused as well as traditional investment funds (Vigliotti and Jones, 2020). However, the crypto asset market remains highly volatile. An investment in Bitcoin in 2013 would have seen a return of roughly $20,000\%$ in 2017, but an investment in 2017 would have led to a performance of -75% in 2019[1]. Consequently, there is a need to monitor the inherent volatility to manage the risks associated with cryptocurrencies. To address this, we find that classic approaches such as the historical simulation or CaViaR methods are too restrictive. More general non-linear methods provide more flexibility to account for such behavior which could be caused in part by speculators.

In this paper we propose a novel way for out-of-sample prediction of the Value at Risk, one of the standard and mostly used risk measures in practice. We use a quantile version of Generalized Random Forests (GRF, see Athey et al., 2019), which builds upon standard random forests (Breiman, 2001) and extends them to fit quantiles as opposed to the mean in standard ones. This framework shows to be especially promising when dealing

[1]See e.g. `https://coinmarketcap.com/charts/`; accessed at 22nd March 2022.

with more volatile classes of cryptocurrencies due to the non-linear structure of their returns. In a comprehensive out-of-sample scenario using more than 100 of the largest cryptocurrencies, GRF outperforms other established methods such as CAViaR (Engle and Manganelli, 2004), quantile regression (Koenker and Hallock, 2001) or GARCH-models (Bollerslev, 1986; Glosten et al., 1993) over a rolling window, particularly in unstable times. This can be attributed to the nonparametric approach of random forests that is flexible and adaptable considering important factors and non-linearity. We further analyze performance in different important subperiods, consider different classes of cryptocurrencies, and employ different sets of covariates with the forest-based methods and the benchmark procedures.

Previous studies have confirmed that there exist speculative bubbles (Cheah and Fry, 2015; Hafner, 2020), and we find that our approach assesses risks especially well during such times. Moreover, we account for a large number of covariates that describe volatility, liquidity, and supply (Liu and Tsyvinski, 2020). It can be seen that variable importance differs substantially depending on time, where long-term measures of standard deviation, that are an important predictor in stable times, are not relevant predictors for VaR in unstable, volatile times. Furthermore, only few of the additional covariates beside lagged standard deviations and lagged returns are relevant. We find that for other, less volatile classes of cryptocurrencies such as stablecoins, especially GJR-GARCH models and quantile regression can compete with GRF.

Our paper contributes to the growing literature on cryptocurrencies. Analyses performed in the past include GARCH models (Chu et al., 2017) as well as ARMA-GARCH models (Platanakis and Urquhart, 2019), approaches using RiskMetrics (Pafka and Kondor, 2001) and GAS-models (Liu et al., 2020), application of extreme value theory (Gkillas and Katsiampa, 2018), vine copula-based approaches (Trucios et al., 2020), Markov-Switching GARCH models (Maciel, 2020), non-causal autoregressive models (Hencic and Gouriéroux, 2015) and also some machine learning based approaches (see e.g. Takeda and Sugiyama (2008)). Additionally, cryptocurrencies can be used for diversification in investment strategies with other, traditional assets (see. e.g Trimborn et al. (2020); Petukhina et al. (2021)), as the correlation between them and more established assets tends to be low (Elendner et al., 2017; Platanakis and Urquhart, 2019). This

again poses the question of assessing the risks of cryptocurrencies, where new methods of addressing the above mentioned challenges need to be explored.

The paper is structured as follows. Section 2.2 presents the underlying data and cryptocurrencies we use in our analysis. In Section 2.3, we introduce the main methods used to analyze the data, specify VaR and present the evaluation tests and framework. Section 2.4 demonstrates the performance of the different methods under various data generating processes in a thorough simulation study. Results for the data are shown in Section 2.5, where we compare the performance of methods directly over all currencies and using classic backtests in Section 2.5.1. In Section 2.5.2, we analyze representative, important currencies in detail and look at their variable importance. Finally, we conclude in Section 2.6.

2.2 Data

We use daily log-returns of 105 of the largest cryptocurrencies[2] from coinmetrics by market capitalization[3], with their value compared to US-Dollar (USD), in the period from 07/2010 to 03/2022. Depending on the currency (i.e. the date of creation), the number of available observations varies between 261 and 4264. The data is summarized in Table 2.1. The coinmetrics dataset includes spot-market information from 30 different exchanges, such as Binance, ZB.COM, FTX, OKX, Coinbase, KuCoin, or Kraken[4]. We see that there are some very negative and positive returns in the data, as well as high excess-kurtosis confirming the observations in the depicted quantiles. Furthermore, there are some assets with a high skewness, both positive and negative, indicating asymmetry in the distribution of returns.

We do not detect any stochastic non-stationarities in the data, which is supported by Augmented Dickey-Fuller (ADF) tests and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests (Kwiatkowski et al., 1992). With Alpha Finance Lab (alpha), Polymath (poly),

[2]The data was obtained on 23rd March 2022 from `https://docs.coinmetrics.io/` using the community data set, which can be downloaded from a public Github repository at `https://github.com/coinmetrics/data/`.

[3]All currencies have a maximum market capitalization of more than 15 million USD each.

[4]See `https://docs.coinmetrics.io/exchanges/all-exchanges` for an overview of all exchanges included.

and Synthetix (snx), KPSS tests against level stationarity seem slightly significant, while trend KPSS tests and ADF tests suggest stationarity. With Algorand (algo), Binance Coin (bnb), Curve DAO Token (crv), FTX Token (ftt), Internet Computer (icp), Aave (lend), OMG Network (omg), SushiSwap (sushi), and Monero (xmr), KPSS tests against trend stationarity are slightly significant, while ADF tests and level KPSS tests again suggest stationarity. All results of the stationarity tests can be found in Table 2.10 in the appendix.

Table 2.1: Descriptive Statistics of Log-Returns of Cryptocurrencies

	Min	1%	5%	Median	95%	99%	Max	Skewness	Excess-Kurtosis	Standard Deviation	Observations
Min%	-1.264	-0.308	-0.168	-0.006	0.001	0.002	0.006	-3.556	1.671	0.001	261.000
1%	-1.120	-0.268	-0.158	-0.006	0.001	0.002	0.006	-2.691	1.965	0.001	314.960
5%	-0.830	-0.243	-0.139	-0.004	0.002	0.006	0.016	-1.195	3.043	0.002	526.200
25%	-0.576	-0.199	-0.114	-0.001	0.089	0.170	0.342	-0.190	6.482	0.060	717.000
50%	-0.492	-0.181	-0.104	0.000	0.109	0.209	0.461	0.298	10.708	0.073	1354.000
75%	-0.362	-0.160	-0.086	0.001	0.120	0.239	0.704	1.030	23.164	0.080	1669.000
95%	-0.022	-0.006	-0.002	0.002	0.153	0.318	1.258	2.241	75.697	0.101	2851.800
99%	-0.006	-0.002	-0.001	0.003	0.197	0.412	1.431	3.472	160.129	0.117	3264.120
Max%	-0.005	-0.002	-0.001	0.003	0.201	0.539	1.462	3.753	216.567	0.136	4264.000

Notes: The rows show the quantiles of the sample-measures in the columns that are for log-returns of all cryptocurrencies combined. For example, over all cryptocurrencies, there is a median of 1354 observations per cryptocurrency.

For the cryptocurrencies, Figure 2.1 illustrates the median returns (black) over all cryptocurrencies by date. We can see that in the beginning of the time period, only one currency, namely Bitcoin, was present in the data set. From 2014, we see an incremental increase (red line), while there is a jump up in 2017 and an consecutively faster increase in available cryptocurrencies. We also see that the returns are very different between currencies from 2014 to 2018 (blue), which marks a period of hype leading to a crash in the beginning of 2018. Later, there a large negative spikes corresponding to the many waves of the Covid-19 pandemic. Based on these observations, we divide our data into three periods. The first period ranges from August 2015 (2015-08-22) to the end

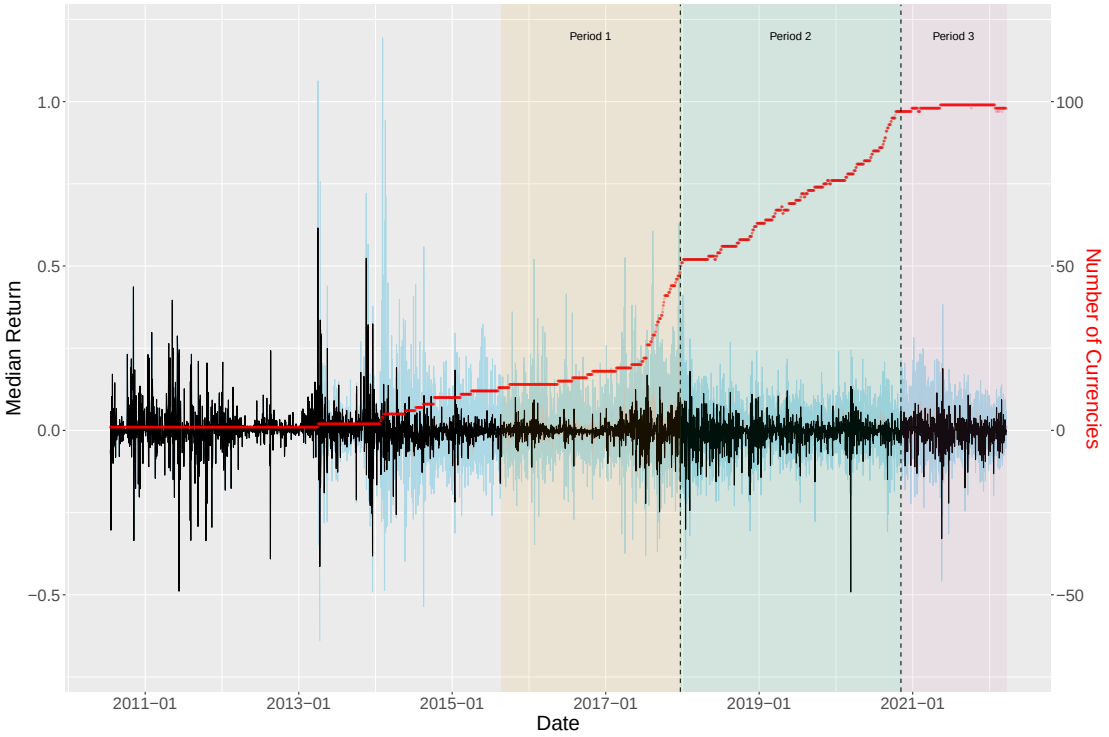


Figure 2.1: Pointwise Median Returns Over All Currencies With 5% And 95% Sample Quantiles in Blue Over Each Date

Notes: The number of currencies that have a return at the time is indicated in red. The orange, green, and purple shaded areas correspond to the three periods we analyze closer in Section 2.5.

of 2017 (2017-12-21), while the second period subsequently lasts until November 2020 (2020-11-05). The last period then covers the rest of our data (until 2022-03-20)[5].

Since we are in a time series setup, we include classic covariates based on lagged return in our analysis. Additionally, we employ information specific to each cryptocurrency in 7 external covariates. The five time-series based covariates consist of the one-day lagged return and the lagged 3,7,30, and 60 day return standard deviation. The external covariates are the number of unique active daily addresses (Active_Users), the number of unique addresses that hold any amount of native units of that currency or at least 10 or 100 USD equivalent (Total_Users, Total_Users_USD10,Total_Users_USD100),

[5]The specific dates account for the training periods of currencies and creation of new assets to make sure that we capture a maximum number of cryptocurrencies in each time period.

the supply equality ratio (SER), i.e. the ratio of supply held by addresses with less than $1/10^7$ of the current supply to the top 1% of addresses with the highest current supply, the number of initiated transactions (Transactions), and the velocity of supply in the current year (Velocity), which describes the the ratio of current supply to the sum of the value transferred in the last year. See also Table 2.13 in the Appendix for details on the covariates.

Table 2.2: Summary of Covariates for Different Time Periods

Quantile	Ret	Active_Users	Total_Users	Total_Users_USD100	Total_Users_USD10	SER	Transactions	Velocity	sd_3	sd_7	sd_30	sd_60	CapMrktCurMUSD
Period 1: 5 Currencies													
5%	-0.094	7691	233361	7768	32688	0.001	3304	9.321	0.007	0.014	0.022	0.026	17
Median	-0.001	127679	2579591	297029	615602	0.013	49170	33.708	0.036	0.044	0.051	0.055	2611
95%	0.104	324662	9697275	3562101	5930594	0.031	110698	109.281	0.137	0.134	0.123	0.124	185407
Period 2: 15 Currencies													
5%	-0.085	3592	95320	5271	16960	0.001	16618	4.668	0.008	0.015	0.024	0.027	70
Median	-0.001	66921	2629309	233835	540993	0.008	128481	16.244	0.034	0.041	0.047	0.049	5008
95%	0.094	179852	8601014	2135275	4469914	0.018	655221	67.849	0.122	0.119	0.111	0.109	99979
Period 3: 77 Currencies													
5%	-0.082	2138	533215	15647	41396	0.002	25250	5.255	0.008	0.017	0.026	0.030	233
Median	-0.000	19371	1218761	80236	215034	0.007	137082	15.325	0.035	0.041	0.047	0.049	1794
95%	0.086	76081	2793940	565048	1195791	0.012	424953	53.743	0.111	0.105	0.101	0.096	26105
Full Data: 105 Currencies													
5%	-0.089	1857	559862	16964	44887	0.002	19135	5.626	0.009	0.019	0.028	0.033	461
Median	-0.000	15406	1095198	72651	186751	0.006	106286	14.371	0.038	0.045	0.051	0.054	2367
95%	0.095	62448	2389786	434200	917576	0.010	339294	44.627	0.120	0.113	0.108	0.102	21085

Notes: Values are means of quantiles over all assets contained in the specific time period. CapMrktCurMUSD describe the market cap in Mio. USD and is not included as a covariate due to multicollinearity reasons.

Table 2.2 gives an overview of the employed covariates and their values in the different time periods. We can see that in Period 1, we have the most extreme returns on average as well as the most extreme lagged standard deviations. This is not surprising looking at the first period (853 days), which arguably marks the most volatile period, with many new currencies being created, as well as the second longest period. In the following period (1050 days), the average median market cap reaches a high as well as the number of users invested in the currencies, indicating that the market is growing while stabilizing more. This is followed by a sharp drop in the market cap for the last, shortest period (500 days), which starts at the beginning of the Covid-19 pandemic. There, the number of active users as well as the SER decreases, indicating that more smaller addresses are pushed off the market, while it is the period with the most currencies.

All in all, we therefore have three very distinct periods. The first one is characterized by a few, rapidly changing currencies and extreme returns and volatilities, while the

second period is less extreme and more characterized by a strong increase in median market caps. The third period, in the end, is very short but contains more than five times the number of currencies in comparison to the second period.

2.3 Methodology

For the prediction of cryptocurrencies, we advocate the use of non-linear machine learning based techniques. In this way, we intend to accommodate the documented large share of speculation (Ghysels and Nguyen, 2019; Baur et al., 2018; Selmi et al., 2018; Glaser et al., 2014) and resulting frequent changes in unconditional volatility which make predictions in this market peculiar. In particular, we focus on generalized random forest methods that are tailored for conditional quantiles of returns and thus allow to forecast the VaR. The flexible but interpretable non-linearity of the approach allows for a direct comparison to standard linear and (G)ARCH type models. We also argue that the difference in forecasting performance can moreover be employed to detect periods of bubbles and extensive speculation.

Recall that for daily log returns r_t the VaR_t at level $\alpha \in (0, 1)$ conditional on some covariates x_{t-1} is defined as

$$VaR_t^{\alpha}(x_{t-1}) = \sup_{r_t} \left(F(r_t|x_{t-1}) < \alpha \right) \ , \tag{2.1}$$

where F marks the distribution of r_t conditional on x_{t-1}. Generally, the conditioning variables could consist of past lagged returns, standard deviations but also external (market) information or other assets. We employ these as covariates that are explained in Section 2.2.

We propose the use of two different types of random forest based techniques which directly model the conditional VaR in (2.1). Both build on the classic random forest (Breiman, 2001) which is an ensemble of (decorrelated) decision trees (see e.g. Hastie et al. (2009)) for the mean of r_t. In a decision tree, each outcome r_t is sorted into leafs of the tree by binary splits. These splits are performed based on different x_{t-1} components falling above or below specific adaptive threshold values that need to be calculated, for example by the Gini Impurity or MSE-splitting in Classification and Regression Trees

(CART) (Breiman et al., 1984), or using other criteria. Finally, the prediction for a new r_t is a weighted version of each tree prediction.

In the proposed method that we employ, the generalized random forest (GRF) from on Athey et al. (2019), the random forest split criterion is adapted to mimic the task of quantile regression rather than minimizing a standard mean squared loss criterion for mean regression tasks. Intuitively, the splits in each leaf are conducted by minimizing the Gini-loss, which separates the returns r_t as best as possible at different quantiles. To transform the minimizing problem in the splits into a classification task, the response variable r_t is transformed in each split to obtain pseudo-outcomes $\rho_t = \sum_{k=1}^{K} 1\{r_t > \theta_k\}$, where $\Theta = (\theta_{q_1}, \ldots, \theta_{q_K})$ describe a set of K pilot-quantiles of r_t in the parent node. These quantiles with levels $\tau = q_1, \ldots, q_K$ are then used to calibrate the split[6]. In Athey et al. (2019), this is formally motivated by moment conditions and gradient approximations, but practically, r_t is relabeled to a nominal scale depending on the largest quantile it does not exceed. In a final step, the optimal split on a variable component p of x_{t-1} and $j = 1, \ldots, J$ observations in the parent node is then based on minimizing the above-mentioned Gini impurity criterion for classification. For a separation into two possible leaf sets $v = l, r$, the Gini impurity for one leaf v is $G_p^v = 1 - \sum_{k=1}^{K} p_{k,v}^2$, where $p_{k,v} = \sum_{j=1}^{J} 1\{\rho_j = k \text{ and } \rho_j \in v\}/|v|$ is the proportion of ρ_j in group v with value $k = 1, \ldots, K$. The full loss is then an average weighted by leaf size, yielding

$$G_p = (|l|G_p^l + |r|G_p^r)/(|l| + |r|) \ . \tag{2.2}$$

We choose the Gini-loss since it is fast and, for certain configurations, produces purer nodes than for example using entropy as a splitting criterion (see e.g Breiman, 1996). This can be particularly helpful when dealing with changing variance (and thus time-varying quantiles) of returns, where we would like to detect single extreme events. For our specific case of $\alpha = 0.05$, this implies that values larger than $\theta_{0.05}$ in the parent node are given the value 1, while others are 0. Algorithm 1 briefly summarizes the tree building algorithm from Athey et al. (2019) for the quantile version of GRF, where the main differences with regard to the splitting regime in comparison to a classic CART occur in every step the tree is grown.

[6] We use the tuning parameters $K = 1$ and $\tau = \alpha$, i.e. the level of VaR we are analyzing, in this classification pre-step.

In addition to that, the outcome that is predicted is not the mean but the α-quantile (i.e. VaR^{α}), which is done in a way that you do not calculate a weighted average of r_t but a weighted average of the empirical CDF $\hat{F}(r_t|x_{t-1}) = E[1_{\{r_t\}|x_{t-1}}]$. Intuitively, log returns r_t that have similar x_{t-1} in comparison to a new observation x_v receive higher weight in the empirical CDF. Similarity weights $w_t(x_v)$ are measured as the relative frequency on how often x_v falls in the same terminal leaf as x_{t-1}, for $t = 1, \ldots, T$, and averaged over all trees for each x_{t-1}. This last step was originally introduced by Meinshausen (2006) for random forests.

As a benchmark, we employ the quantile regression forest (QRF) based on Meinshausen (2006). This random forest, however, uses the same splitting regime as the original CART random forest and therefore does not account explicitly for situations where the variance and therefore the quantile changes, as splits are conducted based on a mean-squared error criterion. Since such volatility changes are to be expected for cryptocurrencies in our data, we expect GRF to perform better than QRF, but still include both in the analysis to see potential differences in predictions. Furthermore, GRF uses so-called "honest" trees, meaning that different data (usually the subsampled data for each tree is split again in half) is used for building and "filling" each of the trees with values.

As benchmarks we further include two types of standard time series methods. We use the CAViaR (CAV) methodology by Engle and Manganelli (2004) and standard quantile regression (Koenker and Hallock, 2001). Both make use of quantile regression (QR) techniques (Koenker and Bassett, 1978) that do not minimize the squared error as in ordinary regression, but use the check function $\rho_\alpha(u) = u\,(\alpha - 1\{u \leq 0\})$ to minimize $L_\alpha(f_\alpha(\cdot), x_t) = \sum_{t=1}^{T} \rho_\alpha\,(r_t - f_\alpha(x_t))$. For CAViaR, we use a symmetric absolute value (SAV) component for $f_\alpha(\cdot)$, i.e $f_\alpha(x_t, r_t) = \beta_1 + \beta_2 f_\alpha(x_{t-1}, r_{t-1}) + \beta_3|r_{t-1}|$, and $f_\alpha(r_{t-1}, x_t) = \beta_4' x_t + \beta_5 r_{t-1}$ for the quantile regression. In contrast to the former methods, they can only capture parametric (non-) linear effects which limits their flexibility.

For comparison we use a GJR-GARCH(1,1) model (Glosten et al., 1993), a GARCH(1,1)(-X) (Bollerslev, 1986) model, a simple historical simulation (Hist), meaning that we predict VaR^{α}_{t+1} at level α as the sample α-quantile of the preceding returns in a window of length K, i.e. $(r_{t-K+1}, \ldots, r_t)$, and one that fits a normal distribution to the sample data and uses the theoretical fitted α-quantile as the prediction for VaR^{α}_{t+1}

Algorithm 1 Generalized Random Forest - Tree Building

Input: Set of "honest", subsampled observations X_T and R_T; minimum node size n_m; quantile probabilities $\tau_K = (\tau_1, \ldots, \tau_K)$

Growing the tree Create root node P_0
Initialize queue Q with P_0
while Queue is not empty **do**
  Take the oldest element from Q (Parent Node P) and remove it from Q
  Take a random subsample of p variables index by set $P_{sub} = \{\tilde{1}, \ldots, \tilde{p}\}$ from X_T on which to potentially split and take observations $x_i^{(P_{sub})} = (x_i^{(\tilde{1})}, \ldots, x_i^{(\tilde{P})})$ from P.
  Set $loss = \infty$
  for h in 1 to p **do**
    Compute quantiles θ_k of r_t from parent node P at $\tau_1, \ldots, \tau_K$ and compute pseudo outcomes $\rho_t = \sum_{k=1}^{K} 1\{r_t > \theta_k\}$ for each $r_t \in P$.
    For each possible split point in $x^{(h)}$, compute the criterion from Equation (2.2)
    Save loss s_h that minimizes this splitting criterion
    Save optimal split point $split_h$
    if $s_h < loss$ **then**
      $loss \leftarrow s_h$
      $ind_h \leftarrow h$
    end if
  end for
  if Split on variable h with $split_{ind_h}$ succeeded (based on hyperparameters) **then**
    Determine children C_1, C_2 according to optimal split
    Add both children C_1, C_2 to a new daughter node each with corresponding observations left and add these to Q
  end if
end while

Output: One tree of the forest

(NormFit). We do not expect the latter to perform well as we have high skewness and excess-kurtosis in the data (see Table 2.1 in Section 2.2).

For the proposed random forest type and all benchmark procedures that allow for additional covariates (GRF, QRF, QR) we include lagged standard deviations (SD) in addition to the lagged level r_{t-1} in the model in order to capture the strongly varying levels of unconditional volatility in particular for the cryptocurrencies in the non-linear structure. Additionally, we also employ the above methods and the GARCH(1,1) model using both the latter covariates as well as the 7 external covariates described in Section 2.2. These methods are GRF-X, QRF-X, QR-X, and GARCH-X. To establish a fair common ground in used model complexity for the QR, QRF, GARCH-X, and GRF, we select a common set of different SD lags as covariates from an additional Monte-Carlo study. In this, we use the simple SAV-model from Section 2.4 as a baseline of which the respective specification is tailored to regime changes in the unconditional volatilities as observed form the cryptocurrencies. The model is essentially linear autoregressive of order 1 in the VaR and thus directly yields the VaRs as outputs. With this, it is possible to use a MSE-minimizing criterion and select the MSE-minimizing variables for the subsequent simulations and data analysis. Table 2.3 summarizes the results of this short simulation study, which is why we use 3, 7, 30, and 60 day lagged SD as covariates.

Table 2.3: MSE Prediction Error for Different Covariate Combinations

Lagged SD (in days)	3	7	30	3 and 7	3 and 30	7 and 30	3, 7 and 30	3, 7, 30, and 60
GRF-MSE	0.124	0.095	0.089	0.075	0.070	0.069	0.059	**0.057**

Notes: MSE prediction error for a simulated SAV-model as in Section 2.4 for GRF (QRF). The minimum MSE is marked in bold.

To compare the performance of the above methods, we use two types of evaluation approaches. First, we test how well each model predicts the conditional α-VaR over the entire out-of-sample horizon using three different sets of evaluation techniques. The simplest way of checking whether a model predicts VaR^{α} correctly over a time horizon is to look at its coverage meaning the number of times r_t is smaller than the predicted VaR_t^{α}. In a well calibrated model, this should be exactly αT times.

This measure is called Actual over Expected Exceedances (AoE) and is computed as $AoE_\alpha = 1/(\alpha T)\sum_{t=1}^{T} 1\{r_t < VaR_t^\alpha\}$. To test this intuition formally, we employ three tests, the DQ-test[7] from Engle and Manganelli (2004), the Christoffersen-test (Christoffersen, 1998) and the Kupiec-test (Kupiec, 1995). All three tests assume that under the null hypothesis the forecasts have correct coverage. The Kupiec-test is simply the formalization of the above intuition, the Christoffersen-test is robust against serial correlation by assuming that $g_t = 1\{r_t < VaR_t^\alpha\} \sim Bern(\alpha)$, and the DQ-test additionally accounts for problems with conditional coverage due to clustering of the hits exceedance sequences g_t with a regression-based approach. In the empirical analysis, we only report the values of the DQ-test, which is the strictest of the tests, for reasons of clarity. Results for the other tests do not differ substantially and are available upon request from the authors.

Secondly, for comparing the forecast performance of two models 1 and 2 directly, we implement the one-step ahead test for conditional predictive ability (CPA) from Giacomini and White (2006), Theorem 1, that assumes under the null hypothesis that forecasts of model 1 and model 2 have on average equal predictive ability conditional on previous information. As suggested by Giacomini and Komunjer (2005), we use the quantile loss function L_α for the test. This tests assumes under the null hypothesis that $H_0 : \ E\left[\Delta L_t | \mathcal{F}_{t-1}\right] = E\left[L_\alpha(f_\alpha^{(1)}, r_t) - L_\alpha(f_\alpha^{(2)}, r_t)|\mathcal{F}_{t-1}\right] \equiv E\left[h_{t-1}\Delta L_t\right] = 0$ and that this loss difference is a martingale difference sequence, where $\mathcal{F}_{t-1}$ contains all information up to time $t-1$ and $f_\alpha^{(1)}$ and $f_\alpha^{(2)}$ are two competing forecasts. The test statistic is computed using a Wald-type test with a set of factors h_{t-1} that can possibly predict the loss difference ΔL_t and is χ_q^2-distributed under H_0. More specifically, we choose $h_{t-1} = (1, \Delta L_{t-1})$ (i.e. $q = 2$), i.e. using the lagged loss difference and an intercept as predictors in a linear regression with parameter β_0 for the simulation and application.

2.4 Simulation

As both the GRF and the QRF have so far have been mostly studied for cross-section data, we provide simulation results on their out-of-sample performance for VaR^α-forecasts

[7] We use the implementation from the `GAS`-package in `R` (Ardia et al., 2019) including 4 lagged Hit-values, a constant, the VaR-forecast, and the squared lagged (log-)return.

in a financial time series set-up. Overall, we find that as expected, GRF performs best in all settings, in particular in settings which are designed to mimic the cryptocurrency behavior over time but also in those similar to stock index behavior. For QRF, the performance is entirely different, at least for the chosen parsimonious model specification and the relatively short estimation intervals.

We study two different types of DGPs. The first one is a standard GARCH(1,1) process, i.e

$$r_t = z_t \sigma_t \tag{2.3}$$

$$\sigma_t^2 = \omega + \beta_0 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \ , \tag{2.4}$$

where the parameters are estimated on the full Bitcoin data to mimic the behavior of cryptocurrencies (with $z_t \sim N(0,1)$). We denote this setting as *sim GARCH Bitcoin fit*. Moreover, we also consider the specification with $\beta_0 = 0.1$, $\beta_1 = 0.8$, $\omega = 10^{-4}$ and $z_t \sim N(0,1)$ (*sim GARCH*) and with z_t as t_5-distributed (*sim GARCH t*), which corresponds to standard stock index data. Secondly, for the *Sim SAV-Model* setting, we fit a symmetric absolute value (SAV) model to normal returns, i.e.

$$VaR_{t+1} = \gamma_0 + \gamma_1 VaR_t + \gamma_2 |r_t^{(init)} - \gamma_3| \ , \tag{2.5}$$

with $r_t^{(init)} \sim N(0, \sigma_t^2)$ and $\dfrac{\sigma_t}{65} \sim \chi_2^2$, where new draws of σ_t are only taken every 100 observations, keeping σ_t constant meanwhile. We then generate the final return as $r_t \sim N\left(0, \dfrac{\widehat{VaR}_t}{\Phi(\alpha)^{-1}}\right)$ from the fitted SAV-model, where $\Phi(\alpha)^{-1}$ is the quantile function of a standard normal variable. We do this to obtain returns that have exactly the *VaR* that we obtained from the SAV model before.

For all settings, we generate 2000 return observations and forecast the one-step ahead VaR over the different rolling window lengths $l = 500, 1000$. We repeat this generation process 200 times for $\alpha = 0.01$ and $\alpha = 0.05$. For comparison of the different methods described in Section 2.3, we use the DQ-test, the Kupiec test, the Christoffersen-test, and the AoE. Note that for all tests, we present aggregate results from two-sided t-tests of the empirical versus the nominal coverage. The results are therefore rejection rates of t-tests against the nominal level of 5%. Therefore, a lower rejection rate and higher

Table 2.4: Simulation: 5% VaR

Rolling Window	$l = 500$				$l = 1000$			
	DQ	Kupiec	Christoffersen	AoE	DQ	Kupiec	Christoffersen	AoE
Sim GARCH Normal								
QRF	0.940 (0.008)	0.325 (0.200)	0.200 (0.275)	1.185	0.725 (0.063)	0.160 (0.377)	0.090 (0.393)	1.148
GRF	0.495 (0.163)	0.000 (0.583)	0.015 (0.549)	1.040	0.225 (0.317)	0.020 (0.547)	0.030 (0.525)	1.030
QR	0.760 (0.048)	0.085 (0.441)	0.090 (0.438)	1.095	0.305 (0.270)	0.040 (0.537)	0.040 (0.496)	1.042
Hist	0.835 (0.045)	0.015 (0.554)	0.195 (0.365)	1.047	0.630 (0.110)	0.060 (0.490)	0.195 (0.368)	1.030
NormFit	0.740 (0.071)	0.040 (0.551)	0.205 (0.329)	1.006	0.555 (0.130)	0.065 (0.475)	0.195 (0.345)	1.001
CAViaR	0.785 (0.051)	0.025 (0.514)	0.010 (0.545)	1.073	0.275 (0.286)	0.055 (0.513)	0.025 (0.511)	1.032
GARCH(1,1)	0.445 (0.209)	0.030 (0.511)	0.155 (0.394)	1.046	0.205 (0.327)	0.055 (0.531)	0.095 (0.428)	1.028
Sim GARCH t								
QRF	0.920 (0.018)	0.315 (0.203)	0.210 (0.277)	1.188	0.655 (0.085)	0.175 (0.348)	0.100 (0.399)	1.162
GRF	0.435 (0.188)	0.010 (0.579)	0.025 (0.537)	1.029	0.260 (0.341)	0.030 (0.522)	0.020 (0.518)	1.022
QR	0.770 (0.060)	0.130 (0.407)	0.105 (0.407)	1.109	0.275 (0.303)	0.040 (0.507)	0.035 (0.468)	1.049
Hist	0.725 (0.077)	0.020 (0.573)	0.200 (0.360)	1.036	0.580 (0.121)	0.115 (0.440)	0.185 (0.334)	1.016
NormFit	0.695 (0.090)	0.310 (0.271)	0.370 (0.220)	0.839	0.550 (0.155)	0.340 (0.275)	0.370 (0.239)	0.817
CAViaR	0.525 (0.135)	0.010 (0.523)	0.010 (0.542)	1.059	0.175 (0.331)	0.055 (0.499)	0.040 (0.502)	1.032
GARCH(1,1)	0.405 (0.232)	0.135 (0.381)	0.210 (0.316)	0.893	0.270 (0.308)	0.200 (0.368)	0.220 (0.344)	0.862
Sim SAV-Model								
QRF	0.960 (0.010)	0.315 (0.188)	0.180 (0.265)	1.187	0.735 (0.071)	0.150 (0.352)	0.115 (0.413)	1.158
GRF	0.495 (0.148)	0.005 (0.571)	0.025 (0.570)	1.047	0.210 (0.317)	0.040 (0.518)	0.035 (0.548)	1.048
QR	0.895 (0.024)	0.050 (0.435)	0.045 (0.452)	1.102	0.310 (0.234)	0.045 (0.490)	0.045 (0.530)	1.064
Hist	0.290 (0.220)	0.035 (0.595)	0.070 (0.523)	1.047	0.165 (0.353)	0.070 (0.525)	0.080 (0.496)	1.048
NormFit	0.215 (0.360)	0.040 (0.539)	0.110 (0.502)	0.994	0.145 (0.418)	0.080 (0.501)	0.070 (0.474)	0.999
CAViaR	0.775 (0.053)	0.030 (0.525)	0.030 (0.563)	1.067	0.235 (0.289)	0.045 (0.527)	0.045 (0.563)	1.038
GARCH(1,1)	0.135 (0.338)	0.030 (0.567)	0.065 (0.528)	1.021	0.065 (0.469)	0.045 (0.546)	0.040 (0.524)	1.018
Sim GARCH Bitcoin fit								
QRF	0.875 (0.018)	0.325 (0.200)	0.200 (0.275)	1.185	0.585 (0.110)	0.160 (0.377)	0.090 (0.393)	1.148
GRF	0.255 (0.303)	0.000 (0.583)	0.015 (0.549)	1.040	0.160 (0.454)	0.020 (0.547)	0.030 (0.525)	1.030
QR	0.635 (0.095)	0.085 (0.441)	0.090 (0.438)	1.095	0.170 (0.380)	0.040 (0.537)	0.040 (0.496)	1.042
Hist	0.600 (0.102)	0.015 (0.554)	0.195 (0.365)	1.047	0.385 (0.269)	0.060 (0.490)	0.195 (0.368)	1.030
NormFit	0.540 (0.159)	0.040 (0.551)	0.205 (0.329)	1.006	0.330 (0.299)	0.065 (0.475)	0.195 (0.345)	1.001
CAViaR	0.970 (0.007)	0.375 (0.178)	0.280 (0.224)	0.814	0.655 (0.055)	0.305 (0.236)	0.270 (0.273)	0.791
GARCH(1,1)	0.230 (0.319)	0.030 (0.536)	0.180 (0.399)	1.054	0.100 (0.458)	0.040 (0.545)	0.080 (0.438)	1.032

Notes: The table displays rejection rates of t-tests of empirical quantile levels against the nominal level of 5% for DQ-, Kupiec- and Christoffersen-tests and mean p-values in parentheses. Thus higher p-values and lower rejection rates indicate better model performance. The GARCH case uses predictions from the QMLE-fit of a GARCH(1,1) specification with normally distributed errors and can therefore be seen as an oracle for the GARCH-simulated specifications.

mean p-values (in parentheses) indicate better performance. For GRF, QRF, and QR, we use a common set of lagged covariates as described at the end of Section 2.3.

Table 2.4 summarizes the results of the simulation for the 5% VaR. According to the more advanced *DQ*- and Christoffersen-tests for evaluation, GRF consistently outperforms the other methods in almost all cases, indicating superior predictive quality. This is very much in contrast to the QRF which is consistently dominated by the other models. For the *Sim GARCH* and *Sim GARCH t* cases which mimic standard stock indices, the performances of GRF, CAV and QR appear in a similar range with mostly advantages for GRF in particular for smaller sample sizes. This holds generally for normally distributed as well as heavy tailed innovations which lead to similar results also in magnitude of the rejection rates. As expected, forecasting performance increases throughout all models with larger estimation windows, though with CAV often profiting the most from the larger sample sizes. For these settings with GARCH as true DGP, the performance of the GARCH model serves as an oracle reference. In the t-innovation case, it has coverage problems and GRF is even able to outperform it in Christoffersen-tests.

For the *Sim SAV Model* and the *Sim GARCH Bitcoin fit* the situation, however, differs substantially. In these cryptocurrency-like cases, the GRF clearly dominates the QR and CAViAR particularly strongly in the small 500 observations setting. Considering our application where only a relatively small time span is available, this seems crucial. Moreover, CAViaR runs into coverage problems according to the AoE results which even deteriorate for larger sample sizes for *Sim GARCH Bitcoin fit*. In the latter case, the GRF rejection rates are close to the GARCH benchmark while the strong conditional dependence structure in the tails of the *Sim SAV Model* setting shows that for such extreme cases, the unconditional coverage of GRF is still excellent, but the conditional coverage measured by the DQ-test is only average among all models for the larger estimation samples and even below for the smaller ones. These relative findings generally prevail for the 1% VaR forecasts, but absolute performance is generally worse for all methods, especially with a smaller rolling window of 500 (see Table 2.17 in the appendix.). Intuitively, this finding is reasonable, since relevant observations for the 1% level should in theory only occur in 5 of the 500 observations, making it harder for the data-driven methods to predict such unlikely events.

Additionally, we conduct direct pairwise comparison tests between the superior random forest type method GRF against the best performing non-oracle other parametric methods via CPA-tests for each scenario. The respective results are reported in Table 2.5. In the majority of cases, GRF outperforms its competitors on average, however, mean p-values are mostly not significant. For example, GRF has a smaller loss (i.e. quantile loss) than QR in about 90% of the forecasts (aggregated over all runs) but a mean p-value of 0.298, which would not qualify as a significant out-performance (on average). This of course does not mean that the test never rejects but is likely be caused by high variances in the p-values over different simulation runs. Furthermore, the competing methods are also slightly improving with window length, which further implies that GRF can deal better with a smaller training time frame.

Table 2.5: CPA-tests on Predictions of 5% VaR for Different Window Lengths and Different Models

Rolling Window	$l = 500$			$l = 1000$		
GRF vs:	QR	Hist	CAV	QR	Hist	CAV
Sim GARCH Normal						
Mean P-Value	0.429	0.351	0.334	0.491	0.388	0.463
No. P-Values < 0.1	34	55	50	24	44	27
GRF-Performance	0.780	0.755	0.847	0.501	0.744	0.659
Sim GARCH t						
Mean P-Value	0.392	0.382	0.396	0.431	0.397	0.446
No. P-Values < 0.1	43	40	34	30	37	15
GRF-Performance	0.732	0.707	0.692	0.439	0.717	0.529
Sim SAV-Model						
Mean P-Value	0.298	0.311	0.374	0.471	0.381	0.448
No. P-Values < 0.1	53	63	42	26	50	22
GRF-Performance	0.903	0.279	0.775	0.657	0.321	0.658
Sim GARCH Bitcoin fit						
Mean P-Value	0.429	0.351	0.001	0.491	0.388	0.003
No. P-Values < 0.1	34	55	200	24	44	199
GRF-Performance	0.780	0.755	0.995	0.501	0.745	0.994

Notes: The table displays CPA-tests for 5% one-day ahead VaR-forecasts of the best performing random forest type techniques GRF in Table 2.4 versus the best parametric time series models. We report mean p-values, the number of significant p-values over 200 iterations and the rate at which GRF outperforms the competing method (i.e. a value of 0.8 means that GRF has a smaller error loss than the competing method in 80% of the rolling window forecasts over all runs). Low p-values paired with performance rates larger than 0.5 indicate that GRF outperforms the competing methods.

2.5 Results

In this section, we highlight the advantages from using non-linear machine learning-based methods for forecasting the VaR of cryptocurrencies. In particular, we show for a large cross-section of more than 100 cryptocurrencies that the proposed random forest method GRF yields superior performance across a wide range of different types of cryptocurrencies and different time periods. Investigating the underlying drivers, we illustrate that the non-linear model predictions excel especially for assets that are frequently traded by a large amount of different users, and for more volatile assets and times.

More specifically, we predict the $5\% - VaR$ as a key quantity in risk management for our comprehensive set of cryptocurrencies. In an extensive out-of-sample forecasting study, we compare the random forest-based machine learning methods to standard linear time series and GARCH-type models including approaches with exogenous asset information in covariates. The prediction performance is assessed with the DQ-test to obtain an overall aggregate picture on the realized coverages as well as pairwise CPA tests across different time periods and types of cryptocurrencies. Based on these findings, we focus our analysis on three important selected currencies comprising Bitcoin (btc) as the largest currency by far regarding market cap, Tether (usdt) as a stablecoin with lower volatility, and Cardano (ada) as a currency specifically allowing for smart contracts. For these we also consider predicted loss series by CPA-tests and variable importance measures to uncover important drivers. We furthermore identify single events that majorly affect the predictive ability of the procedures.

2.5.1 Aggregated Forecasting Performance

In this section, we provide results on aggregate forecast performance of the different modeling approaches over all cryptocurrencies.

Direct Pairwise Forecast Comparisons

As a comparison to GRF we use the wide range of standard financial time series methods as introduced in Section 2.3. For assessing the prediction performance, we conduct pairwise CPA-tests separately for the three different specified time periods over all cryptocurrencies. The results of the tests are contained in Table 2.6 and Figure 2.2. Note that the CPA-tests require the rolling window length to be smaller than the out-of-sample forecast window to produce valid results, which is why the results from the shortest period Period 3, where both sizes are equal to 500 might have less power. Specifically for more recently introduced cryptocurrencies, the out-of-sample size is too low for the CPA-tests to have high power. Therefore, we additionally look at the direct comparisons of predicted losses (as suggested by Giacomini and White (2006), Section 4)[8], where

[8] We use the lagged loss difference and an intercept for loss prediction in an auto-regressive setup since these are the main drivers of the test statistic in the CPA test.

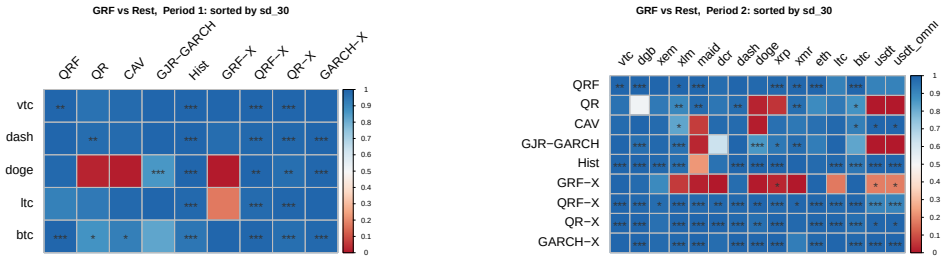
we compare in the loss series how often GRF is better, i.e. has a smaller loss, than its competitors (*GRF-Performance*). Note that a value of one thus indicates that GRF has a smaller predicted loss over the full loss series.

In general, GRF performs better than its competitors for a majority of crypotocurrencies over all time periods. Table 2.6 summarizes the results. We can see that QRF is almost always outperformed, and for around 50% of cryptocurrencies, losses are even significantly smaller. This is not surprising, as QRF has a similar structure to GRF while not being tuned to predict the quantiles directly. Thus, we expect it to be less sensitive to changes that only affect the quantile of the return distribution, for example large shock events. The same holds for the GARCH-X and Hist, which are clearly outperformed by the GRF, as well as the QR-X and QRF-X. Adding exogenous information in covariates as part of the non-linear GRF (i.e. GRF-X) is better especially in later periods (see Figure 2.7 in the appendix). This is interesting, since the other methods cannot benefit as much as GRF from additional covariates. Generally, for cryptocurrencies, the non-parametric form of the GRF helps to extract information from exogenous covariates in contrast to standard parametric methods such as QR and GARCH. As GRF accounts specifically for the quantiles in the random forest splitting function, this helps to also favorably integrate additional covariates X in contrast to QRF. Overall, however, both GRF-procedures seem to perform very similarly, especially in pairwise comparisons. For full results of the GRF-X, see Table 2.14 and Figure 2.7 in the appendix. While CAV, QR, and GJR-GARCH are outperformed over the majority of cryptocurrencies, only 20% to 50% of these out-performances reach significance. In subsection 2.5.2, we will focus on specific cryptocurrencies for a more in depth understanding.

When considering the single time periods, it is notable how for Period 1 and 2 (bottom part of Table 2.6), GRF(-X) is constantly outperforming the other methods for most cryptocurrencies, and only has somewhat worse performance for doge and the stablecoins, although these are insignificant (see below). For the first two periods, QR is maybe the most competitive of the other methods, while there is a general tendency for the classic methods to perform worse with higher volatility of returns, which can be seen in Table 2.6 for Period 2 where the currencies are ordered from highest 30 day lagged SD on the left to the lowest on the right.

Table 2.6: Performance and Significance of CPA-tests Over Different Time Periods for GRF Without Additional Covariates

GRF vs.:	QRF	QR	CAV	GJR-GARCH	Hist	GRF-X	QRF-X	QR-X	GARCH-X
Share of GRF With Better Performance									
Period 1	1.00	0.80	0.80	1.00	1.00	0.60	1.00	1.00	1.00
Period 2	1.00	0.73	0.87	0.80	0.93	0.40	1.00	1.00	1.00
Period 3	0.92	0.68	0.71	0.64	0.90	0.44	0.92	0.96	0.70
Full Data	0.91	0.74	0.75	0.71	0.82	0.44	0.90	0.97	0.72
Share of GRF With Significantly Better Performance									
Period 1	0.40	0.40	0.20	0.20	1.00	0.00	1.00	1.00	0.60
Period 2	0.53	0.33	0.27	0.33	0.73	0.00	1.00	0.87	0.67
Period 3	0.43	0.21	0.23	0.19	0.51	0.09	0.74	0.78	0.53
Full Data	0.41	0.31	0.24	0.18	0.41	0.12	0.69	0.83	0.54



Notes: The top part shows summary values that are shares over all cryptocurrencies in the respective time period. It describes the number of times that GRF had a better performance (i.e. more than 50% of predicted losses by the CPA test were smaller for the GRF) relative to all crytpocurrencies (in that period), and the number of times that GRF was significantly better (at least at a 10% level) as judged by the CPA test over all crytpocurrencies (in that period). The bottom part shows the detailed results of CPA-tests with the color of each box indicating the performance of GRF. Blue signifies a performance of 1, meaning that GRF has a smaller predicted loss in 100% of cases. *, **, *** shows significance on a level of 10%, 5%, and 1%. The values are ordered by 30 day lagged standard deviation from highest to lowest (top/left to bottom/right).

This becomes more apparent in Period 3, where we deal with much more cryptocurrencies (77) and a much shorter time horizon (500 out-of-sample observations). Here, methods such as GJR-GARCH and QR are on par with GRF or even better when looking at low-volatility (partly regulated) stablecoins such as pax, gusd, tusd, dai, and usdt with its derivatives (e.g. usdt_eth,usdt_trx). CAV, on the other hand, is only rarely better here (as indicated by CPA tests), while being significantly outperformed for

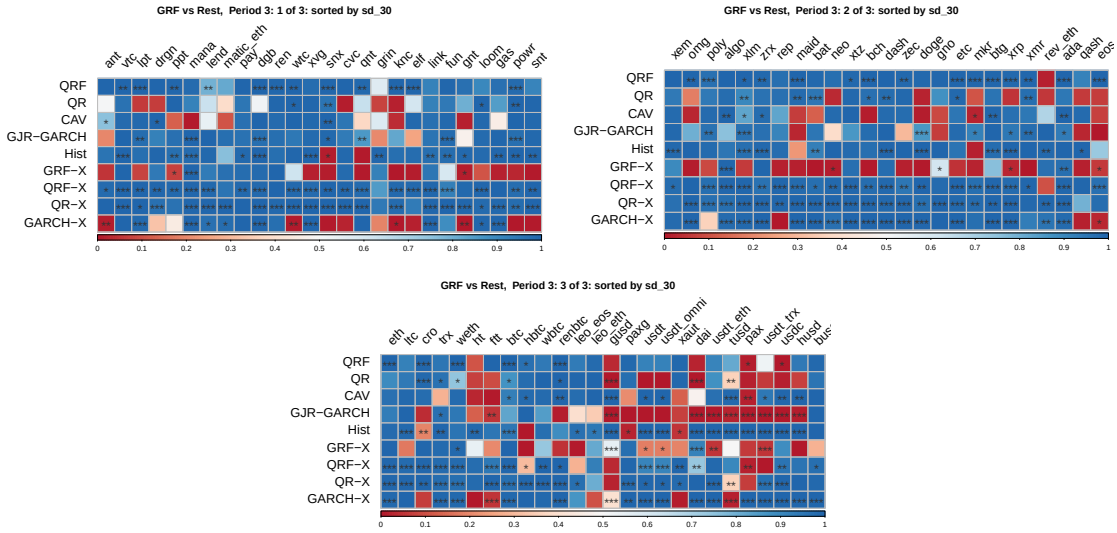


Figure 2.2: Overview of Results for CPA-Tests of GRF vs. All Other Methods for Cryptocurrencies in the Third Period

Notes: Cryptocurrencies are ordered by 30 day lagged standard deviation from highest to lowest, left to right. The color of each box indicates the performance of GRF, with 1 (blue) indicating that GRF has a smaller predicted loss in 100% of cases. *, **, *** indicate significance on a level of 10%, 5%, and 1%.

important and large assets such as btc, ada, xlm, and most stablecoins where the QR and GJR-GARCH performed well.

To highlight the specific properties of currencies where GRF outperforms the other methods, we split the assets into two groups. The first group (Group_low) contains assets where GRF performance is low in comparison to the three other methods that were able to compete in some cases with GRF, namely QR, CAV, and GJR-GARCH. We add an asset into that group when at least two of the methods outperform GRF (in terms of loss difference) for that asset, separately for each time period. All other assets are sorted into the second group (Group_high), indicating high performance of GRF. Table 2.8 summarizes the results over these groups for each time period and covariate. In each group, we take the mean over all cryptocurrencies of median values for each covariate. We then divide Group_low by Group_high. For example, An SER of 0.15 in Period 3 indicates that cryptocurrencies in Group_low have, on average, a median

SER that is only 15% to that of Group_high, or in other words, the median SER for Group_high is around $6.7 = \frac{1}{0.15}$ times higher than that of Group_low on average.

We see that covariates of cryptocurrencies for which GRF performs better have much higher volatility (especially for the second and third period), a much higher SER[9], indicating a larger concentration of supply at a lot of small addresses, a higher market capitalization, a lower rate of turnover (Velocity), and more active and total users. To summarize, this confirms the observation that GRF performs better for assets with highly varying returns that are traded by a large amount of users, which could thus also be prone to speculation. On the other side, methods such as QR or GJR-GARCH are better with more stable currencies that are used more as a hedging device (e.g. stablecoins).

Table 2.8: Difference Between Covariates of Cryptos Where GRF is Better vs. Worse

	Period 1	Period 2	Period 3	Full Data
Ret	1.20	0.16	1.05	0.92
Active_Users	0.34	0.15	0.26	0.34
Total_Users	0.79	0.18	0.13	0.15
Total_Users_USD100	0.22	0.17	0.27	0.35
Total_Users_USD10	0.41	0.17	0.19	0.25
CapMrktCurUSD	0.08	0.11	0.37	0.33
SER	1.07	0.50	0.15	0.21
Transactions	0.42	0.04	1.20	1.53
VelCur1yr	4.68	3.11	1.48	1.43
sd_3	0.89	0.52	0.59	0.58
sd_7	0.91	0.54	0.59	0.57
sd_30	0.90	0.52	0.59	0.58
sd_60	0.91	0.54	0.58	0.57

Notes: Values are shares of groups of cryptocurrencies where at least two of CAV, QR, and GJR-GARCH have better CPA-performance than GRF divided by the remaining rest. Raw values before division are mean values over all cryptocurrencies for the median of each covariate in the respective time period.

Backtesting

Apart from directly comparing the methods against each other, we also check their ability of predicting VaR in general using the DQ-test (see Section 2.3 for details), which is

[9] Apart from the first period where the only currency belonging to Group_low is doge.

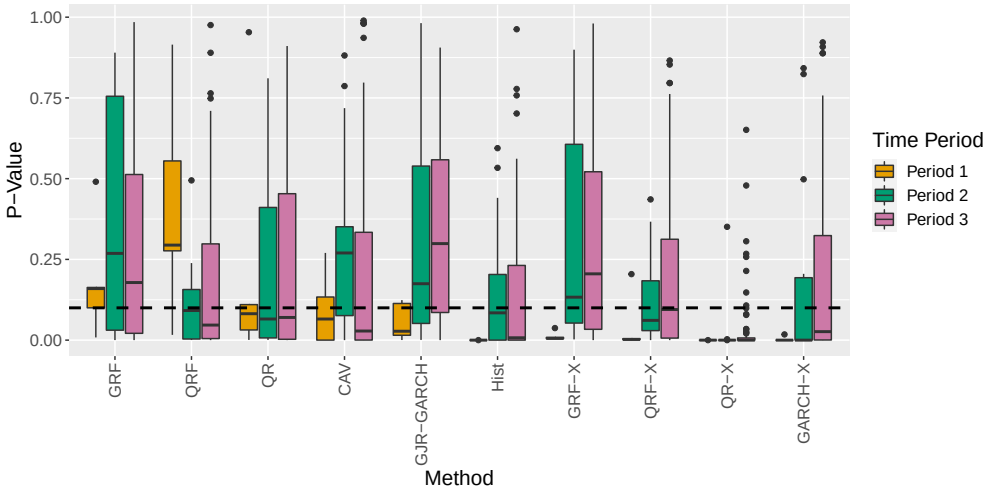


Figure 2.3: Boxplots of P-Values of DQ-Tests Over All Cryptocurrencies

Notes: P-values are separated over each time period and each method. The dashed horizontal line depicts a level of 0.1.

representative for all backtests[10]. The results over the different time periods for the 5% VaR-predictions are shown in Figure 2.3. We can see that depending on the time period, the p-values vary strongly, which is not surprising giving the different characteristics of each period and the increasing number of cryptocurrencies in the later periods. In general, GRF is the only method with median values consistently over the 10% level, indicating that is the most consistently calibrated forecasting method. The QRF performs extremely well in the first time period, but rejects the test often in the third period. Adding external covariates does not help in general lowers p-values substantially especially for the QR. Only in the final third period, adding external covariates increases the p-values slightly for the GRF and QRF (GRF-X, QRF-X). This indicates that those extra covariates are not necessarily predictive for extreme returns, or rather that the existing measures such as lagged returns and SD comprise the information already quite well. Compared to its non-forest counterparts, only CAV and GJR-GARCH can partly keep up with GRF. GJR-GARCH has slightly higher p-values than GRF in the third period, which is the shortest and which contains the most currencies. It is also marked by less extreme returns

[10]Detailed results for the other tests are omitted here for reasons of clarity, do not differ substantially, and are available upon request from the authors.

and a large reduction in active users, which could indicate that the forest methods excel in particular in highly volatile periods, when large shifts in the market are present. On the other hand, CAV has a similar, slightly worse performance in the second period while performing much worse than GRF in the other periods. This highlights the inability of the parametric methods to adapt to rapidly changing situations such as in Period 1 and 3. QR, Hist, and GARCH-X are all fully dominated by GRF throughout the three time periods as expected, as QR can only incorporate changes linearly, and GARCH-X and Hist serve as simple baselines.

Investigating the single time periods more in detail, we can see that for all methods except the QRF, the later time periods are easier to forecast, while only the forest-based methods seem to benefit from additional covariates in the third period. It is also notable that the p-values are much wider spread for most methods in the later two periods, which can be explained by a more heterogeneous structure in cryptocurrencies in Period 2 and 3. For an overview of DQ-tests over the full period, where achieving good coverage results is harder in general for all the methods due to the changing dynamics in cryptocurrencies, see Tables 2.11 and 2.9 in the appendix.

2.5.2 Extension: In-Depth Analysis of Specific Classes of Assets

To identify which specific events drive the performance of these methods, we analyze the predicted loss series of CPA tests over the full horizon of availability for the three cryptocurrencies Bitcoin, Cardano, and Tether separately. We furthermore show which covariates are important over the full data and specific time periods using variable importance measures of GRF-X.

We choose Bitcoin since it is the largest currency by market cap, with the longest data availability, Tether as the largest stablecoin by daily volume and market cap, and Cardano as a fairly new (i.e. fewer observations), however large currency (again by market cap), which can be used for smart contracts, identity verification, or supply chain tracking[11]. Since we deal with VaR-predictions, the initial loss function is the quantile loss with a quantile $\alpha = 0.05$.

[11] See e.g. `https://cardano.org/enterprise/`, accessed 19/05/2022.

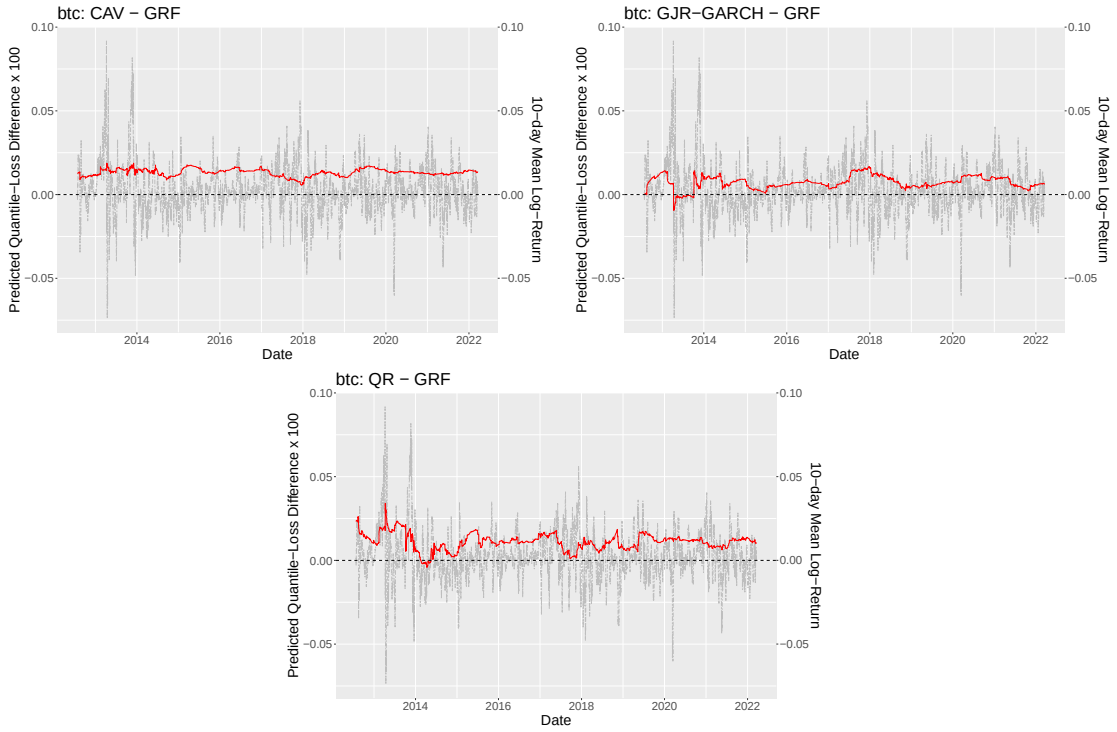


Figure 2.4: Rolling 180-Day Mean of Predicted Loss Difference Series for Bitcoin

Notes: Predicted loss difference $h_t\hat{\beta}_0$ (red) of CPA tests on Bitcoin (btc) predicted 5% VaR with $l = 500$ for GRF vs. CAV (left), GJR-GARCH (right), and QR (bottom center) with rolling mean 10-day log-returns in gray. A positive predicted loss difference indicates that the prediction error of GRF is smaller than of the compared method.

First, we look at Bitcoin (btc), the largest and most popular currency, where GRF largely outperforms QR, CAV, and GJR-GARCH in the CPA-tests. Figure 2.4 shows the predicted loss difference for each of the different methods. GRF outperforms the other methods consistently for most time frames. This is most likely due to the specific tailoring of the methodology to quantiles, as it outperforms QRF (not plotted) consistently here. For the parametric methods GJR-GARCH and QR, there are two short time periods where they have a smaller loss. For GJR-GARCH, this happens in the very beginning of the out-of-sample periods in April 2013, where btc crashed with negative log-returns of up to -0.66. Since this was the first drop of that magnitude for btc, GRF, as a forest based-method, had never seen such an extreme event, therefore could have technically

not predicted it. GJR-GARCH, on the other hand, as a parametric method, has no range restriction in that regard. In the following crashes, GRF correctly predicts these extreme events better than GJR-GARCH, which is visible from the loss series. For QR, the short time frame is only caused by the predictions of the CPA test itself, while actual losses are smaller for GRF[12].

Secondly, as summarized on the left in Figure 2.5, we look at Cardano (ada), a large and fairly new currency offering e.g. smart contracts or supply chain tracking. There, GRF is significantly outperforming GJR-GARCH and CAV, while being slightly better than QR, although not reaching a significant level. This lack of power for QR is likely due to the small out-of-sample size (roughly 1000) for the fairly new asset compared to the training window of $l = 500$. Again, we can see that for the most extreme event in March 2020, GJR-GARCH is slightly better, as GRF has not yet seen such an extreme event, therefore is not able to correctly predict the size of the loss. One normal solution would be to increase the training length, which is not possible in this case with a fairly new currency. Interestingly, in the later extreme events in May 2022, GRF increases in performance compared to GJR-GARCH, confirming the challenge with lacking training data.

Finally, we also look more closely at Tether (usdt) as the largest stablecoin that is roughly bound to the USD[13]. The right part of Figure 2.5 shows the 30-day rolling means of the predicted loss differences. Notably, the rolling loss difference and rolling mean-return are already around 10 times smaller than those of btc or ada, indicating that usdt substantially differs from the other two currencies. While the losses of QR are very similar to those of GRF, CAV is significantly worse. Only GJR-GARCH consistently has lower predicted losses than GRF, although they are deemed not significant by the CPA tests (see Figure 2.10 in the appendix for detailed results for the full time frame). We can see, however, that for the relatively rare tail events in 2017, which are still not too extreme, there is some variation in the predicted loss difference. For the first event that is quite extreme, GJR-GARCH reacts too late, and only the second event is correctly detected by GJR-GARCH, as there is some limited information of anticipation.

[12] It might indicate, however, that the methods perform quite equally during that specific time, since there is no predictable difference.

[13] As it is backed by USD cash reserves, see `https://tether.to/en/`.

In general, GRF tends too overshoot less in these situations, although both methods are somewhat badly calibrated according to DQ-tests (see Table 2.11 in the appendix for details).

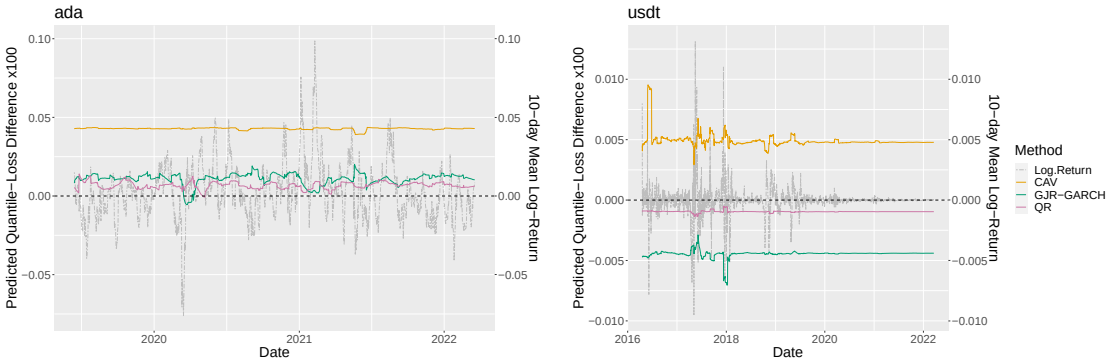


Figure 2.5: Rolling 30-Day Mean of Predicted Loss Difference Series for Cardano and Tether

Notes: Predicted loss difference $h_t\hat{\beta}_0$ of CPA tests on Cardano/Tether (ada/usdt) predicted 5% VaR with $l = 500$ for GRF vs. CAV (orange), GJR-GARCH (green), and QR (purple) with rolling mean 10-day log-returns in gray. A positive predicted loss difference indicates that the prediction error of GRF is smaller than of the compared method.

To further understand the drivers of the GRF performance, we obtain variable importance measures that depict the frequency of inclusion in splits of the forest[14]. In Figure 2.6, the importance difference of certain covariates over time for the three currencies is clearly visible. Overall, the lagged return is very important for predicting VaR when returns are quite extreme relative to all returns in a specific asset, in the case of btc in times of hypes and crashes. Intuitively, this finding seems reasonable as in times of bubbles, when the volatility is driven by some short, bubble-like events and returns are highly variable, volatility lagged over a longer time horizon is less predictive for VaR and predictions are driven by events happening shortly before the prediction. In rather unstable times, but not in extreme crashes, the lagged SD-measures gain importance,

[14] We use a maximum depth of $d_{max} = 5$ corresponding to the number of covariates and a weight decay of 2, meaning a split further down in each tree receives less weight w_l in the final frequency as it is less important for the three specific currencies analyzed in the previous section. Specifically, for layer $l = 1, \ldots, 5$, $w_l = \dfrac{l^{-2}}{\sum_{l=1}^{5} l^{-2}}$.

while the extra covariates only play a role for assets with relatively small volumes, e.g. when new currencies are created. This also explains why GRF-X performs much better for new, low-market-cap assets in Period 3 (see e.g. Figure 2.9 in the appendix).

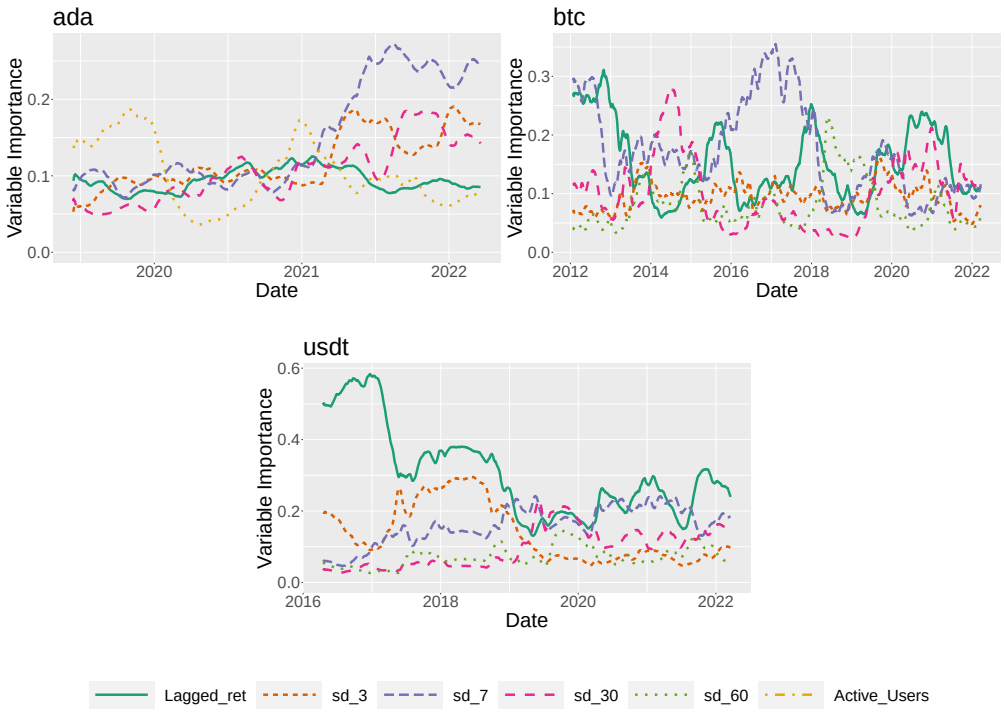


Figure 2.6: Rolling 30-Day Mean of the GRF Variable Importance

Notes: Values are for the out-of-sample period on the full data of ada, btc, and usdt, for predicted one-day ahead 5% VaR with $l = 500$. The 5 most important variables for each cryptocurrency are plotted. Variable importance of covariate x_p is measured as the proportion of splits on x_p relative to all splits in a respective layer l (over all trees in a trained forest), weighted by layer l. Variable description can be seen in Section 2.2.

Starting with ada, we see that it is the only asset of the three where the number of active addresses play an important role, where for the other two assets, the 60-day lagged standard deviation is more important. The importance of variables can be split into two periods. The period until the beginning of 2021 is largely dominated by measures that somehow account for trading activity (Active_Users,Total_Users/_USD100/_USD10, Transactions). For reasons of clarity, we only plot the most important one of these

measures, Active_Users. The spike of the latter in importance at the beginning of 2021 is likely caused by the massive increase in price and market cap during that time, representing a period of hype with many actively trading users[15]. For the rest of the time period, lagged SD, mostly 3-day lagged SD, followed by 30-day and 60-day SD, is dominating the predictions of GRF. This change of importance seems reasonable as the structure of the asset fundamentally changes with the price increasing tenfold and the volume increasing strongly at the same time.

For btc, we have much more data covering 10 years, which is why the important variables change frequently in different periods. Lagged return is naturally important in phases of extreme hype and crashes that are characterized by large positive and negative returns, e.g. in the very beginning (where the price was still quite low), at the end of 2013 (the first time btc had a price of USD 1000), at the end of 2017 (with a price over USD 19,000), and from mid 2020 to the mid 2021, where there were multiple hypes and crashes during the Covid-19 pandemic. Between these hype periods, the lagged SDs are most important. In 2014-2015 and from the end of 2019 to mid 2021 , 30-day SD is contributing most to the GRF-predictions, followed by 7-day SD in 2016-2018 and 60-day SD 20 from 2018 to the end of 2019. This changing scheme is interesting, as 30-day SD seems to be a good predictor especially in very unstable times (return-wise), while 7-day and 60-day SD are more important in relatively stable times.

Finally, usdt is an exceptions, being largely dominated by lagged return, which makes sense considering the performance of GJR-GARCH in that asset, where lagged-returns play an important role in terms of leverage. From mid 2019, the prices and returns are rather stable and the volume increases strongly, and the influence lagged 7-day SD increases slightly, while still being less important than lagged return. This is not surprising, as usdt is quite stable in comparison to btc and ada.

2.6 Conclusion

In this paper, we show that random forests can significantly improve the forecasting performance for VaR-predictions when tailored to the task of quantile regression. In

[15]See also Figure 2.11 in the Appendix for an overview of the log-returns of the three currencies.

both simulations and analyzing return data of 105 of the largest cryptocurrencies, the proposed random forest (i.e. GRF) proves to be the most reliable method. This can be attributed to the non-linear form of the return data with large time-variations of volatility that call for methods that can adapt to changes in a nonparametric way, while other classic methods break down. We further show that the GRF is better in assessing the tail risk of cryptocurrencies in times where speculation and therefore volatility in returns is high, e.g. when there is a speculative bubble. There, more simple procedures perform especially bad and the comparison of predicted losses could thus GRF could serve as an easy, empirical alternative to detect such bubbles.

Our findings are highly relevant for the risk assessment of cryptocurrencies, where high volatility changes and large returns are often found. Classic methods can therefore lead to false security and miss-assessment of risks (and chances) of these assets. We further identify periods and assets where GRF performs especially well, which is especially with volatile assets that have a high number of active users and could thus be prone to speculation and hypes. On the other hand, for the class of stablecoins that are usually bound to some large, classic currency such as the USD, and where markets are usually dominated by a smaller number of large accounts, other classic methods such as GJR-GARCH or quantile regression are on par with GRF.

The random forest methodology allows us to identify important factors which we show to be time-varying and that are changing particularly in unstable times. For future research, an interesting extension would therefore be to even augment this set of covariates with other potentially driving real-time factors, such as for example social media information. The relevance of such factors might also provide additional guidance for relevant exogenous information to be included in standard parametric models such as CAViaR.

2.7 Appendix

Table 2.9: Summary of P-Values of DQ-Tests Over All Cryptos

	GRF	QRF	QR	CAV	GJR-GARCH	Hist	GRF-X	QRF-X	QR-X	GARCH-X
0%	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.010	0.000	0.000	0.000	0.005	0.000	0.008	0.001	0.000	0.000
50%	0.081	0.006	0.001	0.011	0.086	0.003	0.063	0.032	0.000	0.007
75%	0.296	0.105	0.139	0.160	0.341	0.045	0.353	0.159	0.000	0.246
100%	0.939	0.784	0.937	0.939	0.987	0.884	0.980	0.866	0.528	0.794

Notes: Rows depict the quantiles of p-values of DQ-tests aggregated over all cryptos for the full available time frame and each method, respectively.

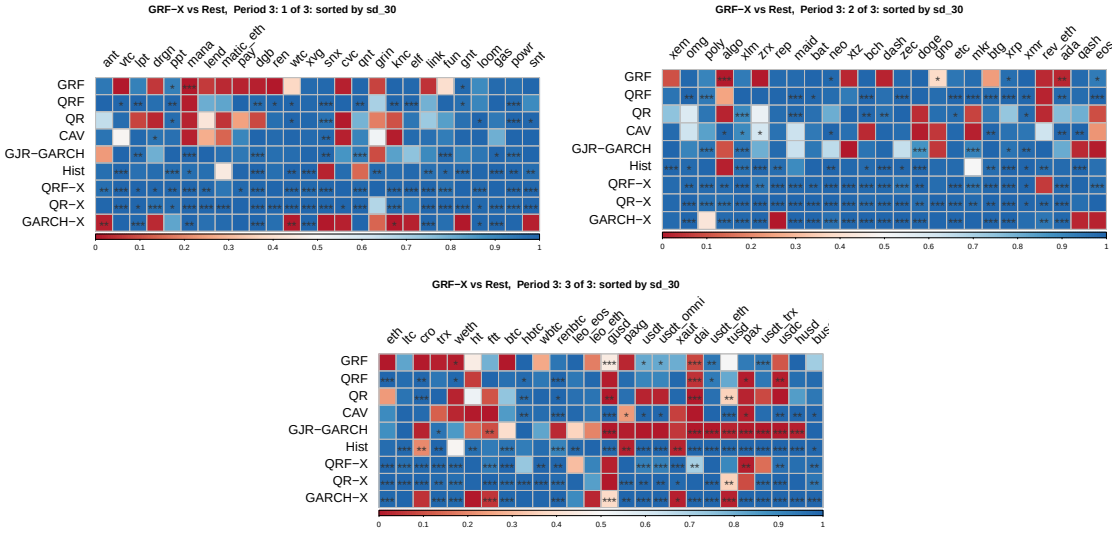


Figure 2.7: Overview of Results for CPA-Tests of GRF-X vs. All Other Methods in the Third Period

Notes: Results are ordered by 30 day lagged standard deviation from highest to lowest, left to right. The color of each box indicates the performance of GRF-X, with 1 indicating that GRF-X has a smaller predicted loss in 100% of cases. *, **, *** indicate significance on a level of 10%, 5%, and 1%.

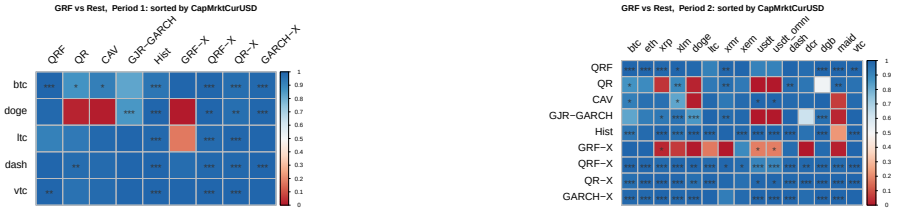


Figure 2.8: Overview of Results for CPA-Tests of GRF vs. All Other Methods in the First and Second Period Ordered by Market Cap

Notes: Results are ordered from highest to lowest. The color of each box indicates the performance of GRF, with 1 indicating that GRF has a smaller predicted loss in 100% of cases. *, **, *** indicate significance on a level of 10%, 5%, and 1%.

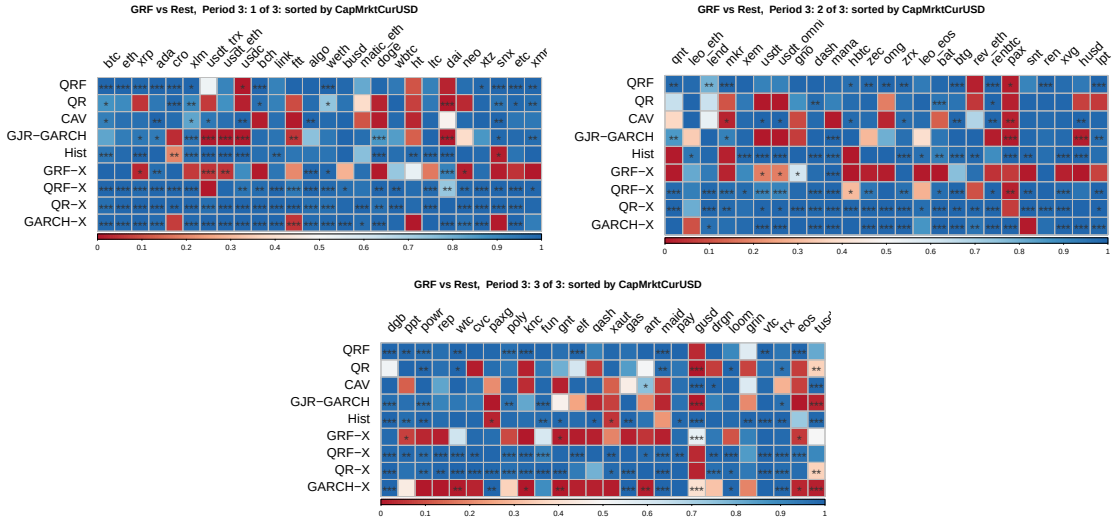


Figure 2.9: Overview of Results for CPA-Tests of GRF vs. All Other Methods in the Third Period Ordered by Market Cap

Notes: Results are ordered from highest to lowest, left to right. The color of each box indicates the performance of GRF, with 1 indicating that GRF has a smaller predicted loss in 100% of cases. *, **, *** indicate significance on a level of 10%, 5%, and 1%.

Table 2.10: Overview of Non-Stationarity Tests

	KPSS_level	KPSS_trend	ADF
algo	0.09	0.02	0.01
alpha	0.02	0.10	0.01
bnb	0.06	0.05	0.01
bnb_eth	0.06	0.05	0.01
btc	0.05	0.10	0.01
crv	0.10	0.03	0.01
dash	0.07	0.10	0.01
dcr	0.10	0.08	0.01
dot	0.10	0.10	0.01
eth	0.10	0.08	0.01
ftt	0.10	0.04	0.01
gno	0.10	0.10	0.01
gnt	0.10	0.04	0.01
icp	0.10	0.02	0.01
lend	0.10	0.02	0.01
loom	0.05	0.10	0.01
neo	0.10	0.08	0.01
omg	0.10	0.02	0.01
poly	0.04	0.10	0.01
ppt	0.10	0.07	0.01
snx	0.02	0.10	0.01
sushi	0.10	0.01	0.01
uni	0.10	0.08	0.01
weth	0.05	0.10	0.01
wtc	0.10	0.09	0.01
xem	0.07	0.10	0.01
xmr	0.10	0.04	0.01
xtz	0.10	0.09	0.01
yfi	0.08	0.10	0.01

Notes: Values in columns show p-values for KPSS-test and ADF-tests as described in Section 2.2. We only show values for assets that have values smaller than 0.1 for the KPSS-tests. No asset had p-values larger than 0.01 for the ADF-tests.

Table 2.11: P-Values of DQ-Tests for Employed Crypto Assets

	GRF	QRF	QR	CAV	GJR-GARCH	Hist	GRF-X	QRF-X	QR-X	GARCH-X
1inch	0.006	0.004	0.000	0.010	0.001	0.006	0.006	0.533	0.000	0.007
aave	0.131	0.008	0.000	0.042	0.020	0.010	0.209	0.106	0.000	0.103
ada	0.266	0.000	0.095	0.000	0.645	0.000	0.031	0.001	0.000	0.000
algo	0.820	0.764	0.455	0.000	0.545	0.300	0.673	0.181	0.000	0.001
alpha	0.276	0.095	0.000	0.458	0.000	0.115	0.102	0.239	0.000	0.088
ant	0.189	0.015	0.306	0.024	0.113	0.102	0.368	0.063	0.000	0.566
bal	0.000	0.000	0.000	0.000	0.090	0.000	0.000	0.001	0.000	0.113
bat	0.002	0.000	0.000	0.032	0.072	0.001	0.004	0.036	0.000	0.000
bch	0.366	0.001	0.001	0.104	0.002	0.013	0.944	0.157	0.000	0.000
bnb	0.205	0.000	0.000	0.024	0.488	0.216	0.245	0.022	0.000	0.002
bnb_eth	0.205	0.000	0.000	0.002	0.474	0.216	0.245	0.022	0.000	0.036
bsv	0.041	0.000	0.139	0.013	0.000	0.000	0.300	0.427	0.000	0.000
btc	0.028	0.000	0.000	0.000	0.039	0.000	0.002	0.000	0.000	0.000
btg	0.922	0.165	0.706	0.002	0.138	0.205	0.847	0.186	0.000	0.000
busd	0.939	0.005	0.589	0.011	0.987	0.562	0.931	0.362	0.000	0.001
comp	0.011	0.079	0.000	0.000	0.019	0.010	0.030	0.001	0.000	0.001
cro	0.000	0.000	0.000	0.000	0.278	0.001	0.000	0.000	0.000	0.004
crv	0.047	0.133	0.000	0.065	0.000	0.035	0.031	0.108	0.000	0.546
cvc	0.000	0.000	0.000	0.000	0.377	0.000	0.000	0.001	0.000	0.002
dai	0.011	0.514	0.811	0.407	0.596	0.001	0.010	0.417	0.000	0.000
dash	0.000	0.016	0.000	0.000	0.055	0.000	0.000	0.000	0.000	0.000
dcr	0.885	0.002	0.033	0.075	0.597	0.092	0.729	0.003	0.000	0.007
dgb	0.232	0.000	0.100	0.102	0.029	0.000	0.157	0.003	0.000	0.000
doge	0.172	0.337	0.346	0.753	0.001	0.000	0.026	0.030	0.000	0.000
dot	0.006	0.279	0.000	0.000	0.203	0.003	0.151	0.224	0.000	0.000
drgn	0.788	0.507	0.757	0.023	0.227	0.130	0.704	0.040	0.000	0.542
elf	0.000	0.000	0.000	0.000	0.289	0.000	0.000	0.000	0.000	0.000
eos	0.049	0.006	0.002	0.012	0.000	0.002	0.029	0.003	0.000	0.192
eos_eth	0.037	0.000	0.000	0.001	0.000	0.062	0.635	0.028	0.000	0.383
etc	0.052	0.000	0.000	0.027	0.004	0.001	0.014	0.000	0.000	0.000
eth	0.379	0.000	0.121	0.001	0.009	0.024	0.011	0.000	0.000	0.000
ftt	0.019	0.007	0.001	0.085	0.121	0.000	0.013	0.000	0.000	0.538
fun	0.595	0.096	0.194	0.624	0.051	0.039	0.117	0.399	0.000	0.036
fxc	0.080	0.063	0.000	0.000	0.274	0.001	0.018	0.041	0.000	0.514
gas	0.340	0.003	0.000	0.122	0.115	0.000	0.414	0.348	0.000	0.000
gno	0.482	0.086	0.038	0.131	0.168	0.006	0.249	0.052	0.000	0.639
gnt	0.023	0.099	0.001	0.014	0.347	0.000	0.101	0.032	0.000	0.757
grin	0.000	0.001	0.322	0.000	0.083	0.000	0.002	0.001	0.000	0.037
gusd	0.007	0.016	0.043	0.000	0.215	0.000	0.021	0.032	0.000	0.006
hbtc	0.012	0.007	0.001	0.005	0.003	0.003	0.090	0.074	0.000	0.605
hedg	0.096	0.000	0.000	0.000	0.957	0.000	0.011	0.000	0.000	0.000
ht	0.001	0.006	0.009	0.006	0.109	0.000	0.000	0.004	0.000	0.464
husd	0.296	0.392	0.644	0.490	0.269	0.485	0.330	0.481	0.000	0.000
icp	0.085	0.142	0.000	0.000	0.002	0.077	0.149	0.049	0.000	0.794
kcs	0.013	0.000	0.000	0.376	0.454	0.014	0.013	0.001	0.000	0.000
knc	0.173	0.000	0.075	0.015	0.016	0.000	0.169	0.083	0.000	0.054
lend	0.002	0.003	0.028	0.008	0.000	0.000	0.002	0.008	0.000	0.002
leo_eos	0.900	0.585	0.471	0.302	0.628	0.000	0.980	0.866	0.258	0.340
leo_eth	0.852	0.528	0.554	0.224	0.639	0.000	0.736	0.580	0.023	0.499
link	0.218	0.002	0.005	0.000	0.317	0.005	0.085	0.002	0.000	0.000
loom	0.010	0.021	0.005	0.013	0.008	0.000	0.016	0.007	0.000	0.536
lpt	0.388	0.019	0.409	0.168	0.594	0.277	0.353	0.127	0.000	0.001
ltc	0.149	0.004	0.001	0.029	0.000	0.000	0.038	0.000	0.000	0.000

Table 2.11: Continued

	GRF	QRF	QR	CAV	GJR-GARCH	Hist	GRF-X	QRF-X	QR-X	GARCH-X
maid	0.239	0.000	0.000	0.160	0.013	0.105	0.316	0.000	0.000	0.465
mana	0.006	0.006	0.827	0.001	0.101	0.000	0.001	0.001	0.000	0.007
matic_eth	0.462	0.328	0.554	0.719	0.341	0.296	0.921	0.012	0.000	0.025
mkr	0.001	0.000	0.001	0.431	0.151	0.042	0.114	0.123	0.000	0.005
neo	0.290	0.016	0.182	0.002	0.157	0.006	0.310	0.292	0.000	0.022
nxm	0.052	0.000	0.000	0.002	0.005	0.045	0.046	0.001	0.000	0.244
omg	0.500	0.002	0.197	0.747	0.453	0.029	0.380	0.002	0.000	0.000
pax	0.236	0.272	0.400	0.025	0.651	0.000	0.471	0.652	0.071	0.000
paxg	0.021	0.017	0.000	0.028	0.044	0.086	0.008	0.095	0.000	0.000
pay	0.000	0.000	0.022	0.000	0.016	0.000	0.000	0.000	0.000	0.659
perp	0.581	0.463	0.017	0.599	0.022	0.644	0.627	0.159	0.000	0.126
poly	0.019	0.016	0.016	0.003	0.010	0.022	0.041	0.059	0.000	0.246
powr	0.074	0.003	0.006	0.113	0.004	0.000	0.482	0.247	0.000	0.777
ppt	0.035	0.000	0.008	0.054	0.020	0.000	0.050	0.025	0.000	0.043
qash	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
qnt	0.004	0.083	0.001	0.004	0.007	0.334	0.034	0.116	0.000	0.023
ren	0.150	0.048	0.047	0.001	0.446	0.062	0.028	0.089	0.000	0.163
renbtc	0.060	0.001	0.000	0.000	0.217	0.009	0.031	0.001	0.000	0.552
rep	0.019	0.000	0.002	0.005	0.086	0.000	0.070	0.039	0.000	0.722
rev_eth	0.010	0.060	0.000	0.000	0.760	0.000	0.033	0.030	0.000	0.103
sai	0.727	0.215	0.001	0.319	0.742	0.173	0.866	0.076	0.000	0.001
snt	0.241	0.000	0.062	0.008	0.213	0.000	0.196	0.038	0.000	0.735
snx	0.241	0.047	0.000	0.000	0.000	0.557	0.623	0.014	0.000	0.179
srm	0.198	0.186	0.000	0.000	0.000	0.433	0.499	0.205	0.000	0.265
sushi	0.030	0.000	0.000	0.000	0.000	0.020	0.063	0.000	0.000	0.039
swrv	0.001	0.000	0.000	0.000	0.000	0.001	0.004	0.045	0.000	0.678
trx	0.363	0.271	0.483	0.555	0.010	0.031	0.353	0.050	0.090	0.000
trx_eth	0.853	0.322	0.937	0.268	0.981	0.811	0.958	0.179	0.000	0.654
tusd	0.502	0.094	0.528	0.838	0.374	0.017	0.502	0.380	0.528	0.228
uma	0.000	0.001	0.000	0.000	0.001	0.000	0.010	0.007	0.000	0.062
uni	0.000	0.000	0.000	0.000	0.012	0.000	0.003	0.273	0.000	0.000
usdc	0.367	0.448	0.645	0.568	0.706	0.063	0.213	0.470	0.000	0.000
usdk	0.291	0.710	0.000	0.665	0.773	0.884	0.622	0.810	0.000	0.000
usdt	0.000	0.291	0.000	0.278	0.000	0.000	0.000	0.174	0.000	0.000
usdt_eth	0.001	0.147	0.030	0.635	0.148	0.000	0.002	0.337	0.000	0.000
usdt_omni	0.000	0.291	0.000	0.290	0.000	0.000	0.000	0.174	0.000	0.000
usdt_trx	0.081	0.784	0.258	0.939	0.752	0.004	0.152	0.853	0.000	0.000
vtc	0.001	0.000	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.001
wbtc	0.692	0.169	0.232	0.454	0.309	0.044	0.363	0.014	0.000	0.175
weth	0.171	0.000	0.233	0.000	0.021	0.000	0.000	0.009	0.000	0.000
wnxm	0.052	0.000	0.000	0.001	0.005	0.045	0.012	0.008	0.000	0.394
wtc	0.075	0.000	0.000	0.000	0.064	0.001	0.003	0.025	0.000	0.455
xaut	0.043	0.196	0.000	0.007	0.013	0.005	0.000	0.002	0.000	0.120
xem	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.013
xlm	0.543	0.001	0.005	0.002	0.228	0.000	0.152	0.000	0.000	0.000
xmr	0.082	0.000	0.000	0.182	0.045	0.000	0.092	0.000	0.000	0.246
xrp	0.010	0.000	0.000	0.000	0.022	0.000	0.001	0.000	0.000	0.000
xtz	0.463	0.005	0.133	0.011	0.787	0.007	0.003	0.006	0.000	0.000
xvg	0.177	0.014	0.100	0.234	0.547	0.001	0.518	0.032	0.000	0.000
yfi	0.407	0.105	0.000	0.001	0.000	0.045	0.419	0.087	0.000	0.002
zec	0.347	0.000	0.000	0.018	0.276	0.007	0.741	0.024	0.000	0.000
zrx	0.107	0.039	0.159	0.009	0.226	0.000	0.042	0.002	0.000	0.004

Table 2.12: Overview of All Employed Crypto Assets

id	Asset	Start-Date	End-Date	Obs.	X	Time Periods	id	Asset	Start-Date	End-Date	Obs.	X	Time Periods
1inch	1inch	2020-12-26	2022-03-21	451	7		eos	EOS	2018-06-09	2022-03-21	1382	2	3
aave	Aave	2020-10-10	2022-03-21	528	7		eos_eth	EOS ETH	2017-06-29	2018-06-02	339	7	
ada	Cardano	2017-12-01	2022-03-21	1572	7	3	etc	Ethereum Classic	2016-07-25	2022-03-21	2066	9	3
algo	Algorand	2019-06-22	2022-03-21	1004	7	3	eth	Ethereum	2015-08-08	2022-03-21	2418	9	2,3
alpha	Alpha Finance Lab	2020-10-11	2022-03-21	527	7		ftt	FTX Token	2019-08-20	2022-03-21	945	7	3
ant	Aragon	2017-08-29	2022-03-21	1666	7	3	fun	FunFair	2017-09-02	2022-03-21	1662	7	3
bal	Balancer	2020-06-25	2022-03-21	635	7		fxc	Flexacoin	2019-07-17	2021-01-25	559	6	
bat	Basic Attention Token	2017-10-06	2022-03-21	1628	7	3	gas	Gas	2017-08-08	2022-03-21	1687	7	3
bch	Bitcoin Cash	2017-08-01	2022-03-21	1694	8	3	gno	Gnosis	2017-05-02	2022-03-21	1785	7	3
bnb	Binance Coin	2017-07-15	2019-04-22	647	7		gnt	Golem (gnt)	2017-02-19	2022-03-21	1857	7	3
bnb_eth	bnb_eth	2017-07-15	2019-04-22	647	7		grin	Grin	2019-01-29	2022-03-21	1148	2	3
bsv	Bitcoin SV	2018-11-15	2022-03-21	1220	8		gusd	Gemini Dollar	2018-09-16	2022-03-21	1283	7	3
btc	Bitcoin	2010-07-18	2022-03-21	4265	9	1,2,3	hbtc	Huobi Bitcoin	2019-12-09	2022-03-21	834	5	3
btg	Bitcoin Gold	2017-10-25	2022-03-21	1609	7	3	hedg	HedgeTrade	2019-11-02	2022-01-27	818	7	
busd	Binance USD	2019-09-20	2022-03-21	914	7	3	ht	Huobi Token	2019-03-06	2022-03-21	1112	7	3
comp	Compound	2020-06-18	2022-03-21	642	7		husd	HUSD	2019-07-20	2022-03-21	976	7	3
cro	Crypto.com Coin	2019-03-20	2022-03-21	1098	7	3	icp	Internet Computer	2021-05-11	2022-03-21	315	7	
crv	Curve DAO Token	2020-08-15	2022-03-21	584	7		kcs	KuCoin Token	2020-04-04	2022-03-22	718	0	
cvc	Civic	2017-09-11	2022-03-21	1653	7	3	knc	Kyber Network	2017-09-27	2022-03-21	1637	7	3
dai	Dai	2019-11-20	2022-03-21	853	7	3	lend	Aave (lend)	2017-12-09	2022-03-21	1564	7	3
dash	Dash	2014-02-08	2022-03-21	2964	8	1,2,3	leo_eos	UNUS SED LEO EOS	2019-05-21	2022-03-21	1036	3	3
dcr	Decred	2016-05-17	2022-03-21	2133	8	2	leo_eth	UNUS SED LEO ETH	2019-05-21	2022-03-21	1036	7	3
dgb	DigiByte	2015-02-10	2022-03-21	2597	8	2,3	link	Chainlink	2017-09-29	2022-03-21	1635	7	3
doge	Dogecoin	2014-01-23	2022-03-21	2980	8	1,2,3	loom	Loom Network	2018-05-03	2022-03-21	1419	7	3
dot	Polkadot	2020-08-20	2022-03-21	579	7		lpt	Livepeer	2018-12-20	2022-03-21	1182	7	3
drgn	Dragonchain	2018-01-03	2022-03-21	1539	7	3	ltc	Litecoin	2013-04-01	2022-03-21	3277	8	1,2,3
elf	aelf	2017-12-22	2022-03-21	1551	7	3	maid	MaidSafeCoin	2014-07-10	2022-03-21	2812	7	2,3

Table 2.12: (Continued)

id	Asset	Start-Date	End-Date	Obs.	X	Time Periods	id	Asset	Start-Date	End-Date	Obs.	X	Time Periods
mana	Decentraland	2017-08-25	2022-03-21	1670	7	3	tusd	TrueUSD	2018-07-06	2022-03-21	1355	0	3
matic_eth	matic_eth	2019-04-27	2022-03-21	1060	7	3	uma	UMA	2020-09-08	2022-03-21	560	7	
mkr	Maker	2017-12-26	2022-03-21	1547	7	3	uni	Uniswap	2020-09-18	2022-03-21	550	7	
neo	Neo	2017-07-15	2022-03-21	1711	7	3	usdc	USD Coin	2018-09-28	2022-03-21	1271	7	3
nxm	Nexus Mutual	2020-08-26	2022-03-21	573	7		usdk	USDK	2020-06-13	2022-03-21	647	7	
omg	OMG Network	2017-07-15	2022-03-21	1711	7	3	usdt	Tether	2014-10-06	2022-03-21	2724	7	2,3
pax	Paxos Standard	2018-11-30	2022-03-21	1208	7	3	usdt_eth	TetherETH	2017-11-28	2022-03-21	1575	7	3
paxg	PAX Gold	2020-02-15	2022-03-21	766	7	3	usdt_omni	usdt_omni	2014-10-06	2022-03-21	2724	7	2,3
pay	TenX	2017-10-03	2022-03-21	1631	7	3	usdt_trx	TetherTRON	2019-04-16	2022-03-21	1071	7	3
perp	Perpetual Protocol	2021-02-04	2022-03-21	411	7		vtc	Vertcoin	2014-01-29	2022-03-21	2974	8	1,2,3
poly	Polymath	2018-06-15	2022-03-21	1376	7	3	wbtc	Wrapped Bitcoin	2018-11-27	2022-03-21	1211	7	3
powr	Power Ledger	2017-11-02	2022-03-21	1601	7	3	weth	Wrapped Ether	2017-12-18	2022-03-21	1555	7	3
ppt	Populous	2017-09-20	2022-03-21	1644	7	3	wnxm	Wrapped NXM	2020-08-26	2022-03-21	573	7	
qash	QASH	2017-11-06	2022-03-21	1597	7	3	wtc	Waltonchain	2017-08-28	2022-03-21	1667	7	3
qnt	Quant	2019-03-16	2022-03-21	1102	7	3	xaut	Tether Gold	2020-02-24	2022-03-21	757	7	3
ren	Ren	2018-12-07	2022-03-21	1201	7	3	xem	NEM	2015-04-01	2022-03-21	2547	2	2,3
renbtc	renBTC	2020-05-13	2022-03-21	678	7	3	xlm	Stellar	2015-09-30	2022-03-21	2365	7	2,3
rep	Augur	2016-10-04	2022-03-21	1995	7	3	xmr	Monero	2014-05-20	2022-03-21	2863	3	2,3
rev_eth	rev_eth	2020-03-26	2022-03-21	726	7	3	xrp	XRP	2014-08-15	2022-03-21	2776	7	2,3
sai	Sai	2017-12-23	2019-11-30	708	7		xtz	Tezos	2018-06-30	2022-03-21	1361	7	3
snt	Status	2017-06-20	2022-03-21	1736	7	3	xvg	Verge	2017-09-30	2022-03-21	1634	8	3
snx	Synthetix	2020-04-09	2022-03-21	712	7	3	yfi	yearn.finance	2020-07-25	2022-03-21	605	7	
srm	Serum	2020-08-11	2022-03-21	588	7		zec	Zcash	2016-10-29	2022-03-21	1970	8	3
sushi	SushiSwap	2020-09-01	2022-03-21	567	7		zrx	0x	2017-08-11	2022-03-21	1684	7	3
swrv	Swerve	2020-09-22	2022-03-21	546	7								
trx	TRON	2018-06-25	2022-03-21	1366	2	3							
trx_eth	TronETH	2017-10-07	2018-06-25	262	7								

Notes: Obs. is the number of observations in total, i.e. including also the time points used for training. X depicts the number of additional covariates contained for each cryptocurrency, and Time Periods describes the sub-periods that the specific currency is in. Blank spaces indicate that a currency is not contained in any sub-period, which can occur when the currency is not available for the full out-of-sample sub-period.

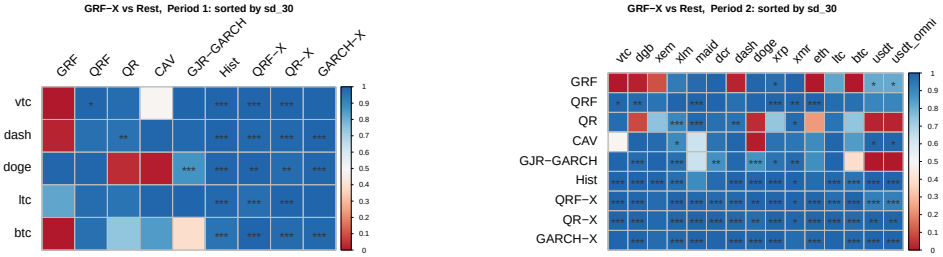
Table 2.13: External Covariates and Descriptions

Variable Name	Coding Coinmetrics	Description
Active_Users	AdrActCnt	The number of unique active daily addresses
Total_Users	AdrBalCnt	The number of unique addresses that hold any amount of native units of that currency
Total_Users_USD100	AdrBalUSD100Cnt	The number of unique addresses that hold at least 100 USD of native units of that currency
Total_Users_USD10	AdrBalUSD10Cnt	The number of unique addresses that hold at least 10 USD of native units of that currency
SER	SER	The supply equality ratio, i.e. the ratio of supply held by addresses with less than 1 over 10 millionth of the current supply to the top one percent of addresses with the highest current supply
Transactions	TxCnt	The number of daily initiated transactions
Velocity	VelCur1yr	The velocity of supply in the current year, which describes the the ratio of current supply to the sum of the value transferred in the last year

Notes: Variable coding corresponds to `https://docs.coinmetrics.io/`. Detailed variable descriptions are available on `https://docs.coinmetrics.io/info/metrics`.

Table 2.14: Performance and Significance of CPA-tests with GRF-X Over Different Time Periods

GRF-X vs.:	GRF	QRF	QR	CAV	GJR-GARCH	Hist	QRF-X	QR-X	GARCH-X
Share of GRF With Better Performance									
Period 1	0.40	1.00	0.80	0.60	0.80	1.00	1.00	1.00	1.00
Period 2	0.60	1.00	0.67	0.87	0.80	1.00	1.00	1.00	1.00
Period 3	0.55	0.90	0.66	0.75	0.66	0.90	0.94	0.96	0.73
Full Data	0.54	0.89	0.72	0.76	0.73	0.83	0.92	0.97	0.76
Share of GRF With Significantly Better Performance									
Period 1	0.00	0.20	0.20	0.00	0.20	1.00	1.00	1.00	0.60
Period 2	0.20	0.40	0.27	0.20	0.40	0.80	0.93	0.93	0.67
Period 3	0.12	0.40	0.19	0.23	0.19	0.58	0.73	0.81	0.51
Full Data	0.13	0.39	0.30	0.24	0.19	0.47	0.69	0.86	0.52



Notes: The top part shows summary values that are shares over all cryptocurrencies in the respective time period. It describes the number of times that GRF-X had a better performance (i.e. more than 50% of predicted losses by the CPA test were smaller for the GRF-X) relative to all crytpocurrencies (in that period), and the number of times that GRF-X was significantly better (at least at a 10% level) as judged by the CPA test over all crytpocurrencies (in that period). The bottom part shows the detailed results of CPA-tests with the color of each box indicating the performance of GRF-X. Blue signifies a performance of 1, meaning that GRF-X has a smaller predicted loss in 100% of cases. *, **, *** shows significance on a level of 10%, 5%, and 1%. The values are ordered by 30 day lagged standard deviation from highest to lowest.

Table 2.16: Difference Between Covariates of Cryptos Where GRF-X is Better vs. Worse

	Period 1	Period 2	Period 3	Full Data
Ret	1.20	0.63	-0.40	-0.08
Active_Users	0.34	0.22	0.32	0.39
Total_Users	0.79	0.26	0.31	0.34
Total_Users_USD100	0.22	0.23	0.22	0.27
Total_Users_USD10	0.41	0.23	0.16	0.20
CapMrktCurUSD	0.08	0.16	0.45	0.38
SER	1.07	0.72	0.17	0.22
Transactions	0.42	0.06	3.26	3.74
Velocity	4.68	4.44	1.31	1.23
sd_3	0.89	0.31	0.65	0.65
sd_7	0.91	0.32	0.64	0.64
sd_30	0.90	0.33	0.63	0.64
sd_60	0.91	0.34	0.63	0.65

Notes: Values are shares of groups of cryptocurrencies where at least two of CAV, QR, and GJR-GARCH have better CPA-performance than GRF-X divided by the remaining rest. Raw values before division are mean values over all cryptocurrencies for the median of each covariate in the respective time period.

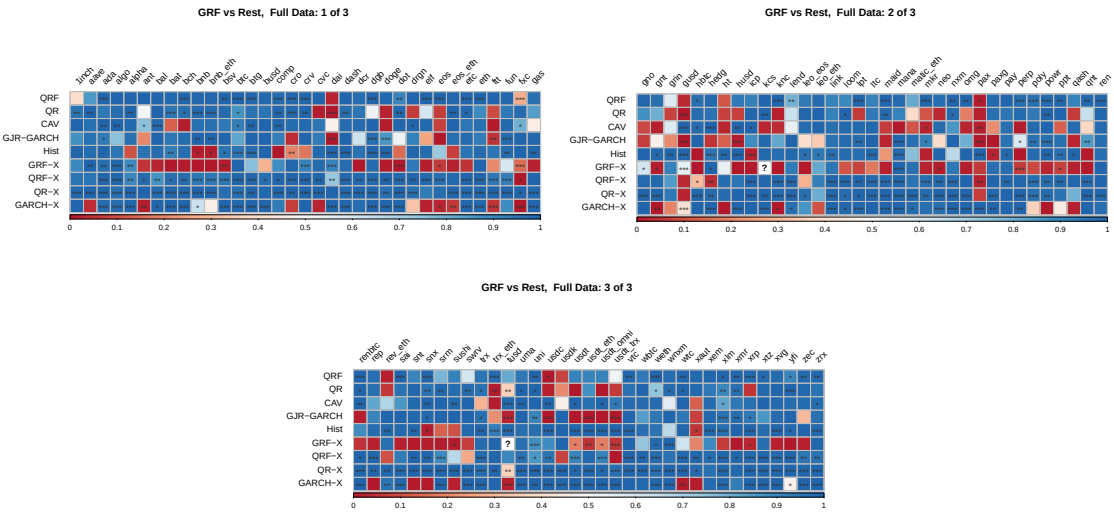


Figure 2.10: Overview of Results for CPA-Tests of GRF vs. All Other Methods for the Full Data Ordered Alphabetically

Note: The color of each box indicates the performance of GRF, with 1 indicating that GRF has a smaller predicted loss in 100% of cases. *, **, *** indicate significance on a level of 10%, 5%, and 1%.

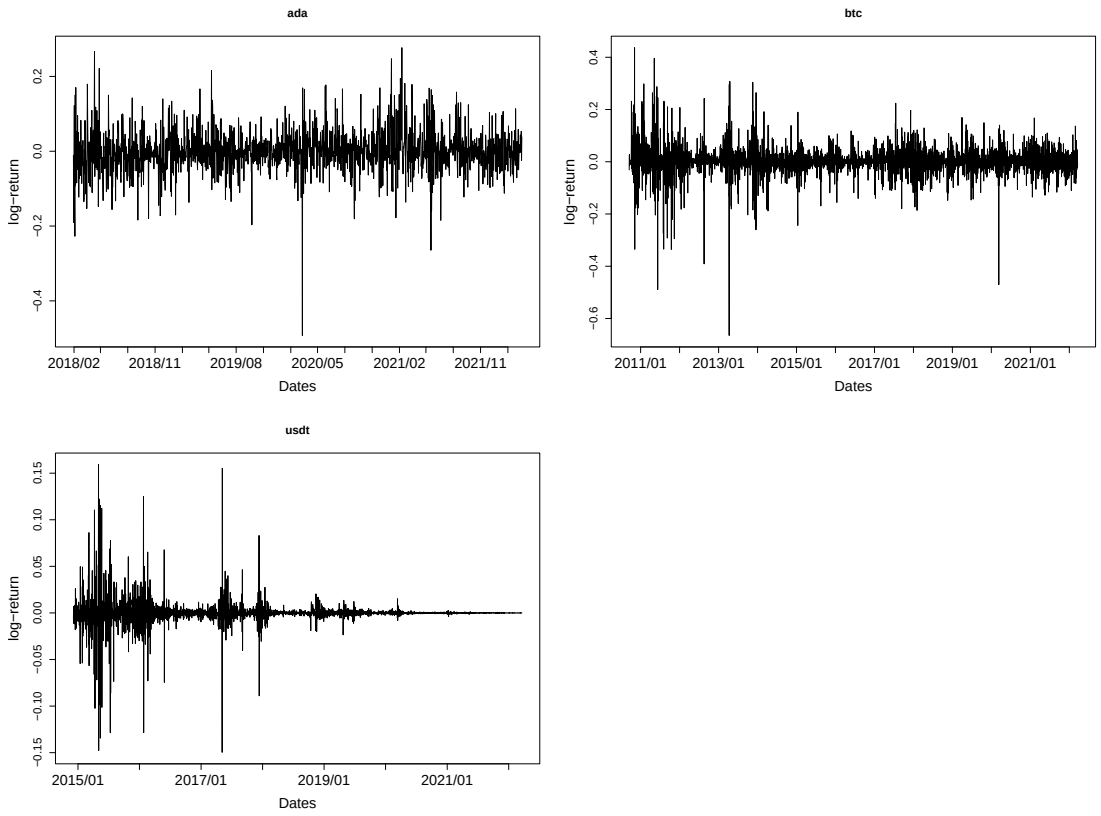


Figure 2.11: Log-Returns of the Specific Cryptocurrencies Analyzed in Subsection 2.5.2

Table 2.17: Simulation: 1% VaR

Rolling Window	$l = 500$				$l = 1000$			
	DQ	Kupiec	Christoffersen	AoE	DQ	Kupiec	Christoffersen	AoE
Sim GARCH Normal								
QRF	0.995 (0.002)	0.910 (0.017)	0.855 (0.029)	1.894	0.830 (0.053)	0.580 (0.128)	0.395 (0.180)	1.751
GRF	0.635 (0.121)	0.120 (0.396)	0.115 (0.388)	1.265	0.410 (0.318)	0.100 (0.459)	0.050 (0.518)	1.193
QR	0.945 (0.018)	0.500 (0.122)	0.485 (0.155)	1.575	0.520 (0.235)	0.140 (0.433)	0.090 (0.469)	1.273
Hist	0.830 (0.046)	0.045 (0.421)	0.195 (0.339)	1.231	0.675 (0.171)	0.120 (0.477)	0.140 (0.456)	1.122
NormFit	0.700 (0.116)	0.055 (0.470)	0.200 (0.392)	1.165	0.625 (0.209)	0.130 (0.441)	0.145 (0.450)	1.152
CAViaR	0.890 (0.038)	0.365 (0.181)	0.340 (0.215)	1.486	0.450 (0.304)	0.130 (0.435)	0.080 (0.508)	1.245
GARCH(1,1)	0.440 (0.268)	0.060 (0.468)	0.150 (0.412)	1.128	0.310 (0.408)	0.060 (0.532)	0.075 (0.525)	1.072
Sim GARCH t								
QRF	0.990 (0.002)	0.890 (0.016)	0.845 (0.029)	1.942	0.815 (0.060)	0.665 (0.108)	0.500 (0.152)	1.801
GRF	0.625 (0.127)	0.135 (0.372)	0.155 (0.396)	1.286	0.410 (0.297)	0.100 (0.455)	0.055 (0.523)	1.222
QR	0.925 (0.014)	0.570 (0.106)	0.575 (0.134)	1.676	0.565 (0.227)	0.165 (0.393)	0.130 (0.431)	1.290
Hist	0.755 (0.077)	0.080 (0.412)	0.215 (0.385)	1.232	0.665 (0.160)	0.125 (0.454)	0.175 (0.434)	1.133
NormFit	0.890 (0.028)	0.655 (0.092)	0.670 (0.096)	1.693	0.760 (0.083)	0.510 (0.184)	0.450 (0.209)	1.669
CAViaR	0.815 (0.042)	0.320 (0.190)	0.310 (0.230)	1.466	0.425 (0.298)	0.155 (0.410)	0.090 (0.472)	1.248
GARCH(1,1)	0.855 (0.045)	0.750 (0.057)	0.745 (0.068)	1.769	0.595 (0.174)	0.500 (0.176)	0.430 (0.196)	1.680
Sim SAV-Model								
QRF	0.990 (0.001)	0.940 (0.015)	0.835 (0.028)	1.891	0.810 (0.063)	0.575 (0.133)	0.405 (0.187)	1.749
GRF	0.585 (0.140)	0.065 (0.417)	0.045 (0.462)	1.251	0.300 (0.352)	0.080 (0.460)	0.035 (0.550)	1.180
QR	0.940 (0.015)	0.445 (0.129)	0.375 (0.179)	1.553	0.560 (0.200)	0.190 (0.406)	0.090 (0.473)	1.289
Hist	0.360 (0.239)	0.060 (0.515)	0.075 (0.534)	1.195	0.275 (0.415)	0.060 (0.513)	0.060 (0.597)	1.097
NormFit	0.215 (0.449)	0.105 (0.452)	0.110 (0.527)	1.166	0.240 (0.520)	0.135 (0.448)	0.090 (0.553)	1.147
CAViaR	0.855 (0.035)	0.255 (0.207)	0.250 (0.239)	1.444	0.485 (0.273)	0.075 (0.439)	0.050 (0.518)	1.221
GARCH(1,1)	0.215 (0.407)	0.090 (0.502)	0.090 (0.532)	1.155	0.210 (0.513)	0.105 (0.471)	0.055 (0.574)	1.122
Sim GARCH Bitcoin fit								
QRF	0.975 (0.004)	0.910 (0.017)	0.855 (0.029)	1.894	0.750 (0.090)	0.580 (0.128)	0.395 (0.180)	1.751
GRF	0.470 (0.207)	0.120 (0.396)	0.115 (0.388)	1.265	0.275 (0.424)	0.100 (0.459)	0.050 (0.518)	1.193
QR	0.880 (0.040)	0.500 (0.122)	0.485 (0.155)	1.575	0.370 (0.328)	0.140 (0.433)	0.090 (0.469)	1.273
Hist	0.675 (0.090)	0.045 (0.421)	0.195 (0.339)	1.231	0.530 (0.259)	0.120 (0.477)	0.140 (0.456)	1.122
NormFit	0.540 (0.203)	0.055 (0.470)	0.200 (0.392)	1.165	0.500 (0.296)	0.130 (0.441)	0.145 (0.450)	1.152
CAViaR	0.745 (0.091)	0.105 (0.405)	0.155 (0.400)	1.243	0.435 (0.345)	0.045 (0.486)	0.030 (0.555)	1.046
GARCH(1,1)	0.345 (0.328)	0.065 (0.459)	0.190 (0.385)	1.157	0.215 (0.510)	0.050 (0.537)	0.070 (0.512)	1.082

Notes: Results for 1% VaR show rejection rates of t-tests of empirical levels against the nominal level of 1% for DQ-, Kupiec- and Christoffersen-tests and mean p-values in parentheses. Higher p-values and lower rejection rates indicate better model performance. GARCH depicts an oracle GARCH-model that fits an GARCH(1,1) process with normally distributed errors.

3 Controlling False Discoveries With Robust Knockoffs: Uncovering Macroeconomic Factors of Bond Recovery Rates

3.1 Introduction

In large-scale financial and economic systems with many potentially influencing variables for a target quantity, there is a key interest in detecting the relevant driving factors in a data-driven way. Such a fully data-adaptive choice of factors yields transparency in the identification of important channels avoiding biases from insufficient pre-specification but also inspiring and complementing future model building. With the availability of new machine-learning (ML) based techniques, there has recently been considerable effort in particular in the empirical asset pricing literature to use such approaches for augmented pricing and prediction results (see e.g. Chen et al. (2019), Freyberger et al. (2020), Chinco et al. (2019)) but also for bond quality determination (Qi and Zhao, 2011; Nazemi et al., 2022).

We contribute to this literature by proposing a new knockoff-type methodology building on e.g. Candès et al. (2018) that offers control over the rate of falsely selected variables. With the false discovery rate (FDR) as the key hyperparameter, the selection and prediction performance based on the proposed technology is dominated by the FDR. It is thus transparent and interpretable while being data-driven since the FDR can be directly estimated as the empirical proportion of false discoveries (FDP). Note that this is in contrast to e.g. LASSO-type approaches (see e.g.Tibshirani (1996)), where penalty

parameters can be chosen adaptively but have no stand-alone interpretation and meaning, which often creates a black-box connotation. Moreover, our technique gains robustness from simultaneously taking several nominal FDR-levels into account. In this way, we mitigate hyperparameter pre-selection effects and obtain robustness of results in the presence of time-dependent data. Both points are key for valid selection results in practice. We show that the proposed methodology provides interesting insights in detecting novel relevant factors for corporate bond recovery rates which might be important from a business but also regulatory perspective. In particular, we study the recovery rates of 2,079 U.S. corporate bonds that defaulted between 2001 and 2016 depending on industry and stock specific information from Bloomberg Financial Markets and 144 macroeconomic market variables from the Federal Reserve Economic Data (FRED). For this, we also document superior out-of sample performance of the resulting sparse model using only relevant factors comparing them to state-of-the-art machine learning models on the entire and the selected set of predictors and to LASSO-type specifications. We confirm our point-wise ranking with results from model confidence sets (Hansen et al., 2011).

In particular, the proposed robustification technique works for the entire set of different knockoff baseline procedures from model-X (Candès et al., 2018) to deep knockoffs (Romano et al., 2020) to group versions (Dai and Barber, 2016) and mitigates the influence of hyperparameter input levels and data dependence challenges. We address the hyperparameter influence problem by proposing several weighted aggregation schemes for variable selection rates of different FDR-levels. By considering different weighting schemes, we account for vanishing scrutiny of the procedures in size of FDR-levels but extract the information from each level for the overall selection result. Secondly, we use a repeated subsampling scheme to control for the variability of the knockoff procedures, which themselves are random. While this shares similarities with Ren et al. (2021), we employ subsampling (see e.g. Meinshausen and Bühlmann (2010)), which provides robustness in the presence of correlated observations and high outliers. This is of key importance for the determination of relevant factors of the considered corporate recovery rates. In this empirical study, we additionally employ principal component analysis (PCA) on groups of macroeconomic variables of similar type to reduce cross-sectional correlation of the knockoff input factors while retaining interpretability on the group level. We

investigate the performance of the proposed methodology for different baseline procedures and show that an ensemble yields superior out-of-sample prediction results. Generally for many financial applications, a rapidly growing literature on ML-based approaches has emerged in particular in empirical asset pricing. These comprise approaches with a strong focus on prediction that use neural networks, other general nonparametric and principal component-type techniques (Chen et al., 2019; Freyberger et al., 2020; Kelly et al., 2019). In this context, variable selection methods serve as a dimension reduction device and are mostly based on the lasso framework (Tibshirani, 1996) with respective sparsity assumptions as e.g. in Chinco et al. (2019); Feng et al. (2020); Freyberger et al. (2020).

There has also been considerable research on recovery rates and loss given default (LGD). Studies that focus on prediction of LGD are Leow and Mues (2012) for mortgage loans, while Qi and Zhao (2011) and Yao et al. (2015) employ machine learning methods for corporate bonds. Other research focuses on identifying macroeconomic factors of LGD for various types of loans (Keijsers et al., 2018), often using machine learning techniques (Bellotti and Crook, 2012; Kaposty et al., 2020; Kellner et al., 2022). Recent studies on recovery rates focus on prediction mostly employing ML-methods, either for corporate bonds (Nazemi et al., 2018, 2022) or for other types of loans such as credit default swaps or consumer credit (Das and Hanouna, 2009; Jansen et al., 2018; Bellotti et al., 2021). Amongst others, these studies use random forests (Breiman, 2001), vector regression (Suykens and Vandewalle, 1999), and power expectation propagation (Bui et al., 2017). Recent studies that identify factors for corporate bond recovery rates are mainly Jankowitsch et al. (2014) and Nazemi et al. (2022). For a full overview also about older studies that consider recovery rates for corporate bonds, see Jankowitsch et al. (2014), Nazemi and Fabozzi (2018), and the references therein. For recovery rates, we built on initial results of a pre-study in Nazemi and Fabozzi (2018) using a far more comprehensive data set and a group structure adaptive selection technique without requiring unrealistic sparsity in any form.

The rest of the paper is structured as follows. Section 3.2 introduces our proposed methodology in a general setting, while Section 3.3 focuses on the application. Therein,

Section 3.3.1 introduces the data set of corporate bond recovery rates and Section 3.3.2 presents the main results of our analysis. Finally, we conclude in Section 3.4.

3.2 Model Selection With Knockoffs

For robust data-driven variable selection, we propose a novel knockoff-type procedure (e.g. Candès et al., 2018) that offers direct control of the false discovery rate (FDR). The key hyperparameter FDR corresponds to the number of coefficients determined as non-zero while being truly zero relative to all obtained non-zero factors. FDR is well-known from the multiple testing literature as the type I error but has also been shown to directly link to the size of error rates in model estimation and prediction of after pre-selection with knockoffs (Barber and Candès, 2019). Thus setting the acceptable (nominal) FDR-level for knockoffs directly controls estimation and prediction performance of the resulting model, while the performance of other model selection techniques depends on parameters that lack a direct interpretation. Interpretable hyperparameters, however, are key for adequate tuning and eventual transparency of the results. Moreover, the suggested knockoffs work irrespective of the type of underlying sparsity and in particular for high-dimensional cases, results are not dependent on a specific form of sparsity. We contribute a robust knockoff version which works in particular in the presence of strong cross-sectional and time dependence. This is key in many economic and financial applications and of peculiar importance for our application of the determination of relevant factors of corporate recovery rates.

We work in the following setting, where $y_i \in \mathbb{R}$ and $X_i \in \mathbb{R}^p$ are observed for $i = 1, \dots, n$ but only some unknown subset of the p components in X is relevant for y and forms the so-called active set $\mathcal{S}$ of X, i.e. for $j \notin \mathcal{S}$, y is independent of component $X_{(j)}$ conditional on $(X_{(k)})_{k \in \mathcal{S}}$. These components $\mathcal{S}$ should be selected in

$$y_i = f(X_i) + \epsilon_i \ , \tag{3.1}$$

with an error term $\epsilon_i \in \mathbb{R}$ and a function $f(\cdot)$ that describes the impact of $X_i = (X_{i1}, \dots, X_{ip}) = (X_{ij}, X_{i-j})$ for any $j = 1, \dots p$ on y_i. In general, we assume that $f(X_i) = X_i\beta$, with $\beta \in \mathbb{R}^p$ for an easily interpretable structure, but we also include unknown nonparametric versions of f in our application. We set as usual $Y = (y_1, \dots, y_n)'$

and $X = (X_1', \ldots, X_n')' = (X_{(1)}, \ldots, X_{(p)}) = (X_{(j)}, X_{(-j)})$ with $X_{(j)} \in \mathbb{R}^n$ for all $j = 1, \ldots, p$.

In the literature, there exist different procedures for the construction of knockoffs such as model-X knockoffs (Candès et al., 2018), deep knockoffs (Romano et al., 2020), and group-knockoffs (Dai and Barber, 2016). They all build on the same main idea to compare the regressors of interest X with randomly generated knockoffs $\tilde{X} = (\tilde{X}_{(1)}, \ldots, \tilde{X}_{(p)})$ that fulfill two properties:

(i) pairwise exchangeability, i.e. the distribution of $(X_{(j)}, X_{(-j)}, \tilde{X}_{(j)}, \tilde{X}_{(-j)})$ and $(\tilde{X}_{(j)}, X_{(-j)}, X_{(j)}, \tilde{X}_{(-j)})$ is identical

(ii) Y is independent of $\tilde{X}$ conditional on X.

When regressing Y on $(X, \tilde{X})$ jointly, only those regressor components in X which fundamentally differ from their corresponding ones in the random $\tilde{X}$ according to a variable importance measure are judged as relevant and are part of the active set $\mathcal{S}$. The variable importance depends on model and estimation techniques, but for the linear case, e.g. the difference of absolute lasso coefficients of $X_{(j)}$ and $\tilde{X}_{(j)}$ must be large enough.

For model-X knockoffs (Candès et al., 2018), the two knockoff conditions (i) and (ii) are addressed by matching first and second moments of of X and $\tilde{X}$ in the construction of $\tilde{X}$ subject to independence of Y and $\tilde{X}$ conditional on X. Matching expectations is straightforward and the second order construction leads to a convex optimization problem minimizing pairwise correlations of X and $\tilde{X}$ under the constraint that $Cov(X, \tilde{X})$ be positive semi-definite. This effectively targets the off-diagonal elements of the covariance of X and $\tilde{X}$ and leads to the approximate semi-definite program algorithm (ASDP) of Candès et al. (2018). The obtained model-X knockoffs $\tilde{X}$ approximately fulfill conditions (i) and (ii), and the construction is exact if $(X, \tilde{X})$ are normal. For a more detailed description of the construction of the more general deep knockoffs (Romano et al., 2020) which fully operationalize the distributional form of (i) and (ii) and of group-knockoffs that use a pre-specified group-structure (Dai and Barber, 2016), see Appendix 3.5.1. In general, the construction principle of all knockoff techniques is based on the standard Gaussian results in Barber and Candès (2015) and the non-Gaussian, high-dimensional extension in Candès et al. (2018) to the model-X knockoff filters.

Once the knockoffs have been constructed, they can be used as a filtering device to select the active set. For this, each knockoff feature $\tilde{X}_{(j)}$ is compared to its true counterpart $X_{(j)}$ via a feature-statistic W_j for all $j = 1, \ldots, p$. In the linear case, for a lasso regression of y on the joint $(X, \tilde{X})$ over a grid of penalty parameters λ with corresponding coefficients $\hat{\beta}_j(\lambda)$ we work with $\lambda_j = \sup\{\lambda | \hat{\beta}_j(\lambda) \neq 0\}$ as the largest λ for which variable j is in the active set and define

$$W_j^{LCD} = |\hat{\beta}_j(\lambda_0)| - |\hat{\beta}_{j+p}(\lambda_0)| \tag{3.2}$$

$$W_j^{LSM} = \text{sgn}(\lambda_j - \lambda_{j+p}) \max(\lambda_j, \lambda_{j+p}) \ . \tag{3.3}$$

where λ_0 is chosen according to some global criterion like cross-validation and the $\text{sgn}(\cdot)$ function returns the sign of the input. Note that the W_j from equations (3.2) and (3.3) correspond to the lasso coefficient difference (LCD) of the model-X knockoffs and the lasso signed max (LSM) as described in Barber and Candès (2015), respectively. In practice, we mostly rely on the LCD measure which was shown to be preferable and robust to highly correlated features as in our application (Candès et al., 2018). Only for group knockoffs, we use the proposed adapted version of the LSM as suggested by Dai and Barber (2016). We only select variable components j as part of the active set $\mathcal{S}$ if W_j is greater or equal to some threshold T with

$$T = \min\left\{ t > 0 : \frac{\#\{j : W_j \leq -t\}}{\#\{j : W_j \geq t\}} \leq \alpha \right\} \ , \tag{3.4}$$

where $\alpha \in [0, 1]$ is the pre-specified level of acceptable (nominal) false discovery rate $FDR = \mathbb{E}[FDP]$, where $FDP = \dfrac{|\hat{S} \setminus S|}{|\hat{S}|}$ is the false discovery proportion, with $\hat{S}$ as the set of selected variables[1] and S as the set of truly relevant variables. In practice, we calculate this proportion as $\widehat{FDP} = \dfrac{\#\{j : W_j \leq -t\}}{\#\{j : W_j \geq t\}}$. Note that for the definition of T we rely on Candès et al. (2018, Equation 3.9) of the original suggestion of Candès et al. (2018).

We suggest two routes for robustification of the knockoffs.
First, we propose an adapted version of the baseline knockoff procedures that can deal with time-dependence and an unknown, possibly non-standard covariate distribution due

[1] In case $\hat{S}$ is empty, we set $FDR = 0$ as in Candès et al. (2018).

to high correlations among X. Secondly, we examine the full grid of possible nominal FDR-levels for the knockoff procedure to uncover dependence of selections on certain specific FDR-levels. We do this by repeating each robustified baseline procedure K times over a grid of K nominal FDR-values and combine the results by weighting the selection probabilities depending on the FDR-level. We call this procedure *weighted FDR selection* (wFDR).

With covariate distributions different from normality and possible time-dependence, the standard assumptions from the Knockoff framework are violated, which could lead to strong variability of knockoff selections that are per definition random. We therefore suggest repeated subsampling to stabilize the selection procedure, motivated by the stability selection of Meinshausen and Bühlmann (2010) and Ren et al. (2021), who suggest a similar procedure, where the knockoff procedure is repeated without subsampling.. We repeat the full knockoff procedure $B = 100$ times only using a subsample of the full n observations, with subsampling rate θ. The subsampling ensures that large outliers and data artifacts do not majorly affect the selection, while the repetition of the knockoff procedure controls the randomness of the knockoffs. This randomness would also allow no subsampling at all as in Ren et al. (2021). For our application with a substantial amount of outliers in finite samples, however, we choose to use $\theta = 0.9$. For a fixed FDR-level α, the procedure ranks variables in decreasing order according to their selection frequency, i.e. empirical selection probability. The variable with the highest selection frequency receives rank p, the second most selected variable gets rank $p - 1$, up to the least selected variable receiving rank 1. Alternatively, we also directly work with the selection probability instead of the ranks, which puts a larger emphasis on the variability of selection probabilities[2]. See also Method 2 for details. Note that by construction the proposed subsampling adapted knock-off procedure keeps the fixed FDR-level α but robustifies the selection result.

Moreover, we propose to conduct each knockoff-baseline selection (Method 2) over a grid of K different FDR values α_k jointly. Thus, for each fixed-level α_k, we detect whether a variable is relevant or not and determine the corresponding selection probability

[2]Technically, it would also be possible to run a standard knockoff machine without subsampling and report either zero (no selection) or one (selection) for each variable. We refrain from such an approach due to the data challenges stated above.

Algorithm 2 Repeated Subsampling for Knockoffs

Input:

Observation pairs $(X, Y) = (X_i, y_i)_{i=1}^n \in \mathbb{R}^{n \times p+1}$

Nominal FDR-level $\alpha \in [0, 1]$

Knockoff procedure $Knock_{proc}$, e.g. model-X knockoffs or deep knockoffs

Subsampling rate θ and number of repetitions B

1: **for** b in 1 to B **do**

2: Draw random subsample $(X_b^{sub}, Y_b^{sub}) = (X_b^s, y_b^s)_{s=1}^{n_{sub}}$ (i.e. without replacement) of size $n_{sub} = \lfloor n\theta \rfloor$

3: Apply $Knock_{proc}$ based on (X_b^{sub}, Y_b^{sub}) and obtain $Ind_b = (ind_1^b, \ldots, ind_p^b)$, where ind_l^b is one if variable l is selected and zero otherwise, $l = 1, \ldots, p$

4: **end for**

5: Compute selection probability pr_l for each variable $l = 1, \ldots, p$ as $pr_l = \dfrac{\sum_{j=1}^B ind_l^j}{B}$

Output: Selection probabilities for each variable $P_\alpha = (pr_1, \ldots, pr_p)$

via subsampling using our methodology. Over the grid of different α_k-values, the corresponding selection probabilities are then weighted depending on the level α_k, where higher values of α_k receive lower weights corresponding to the definition of the FDR. The final selection probability for each variable is then obtained as the weighted sum of all selection probabilities for this component over all α_k. Since the number of selected variables varies depending on the respective α-level, our procedure prevents situations where results crucially depend on the pre-setting of one specific α-level. We show explicitly that such situations happen in our empirical example where high correlations between variables exist and solve this issue by combining the results from distinct weighting schemes that control the influence of selections over the grid of possible FDRs. With that, we transparently control the FDR influence on selections while maintaining flexibility by not restricting the baseline procedure for selections too strongly. An overview of our method is given in Method 3. We suggest two different weighting schemes that all depend on the following observation. By definition, a low nominal FDR implies that the number of falsely selected variables is small compared to the number of selected variables. This suggests that weighting should be conducted in a way that low FDRs, for which

Algorithm 3 Weighted FDR Selection

Input:

Observation pairs $(X, Y) = (X_i, y_i)_{i=1}^n \in \mathbb{R}^{n \times p+1}$

Set of nominal false discovery rates $FDR_k = \alpha_k \in [0, 1]$, $k = 1, \ldots, K$

Baseline procedure $B((X, Y), FDR_k)$ that returns selection probabilities for X given FDR_k, e.g. as in Method 2

Weighting scheme $\omega = (\omega_1, \ldots, \omega_K)$ that assigns a weight depending on selection run k

1: **for** k in 1 to K **do**

2: Run baseline procedure $B(X, Y, FDR_k)$ for FDR_k and obtain selection probabilities $P_k = (pr_{1k}, \ldots, pr_{lk})$ for each variable $l = 1, \ldots, p$

Possible P_k format: 0/1-coding, rank, probabilities, see Method 2 for computation

3: **end for**

4: **for** l in 1 to p **do**

5: Obtain weighted selection probability $WP_l = \sum_{k=1}^K pr_{lk}\omega_k$

6: **end for**

Output: Final weighted selection probabilities for each variable $WPr = (WP_1, \ldots, WP_p)$

selection probabilities are thus more informative, should receive higher weight. Imagine we have two selection probabilities pr_j^{low} and pr_k^{high} for variable j and k at nominal levels $\alpha_{low} = 0.1$ and $\alpha_{high} = 0.95$, respectively. For α_{low}, less than 10% selected variables should be false selections, while for α_{high} less than 95% of selections should be false. Obviously, when pr_j^{low} and pr_k^{high} are very similar, one would give variable j a higher weight in being a true influencing variable compared to variable k. To formalize this intuition, we propose two weighted averages and compare them with an unweighted baseline. One average is just based on weights that decay linearly, while the other one uses weights that decay exponentially and are equidistant on the log-scale. More specifically, for the value (probability or rank) at $FDR_k = \alpha_k$, where $k = 1, \ldots, K$, and $FDR_k \in (0, 1)$ on an equidistant grid, the linear weight for position k on the FDR-grid

is given by

$$\omega_k^{lin} = \frac{K-k+1}{\sum_{k=1}^{K} k} \ , \tag{3.5}$$

and the exponentially decaying weight is given by

$$\omega_k^{exp} = \frac{\exp\left(ln(K) - (k-1)\frac{ln(K)-ln(1)}{K-1}\right)}{\sum_{k=1}^{K} \exp\left(ln(K) - (k-1)\frac{ln(K)-ln(1)}{K-1}\right)} \ . \tag{3.6}$$

We compare these weights with an unweighted average, where we expect the weighted averages to be more informative and thus give better indications of true influencing variables than their unweighted counterparts.

3.3 Empirical Study: Corporate Recovery Rates

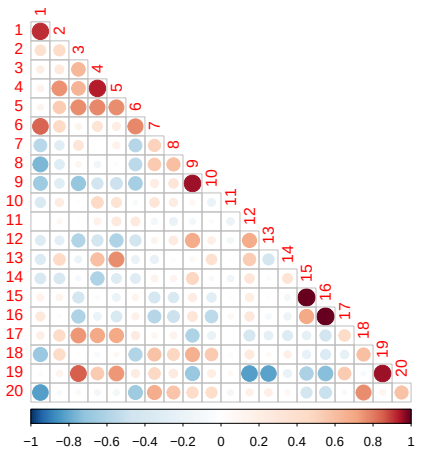
3.3.1 Data

Our empirical study uses a data set consisting of 2,079 U.S. corporate bonds that defaulted between 2001 and 2016 obtained from S&P Capital IQ-similar. We retrieved industry and stock variables from Bloomberg Financial Markets. Moreover, we collected 144 macroeconomic variables that were used in previous credit risk studies from the Federal Reserve Bank of St. Louis (FRED, Federal Reserve Economic Data). We classified these macroeconomic variables into 20 groups as detailed in Appendix C. We structured the groups according to financial conditions (Loans, Bank Credit and Debt), monetary measures (Savings, CPIs, Money Supply), corporate measures (Cash Flow and Profit), business cycle (Unemployment, Industrial Production, Private Employment, Housing, Income, Real GDP, Inventories), stock market (Index Returns and Volatilities), international competitiveness (Exchange Rates, Trade), and micro-level factors (Producer Price Index). These groups are tailored to yield interpretable factors and are more granular than in Nazemi et al. (2022), who consider a prediction-focused analysis. As can be seen from Table 3.1, variables within groups are often highly correlated, which makes it hard to directly analyze them without transforming the data. Although there are still a few highly dependent groups, the correlation between groups is much smaller

Table 3.1: Summary Statistics of the Distribution of Pairwise Correlations: Detailed Within and Schematic Cross-Group

Group	Min	25%	50%	75%	Max	# members
1: Financial Conditions: Loans	0.50	0.78	0.92	0.97	1.00*	6
2: Monetary Measures: Savings	0.14	0.29	0.43	0.67	0.91	3
3: Monetary Measures: CPIs	-0.91	-0.17	0.63	0.94	1.00*	13
4: Monetary Measures: Money Supply	0.90	0.91	0.93	0.96	1.00*	4
5: Corporate Measures: Cash Flow and Profit	0.40	0.66	0.76	0.88	0.94	4
6: Business Cycle: Unemployment	-0.29	0.51	0.78	0.92	1.00*	10
7: Business Cycle: Industrial Production	-0.54	0.20	0.51	0.83	0.98	13
8: Business Cycle: Private Employment	-0.26	0.20	0.59	0.79	0.98	10
9: Business Cycle: Housing	0.68	0.91	0.96	0.97	0.99	12
10: Business Cycle: Income	-0.32	-0.28	-0.21	0.48	0.99	4
11: Stock Market: Index Returns and Volatilities	-0.66	-0.40	-0.20	0.40	0.99	9
12: International Competitiveness: Exchange Rates	-0.66	-0.26	0.68	0.81	0.97	5
13: International Competitiveness: Trade	-0.81	-0.65	-0.43	0.34	0.96	5
14: Micro-level: Bond Yields and Rates	-0.90	-0.40	0.34	0.86	1.00	20
15: Micro-level: Bond Defaults in Industry	-	-	-	-	-	1
16: Micro-level: High Yield Default Rate	-	-	-	-	-	1
17: Financial Conditions: Bank Credit and Debt	-0.87	-0.17	0.45	0.90	1.00*	11
18: Business Cycle: Real GDP	0.35	0.47	0.59	0.74	0.89	3
19: Micro-level: Producer Price Index	0.81	0.92	0.96	0.98	1.00*	6
20: Business Cycle: Inventories	0.29	0.32	0.59	0.83	0.98	4

*These values are only due to rounding.



Notes: In the table on the left values depict summary statistics of the distribution of pairwise correlations within each group. The median can be found on the diagonal of the schematic figure on the right which moreover displays the median of all cross-group correlations. Details on the variable components of each group are listed in Table 3.9.

in general, which can be seen in Figure 3.1. There, the median correlation across groups is shown, which the diagonal indicating the median correlation in-group, corresponding to column 50% in the Table on the left of Figure 3.1.

The recovery rate is defined as the mean of trading price between the default day and data 30 days after default, which we retrieve from Capital IQ. The data is originally from the Trade Reporting and Compliance Engine (TRACE). In our analysis, all corporate bonds have debt values, at the time of default, of greater than $50 million. The mean value of the recovery rate for the 2,079 U.S. corporate bonds in our sample is 45.57 percent, and the sample standard deviation is 35.04 percent. The empirical distribution of the recovery rates of defaulted US corporate bonds naturally peaked in the financial crisis from 2008-2010[3]. Around 30 percent of defaulted bonds have recovery rate less

[3]A more detailed figure on the distribution of defaults over time can be found in Figure 3.4 in Appendix 3.5.2. A large share of defaults was caused by both the Lehman Brother bankruptcy in September 2008, and the CIT Group Inc. bankruptcy in November 2009.

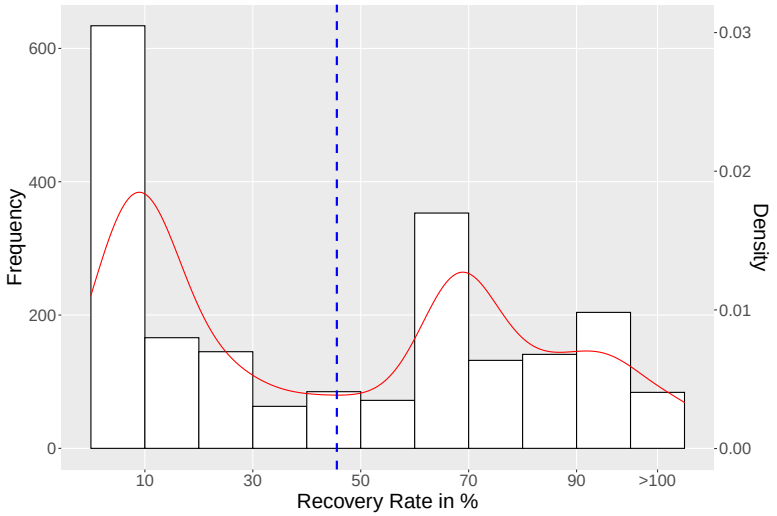


Figure 3.1: Recovery Rate Frequency and Density (Red) for the Defaulted US Corporate Bonds From 2001 to 2016

Notes: Mean recovery rate is depicted in dashed blue lines. The rightmost class is truncated in the plot for reasons of clarity.

than 10 percent. There is another distribution peak in the range of values between 60 percent and 70 percent, which is visualized in Figure 3.1.

In addition, our bonds consist of four seniority levels for the bonds: (i) senior secured, (ii) senior unsecured, (iii) senior subordinated, (iv) subordinated, and junior subordinated. An overview over the distributions over the different bond types can be found in Appendix 3.5.2 in Figure 3.5. Since most defaults (82.5%) occurred in the class of senior unsecured bonds, which is driving the distribution of recovery rates, we decided to not distinguish between groups of seniority levels in our analysis. Additionally, the sample size would be too small for such a sub-analysis.

3.3.2 Empirical Results

In this subsection, we use our methodology to identify and quantify the recovery rates of corporate bonds in a data-driven way. This is key in practice for investments, hedging, and supervision but also for model building and interpretation. As an extension in a comprehensive out-of-sample forecasting study, we also demonstrate that simple linear

predictions based on variables selected by the knockoff procedures can compete with and often improve upon nonparametric methods that employ the full set of variables. Moreover, we show that the obtained knock-off selection of variables is robust to discarding certain time subperiods, e.g. after the financial crisis.

Identification of Important Groups and Effects on Recovery Rates

To determine the driving factors of corporate bond recovery rates, we use our proposed methodology on the full sample from 2001 to 2016 for the data-driven selection of relevant components. We show results of our suggested combined subsampling and weighted FDR selection technique (see Method 2 and 3) across all types of different baseline-knockoff methods, i.e. in particular, we study model-X knockoffs, deep knockoffs with two different neural network architectures, and group knockoffs.

Since our data is highly correlated within groups (see Table 3.1) and we are mainly interested in group effects and selections, we transform our data using principal component analysis (PCA)[4]. To retain interpretability on a group level, we conduct one PCA per group and only use the most important principal components (PC) to describe that specific group, i.e. a maximum of four PCs that explain at least 90% of the variability in the group. This helps to reduce high correlations among variables and break down large groups of variables to one or two components to see their main effects, avoiding multicollinearity issues in post-selection linear models. Additionally as a robustness check for the model-X knockoffs, we employ a smaller version using a maximum of two PCs ("2comp").This group-PCA step greatly reduces the dimensionality of the data and serves as a viable alternative to other pre-screening procedures such as omitting variables with high pairwise correlations. With that, the variables are scaled by their standard deviation and centered around zero. Similar approaches in reducing dimensions have also been taken by Kelly et al. (2019) in an asset pricing application.

Figure 3.2 graphically shows the most important PCA-features for model-X knockoffs over all possible nominal FDRs, while similar figures for the other procedures can be found in Appendix 3.5.2 (Figure 3.6). Most prominently, the selection probabilities of important features are rather high (with selection frequency of 0.6) already at a nominal

[4]See e.g. Hastie et al. (2009, Chap. 14.5).

$FDR = 0.2$ and rise to 0.8 at nominal $FDR = 0.4$ for the model-X procedures, where other, less important factors only attain similar levels from a nominal $FDR = 0.8$ onward. Such levels of FDR clearly undesirable, but investigating the entire grid of FDR-levels jointly and with appropriate weighting is beneficial and yields robustness due to the rather high variability of selection probabilities for minimal changes at a considered specific nominal FDR-level. For the deep knockoff procedures, however, we see high selection probabilities for relevant features throughout all FDR levels (see Figure 3.6 (bottom)). Comparing different structures for the neural networks in the deep knockoffs, this effect is more pronounced for narrower networks with only 5 neurons per layer. Since a wider network can learn more complex structures, it can build more accurate knockoffs and with that, identify variables that are less likely to be true influencing variables, especially for cases where nominal FDR-levels are low. This highlights the importance of considering multiple methods for generating knockoffs and combining their insights to identify the most important groups.

To finally select appropriate variables over all five methods and FDRs, we compare two different ranking procedures for variables at each FDR-level. To combine the different rankings from each FDR-level to a final selection (probability), we employ three distinct weighting approaches. First, we distinguish between using ranking of variable selections (from 20 to one) or selection probabilities from our procedures. The subsequent weighting of ranks/probabilities for each variable over all nominal FDR-values is conducted as in Section 3.2 using either equal weighs, linear decaying weights (ω_k^{lin}, *Lin-decreasing*), or exponentially decaying weights (ω_k^{exp}, *Log-decreasing*). The weights assign the highest weights to low FDR-values and decrease with higher FDR-levels (see Section 3.2 for details). Figure 3.3 shows boxplots over selections from different methods by both ranking procedures and the three weighting schemes. In general, we can see that group 14, 11, and 12 are always among the top four in each procedure, while group 5 only appears as important when looking at probabilities, and group 20 vice versa only when looking at ranks. Table 3.2 highlights the influence selection probabilities and ranks, where both group 5 and group 20 are given higher relative importance in the weighted schemes that focus on low nominal FDRs. Otherwise, results for the most selected groups are mostly stable over both schemes and all weights. This is largely in line with the literature that

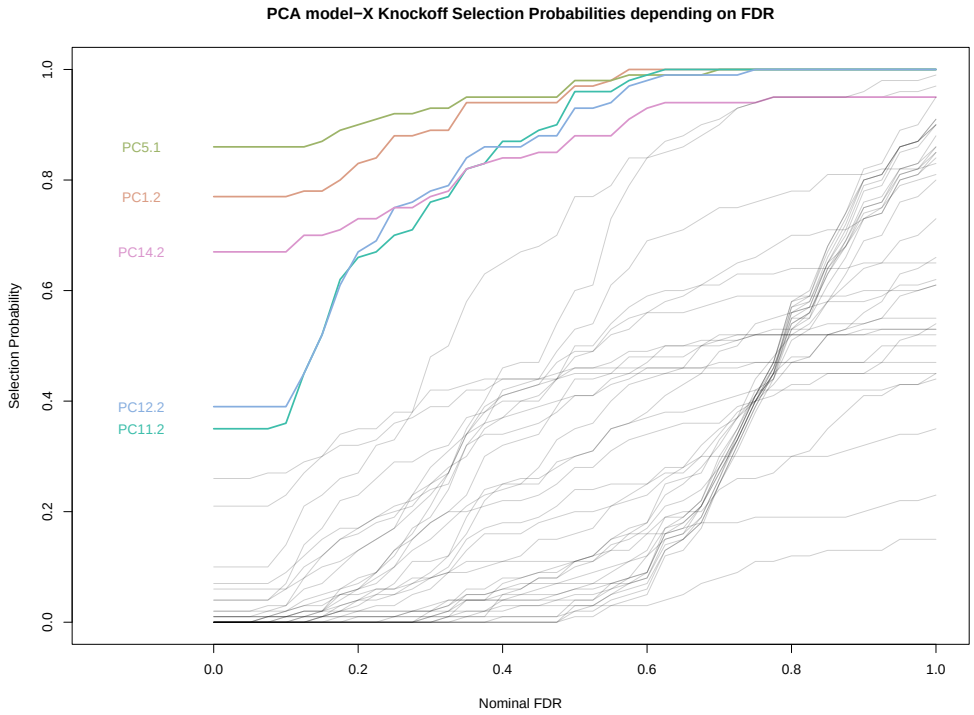


Figure 3.2: Model-X Knockoff Selection Probabilities for Different Nominal FDR Using Group Principal Components

Notes: Selection probabilities are obtained rerunning the full knockoff procedures using repeated subsampling of 90% of the data (100 iterations). Highlighted groups have the highest mean selection rank, i.e. the mean over the rank in each FDR-scenario. The PCA component with the highest probability receives the highest rank (= 41) and vice versa (= 1).

also determines the factors in group 14, 11, 20, and 5 as relevant with a more simplistic and less robust data-driven selection technology (Jankowitsch et al., 2014; Nazemi et al., 2018)[5]. Though different from the existing studies, however, our methods additionally also detect group 12 as important, that consists of exchange rates.

Note that group 14 describes bond yields of major bonds and rates of different general indicators such as mortgage, treasury, and loans, which might be considered naturally

[5]Other, less important groups that have been selected often by our approach include high yield default rates, defaults in the respective industry, and GDP measurements. These findings are also in line with Nazemi et al. (2018) and Jankowitsch et al. (2014), who report similar factors to be important.

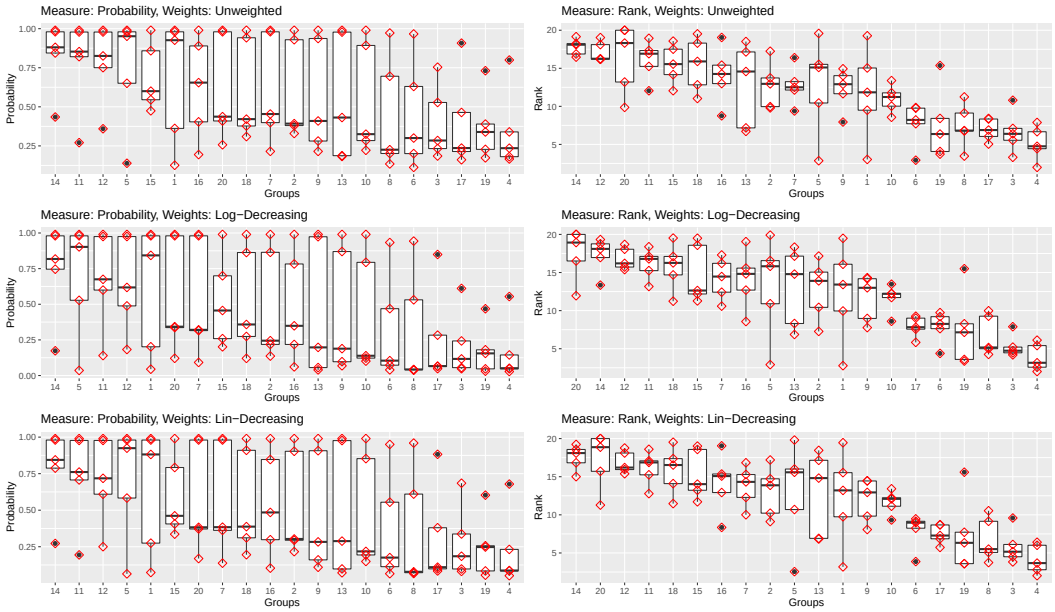


Figure 3.3: Boxplots of Weighted Mean Selection Probabilities/Ranks of Each Group Using Different Procedures

Notes: In each subplot, the red squares represent the means of each procedure. Groups are sorted by highest probability/rank from left to right. For PCA procedures, the group probability/rank is assigned as the highest value over all group-principal components. In case of ranks, the PCA-ranks are rescaled linearly to lie between 1 and 20.

predictive for the state of the economy and thus of bonds. The data-driven selection therefore confirms the intuition that these indicators have an influence on recovery rates. Similarly, the selection of the other groups can be explained. These describe international exchange rates against the USD (group 12) and stock market indicators such as returns and volatilities of the most important indices (group 11). Furthermore, capacity utilization of industries and change in inventories of private and businesses play an important role (group 20), as well as corporate measures such as profit of the firm and cash flow (group 5). More specifically, in the light of the financial crisis in 2008 and the following euro crisis, exchange rates were strongly affected (see e.g. McCauley and McGuire, 2009; Kohler, 2010), which might explain their connection to recovery rates in regard to globally active firms. The capacity utilization group, on the other hand, measures how much of total potential output is actually utilized by industry and

Table 3.2: Most-Selected Groups Over Different Weighting Schemes

Rank:	1		2		3		4	
	Group	Mean Score	Group	Mean Score	Group	Mean Score	Group	Mean Score
Prob_unweight	14	0.826	11	0.783	12	0.781	5	0.742
Prob_exp	14	0.741	5	0.687	11	0.676	12	0.651
Prob_lin	14	0.775	11	0.726	12	0.709	5	0.708
Rank_unweight	14	17.768	12	17.156	20	16.276	11	16.100
Rank_exp	20	17.479	14	17.300	12	16.808	11	16.139
Rank_lin	14	17.547	20	17.172	12	16.880	11	16.117

Notes: In the columns, *Group* depicts the selected variable group, while *Mean Score* shows the (weighted) mean over all five procedures for the four most selected groups. *Prob* and *Rank* refer to whether probabilities or ranks are used, while *unweight*, *exp*, and *lin* refer to the weighting scheme of equal weighting, linear-decreasing weighting, and exponentially decreasing weighting. The ranks for groups are rescaled linearly to lie between 1 and 20 (20 being the best score), while the selection probabilities lie between 0 and 1.

additionally contains information about inventories (and their change over time). This is highly relevant for recovery rates when thinking of a firm's business model in general and inventories of firms that could indicate how much can be recovered given default. Naturally, returns and volatilities of the most important indices such as the S&P 500 or the NASDAQ 100 describe the general situation of the economy and the value of companies, which again is an indicator of recovery rates of these firms. Finally, profit and cash flow are probably the most direct factors for short-term companies finances, and are thus a good predictor for the default of a firm.

As a robustness check, we also computed variable importance measures from a random forest model using mean variance reduction as a measure for the importance of a group[6]. Computational details of this model can be found in Appendix 3.5.1. Table 3.10 in Appendix 3.5.2 shows the mean values aggregated on a group level of variance reduction and corresponding p-values from the PIMP procedure of Altmann et al. (2010). There, we can confirm that especially group 12, 14, and also 11 have high importance, while group 20 and group 5 are not deemed important. This can be explained by the predictive nature of the measure and method that favors groups such as group 15, which measures the bond defaults within the industries. Interestingly, group 15 is also in the top 6 groups

[6]To give the nonparametric random forest maximum flexibility, we used the the raw data as input instead of using the aggregated PCA-groups.

of many of the other procedures, although mostly ranked below all the other selected groups.

Post-Selection Performance

To obtain an unbiased quantification of the effect of each factor, we re-estimate a linear model using only those groups that were selected in the first part of Section 3.3.2. We show the effect of the most important principal components (PCs) for each group in Table 3.3. While the effects of selected variables appear mostly significant, it has to be noted that assuming a linear model might be too optimistic and effects between groups might be affected by some remaining multicollinearity between similar groups.

Using the PCs allows using and working with the strong correlation within groups, and facilitates the interpretation and comparison of effects between different groups since variables in each group are centered and scaled. The results of Table 3.3 show that most of the selected coefficients seem highly significant, where focus should lie on the first and second PC which capture the largest share of the variance in each group. Group 14 has a largely negative impact on recovery rates, although especially the last PC is affected by inclusion of more variables switching signs. Group 11 has a primarily positive impact (in the first two PCs), while adding large explanatory power (Adjusted R^2), It is, however, also correlated with group 12, which adds little explanatory power, but also has a significant positive first PC. Group 20 and 5 have a negative impact but appear to affect the coefficients of other components, which is why we consider their inclusion rather as a robustness check, since they also do not add as much to an increase in R^2 as for example group 14 and 11. Group 12 is a special case, having both positive (PC1) and negative (PC2) significant coefficients. Taking a closer look at the weights of the PCs[7], the first PC assigns a large negative weight to the exchange rates of Canadian dollar, Swiss franc against one USD, and the real broad effective exchange rate for the US, while the second PC gives large negative weight to the rate of USD against one British pound.

It is in line with intuition that higher bond yields and rates such as mortgages have a negative impact on recovery rates (group 14), since they might indicate a riskier environment. The positive impact of group 11 can be attributed to the fact that higher

[7] A complete list of the PCA-weights can be found in Table 3.12 in the Appendix.

Table 3.3: Linear Regression with PCA-Components of Most-Selected Groups

	Most Selected Groups			Additional Groups	
	Group 14	Group 14,11	Group 14,11,12	Group 14,11,12,20	Group 14,11,12,20,5
PC5.1					−14.225*** (2.456)
PC5.2					1.751 (4.604)
PC11.1		1.895*** (0.544)	−0.973 (0.646)	−0.922 (0.645)	0.467 (0.651)
PC11.2		9.386*** (0.594)	8.524*** (0.675)	8.782*** (0.784)	8.841*** (0.717)
PC11.3		−4.464*** (1.150)	−1.316 (1.242)	0.285 (1.474)	−1.337 (1.530)
PC11.4		−8.865*** (1.426)	−1.936 (1.737)	0.232 (2.205)	−1.551 (2.274)
PC12.1			3.953*** (0.706)	2.925*** (0.738)	4.402*** (0.748)
PC12.2			−6.158*** (1.387)	−7.384*** (1.663)	−7.530*** (2.035)
PC14.1	−1.346*** (0.205)	−3.428*** (0.309)	−1.826*** (0.499)	−2.032*** (0.562)	−0.694 (0.595)
PC14.2	3.254*** (0.363)	3.312*** (0.605)	3.888*** (0.636)	4.201*** (0.995)	8.411*** (1.062)
PC14.3	−1.352** (0.567)	0.238 (0.992)	−3.044*** (1.016)	−2.684* (1.467)	−2.816* (1.501)
PC20.1				1.373 (1.181)	−0.643 (1.991)
PC20.2				−5.211*** (1.769)	−18.210*** (4.366)
Constant	45.574*** (0.748)	45.574*** (0.691)	45.574*** (0.682)	45.574*** (0.681)	45.574*** (0.672)
Observations	2,079	2,079	2,079	2,079	2,079
R^2	0.056	0.170	0.188	0.223	0.244
Adjusted R^2	0.055	0.168	0.185	0.219	0.239
Residual Std. Error	34.070 (df = 2075)	31.958 (df = 2073)	31.637 (df = 2071)	30.964 (df = 2067)	30.574 (df = 2065)
F Statistic	41.139*** (df = 3; 2075)	85.117*** (df = 5; 2073)	68.351*** (df = 7; 2071)	54.053*** (df = 11; 2067)	51.147*** (df = 13; 2065)

Notes: PCX.Y stands for principal component Y of group X. Variables were selected taking all groups among the four most-selected groups over all weighting schemes. See Table 3.2 for details on group selection. We include PCs in each group until they explain more than 90% of total variability in that group. Coefficients are shown with stars according to their significance in t-tests. SEs in parentheses are HC3 robust. *p<0.1; **p<0.05; ***p<0.01.

returns and smaller volatilities in the stock indices result in larger recovery rates. At the same time, the positive impact of exchange rates (group 12) in the first PC indicates that when the USD is weak against other major currencies, recovery rates are higher, while the opposite effect is observed in PC2. This effect could be explained by defaulted companies holding assets in foreign currencies that are more valuable when the USD is weak. On the other hand, we cannot fully rule out that this effect is caused by the USD exchange rate dropping against other major currencies (see e.g. Kohler, 2010) because of large crash events that are connected to bond defaulting.

Extension: Out-of-Sample Prediction Performance

In addition to the identification and interpretation of important factors explaining recovery rates, we also assess the out-of-sample forecasting performance of the reduced models in various scenarios. Here, we distinguish between two main cases: firstly, we check the

infeasible forecasting scenario as reference point, where we use the determined models from the first part of Section 3.3.2 employing information from the full data comprising 2001-2016 in the model selection step when forecasting for the year 2012-2016 (Table 3.4). Additionally, we also provide results for the "completely" out-of-sample forecasting case , where we re-determine all models on a limited time period from 2001-2011 and predict 2012-2016 (see Table 3.5). For the post-selection estimation step, we use a wide variety of models ranging from simple linear methods, standard and penalized, up to flexible fully nonparametric methods such as random forests (see Appendix 3.5.1 for implementation details). Moreover, we consider both cases with the full raw data and group-PCA-transformed data in different settings. In all settings, we clearly confirm that knockoff pre-selection improves prediction performance. The results also highlight that this cannot be achieved with lasso pre-selection, thus confirming the importance of our robust approach. After knockoff pre-selection, simple (penalized) linear forecasting models often achieve quite competitive forecasting performance with only slight improvements by a non-linear fit. This highlights that forecasting with a data-driven selection of important predictors pays-off, while maintaining easy interpretation in comparison to their fully nonparametric counterparts.

We assess the forecasting performance by calculating the root-mean-squared forecasting error $RMSE = \sqrt{\frac{1}{K}\sum_{\tau=k}^{T}(\hat{y}_\tau - y_\tau)^2}$ and the mean-absolute error $MAE = \frac{1}{K}\sum_{\tau=k}^{T}|\hat{y}_t - y_\tau|$ for a prediction $\hat{y}_\tau$ of y_τ at forecasting time $\tau = k, \ldots, T$, and forecast length $K = T-k$. We use different forecast constructions with fixed, expanding and rolling windows on annual and daily horizons. For the fixed window type, we set the training data to 2001-2011 and provide daily predictions for 2012-2016. In the expanding window case, we use data from 2001 up to a certain year τ in the set $\{2011, 2012, 2013, 2014\}$ and predict daily values in $\tau + l$ where $l \in \{1, 2\}$[8]. For daily rolling windows, we set the training length to 10 years as in the initial expanding case and the fixed window setting and predict one corporate default observation ahead (Daily)[9] We estimate either cross-validated

[8] We use an expanding window here to account for the difficulty of predicting two full years at once.

[9] This does not necessarily mean that this is one day ahead ahead, as some defaults occurred on the same day. We chose to always jump to the next day containing a default to maintain a realistic time structure in that scenario (see also Nazemi et al. (2022)).

elastic nets (mixing parameter $\alpha = 0.5$)[10], cross-validated lasso regressions, or simple linear models, and use random forests as nonparametric benchmarks. For each window construction, we employ the post-selection methods either with the full raw data or the group-PCA transformed data (as described in the first part of Section 3.3.2, we use as many PCs to explain 90% of the variance in each group, see also Table 3.11). We either use the above data without pre-selection or employ the pre-selected set of variables according to the different setups in the first part of Section 3.3.2. This comprises using our proposed weighted FDR selection (wFDR)[11] combining all baseline knockoff procedures or using only our repeated-subsampling procedure (see Procedure 2 in Section 3.2) in combination with the baseline-methods. These are either model-X knockoffs (MX, or MX 2 Comp. using a maximum of 2 PCs) or deep knockoffs with 5 (Narrow) and 25 (Wide) neurons per layer.

For the infeasible reference scenario in Table 3.4 and the group-PCA-transformed data, we use PCs that are estimated over the full data set. In the "completely out-of-sample" forecasting case in Table 3.5, the out-of-sample PCs for time points after 2011 are created using the weights from the PCs with only data up to the end of 2011[12].

Table 3.4 shows that generally simple linear models with limited pre-selected variables from our proposed weighted FDR procedure that combine different knockoff selections works best for forecasting both longer and shorter time horizons. Moreover, as a single selection techniques, also, the model-X procedure within our robustified framework yields excellent results with a simple linear post-selection fit. Determining variables with the Deep Knockoff robustified framework generally performs slightly worse also for non-linear post-selection models, with the relative best performance for shorter forecasting horizons. This is not unexpected since for the deep knockoffs, selection probabilities were generally much higher, meaning they could contain more noise variables that would bias predictions for longer time horizons (i.e. they do not generalize as well as the weighted

[10] More specifically, the penalty in the objective function is specified as $\sum_{j=1}^{p}(\alpha|b_j| + (1-\alpha)b_j^2)$

[11] For the wFDR, we use all PCA-components from the three most-selected groups, i.e. 14,12,11. See also Table 3.3 for comparison.

[12] For comparability, scaling/centering uses information of the entire sample. But scaling with weights from data up to the end of 2011 does not substantially change the prediction performance. Results are available from authors upon request.

Table 3.4: Out-Of-Sample Predictions (Theoretical Infeasible Case: Model Selection and Principal Component Construction Based on the Entire Sample))

Group-PCA	Selection Method	Post-Selection	Fixed		Annual		Daily	
			RMSE	MAE	RMSE	MAE	RMSE	MAE
✓	wFDR Knock.	Elastic Net	**28.74**	23.41	30.21	24.92	**29.14**	24.33
✓	wFDR Knock.	OLS	**28.50**	22.30	30.05	24.40	**29.23**	24.41
✓	MX Knock.	Elastic Net	28.85	**21.71**	**30.01**	**23.61**	29.26	**23.44**
✓	MX Knock.	OLS	28.93	**21.63**	**30.02**	**23.52**	29.33	**23.29**
✓	MX Knock.	Random Forest	30.87	26.87	33.19	28.51	30.87	25.80
✓	MX Knock. 2 Comp.	Elastic Net	29.86	24.78	31.50	26.63	30.09	25.13
✓	MX Knock. 2 Comp.	OLS	29.79	24.67	31.02	25.89	30.00	24.98
✓	Deep Knock. Narrow	Elastic Net	44.52	40.07	36.61	32.59	33.07	28.89
✓	Deep Knock. Narrow	OLS	43.88	39.46	36.50	32.49	33.03	28.83
✓	Deep Knock. Wide	Elastic Net	37.17	32.47	35.20	29.86	32.87	28.13
✓	Deep Knock. Wide	OLS	36.78	32.08	34.64	29.27	32.87	28.16
✓	No Selection	Elastic Net	183.94	162.04	50.25	37.34	30.87	23.89
✓	No Selection	Lasso	192.16	169.71	52.67	38.49	31.07	24.11
✓	No Selection	Random Forest	35.52	31.29	34.29	29.82	29.53	23.54
	Group Knock.	Elastic Net	223.40	194.16	68.30	51.22	31.48	24.01
	No Selection	Elastic Net	268.10	237.02	79.04	58.49	34.18	25.47
	No Selection	Lasso	240.53	213.98	74.75	56.47	35.01	26.18
	No Selection	Random Forest	36.23	32.03	33.17	28.99	29.42	23.55

Notes: The table shows the predictive performance after different pre-selection or no pre-selection occurred, for PCA or pure data components and across different post-selection methods. In each forecasting scheme, the best two models are marked in bold.

FDR counterparts). The baseline linear models using the full raw data perform poorly for large time horizons, especially for fixed windows, which can be explained by potential overfitting on noisy data. Interestingly, this cannot be fully countered by regularization using elastic nets for forecasting tasks. Only for very short time horizons as the daily rolling window, the baseline procedures can compete. These findings are highly in favor of using our proposed statistical model selection techniques also for forecasting tasks. The machine learning (ML) benchmarks with selection on the full raw data perform similarly, but always slightly worse than the knockoff counterparts with the generally downside of lacking transparency and interpretability of the influence of certain groups and factors. Using the PCs instead of raw data only significantly improves the random forest model

Table 3.5: (Completely) Out-Of-Sample Predictions (Practically Feasible Case): Model Selection and Principal Component Construction Based Only on Period up to 2012

Group-PCA	Selection Method	Post-Selection	Fixed		Annual		Daily	
			RMSE	MAE	RMSE	MAE	RMSE	MAE
✓	wFDR Knock.	Elastic Net	31.38	27.26	**31.95**	27.52	31.28	26.44
✓	wFDR Knock.	OLS	**31.12**	26.98	32.01	27.60	31.39	26.49
✓	MX Knock.	Elastic Net	38.67	32.68	35.18	29.61	34.80	29.43
✓	MX Knock.	OLS	38.88	32.83	35.29	29.70	34.89	29.49
✓	MX Knock.	Random Forest	31.17	**25.37**	33.15	**26.78**	29.59	23.77
✓	MX Knock. 2 Comp.	Elastic Net	38.62	32.64	35.18	29.61	34.72	29.35
✓	MX Knock. 2 Comp.	OLS	38.88	32.83	35.29	29.70	34.89	29.49
✓	No Selection	Elastic Net	107.07	91.70	62.32	49.33	43.14	32.00
✓	No Selection	Lasso	94.93	80.22	57.54	46.40	41.31	31.74
✓	No Selection	Random Forest	**30.80**	**26.45**	**31.77**	**27.08**	**28.84**	**23.13**
	Group Knock.	Elastic Net	223.40	194.16	68.30	51.22	31.48	24.01
	No Selection	Elastic Net	268.10	237.02	79.04	58.49	34.18	25.47
	No Selection	Lasso	240.53	213.98	70.55	50.16	34.83	25.81
	No Selection	Random Forest	36.23	32.03	33.17	28.99	**29.42**	**23.55**

Notes: In contrast to Table 3.4, variable selection and PCA is only performed with data up to the end of 2011. The best two models for each forecasting scheme are again marked in bold.

for the fixed window. Note that for this study, we used the standard recommended data-driven choice of tuning parameters for the machine learning benchmarks but did not additionally fine-tune from there in order to maintain comparability between the simple baselines, the knockoff procedures, and the benchmark models.[13].

In the "completely" out-of-sample scenario, we repeat the model selection step from Section 3.3.2, but only use data up to the end of 2011 to determine the relevant variables with our proposed methodology for all different knockoff-baseline procedures. This represents the most realistic but also most challenging scenario for the knockoff procedure, where post-crisis recovery rates are not contained in the training set but must be predicted. The resulting stability in selections and in forecasting performance therefore indicates

[13]While tuning all hyperparameters cautiously could improve ML-forecasts to some extent, previous studies for recovery rates show that expected changes are minor (see e.g. Nazemi et al. (2022) with additional news-based variables).

that our choices are important also for non-crisis periods. Table 3.5 summarizes the new results[14]. As expected, our proposed methodology ("wFDR") is on par with the top-performing machine learning methods in this case as well, although the random forest with the full raw data is slightly better for the daily window. In comparison to the infeasible full-sample selection results in Table 3.4, the single model-X knockoffs perform slightly worse, with less variability between the different knockoffs employed within the subsampling knockoff framework. This can be explained by the smaller data set, where fewer variables are selected in general and difference between the different model-X knockoffs is smaller. Selections in general are the same compared to the full data case for our proposed weighted FDR procedure, and very similar for the single "baseline" methods, with only a few minor changes (see Table 3.8 in Appendix 3.5.2 for details). In terms of forecasting performance, the ranks for the different procedures are stable, meaning that our proposed weighted FDR methodology together with the random forest are still performing best, while using no or no robust selection still performs worst overall. The margin between the latter and our procedure is smaller only for the daily rolling window and the MAE, where very bad predictions (e.g. in cases of large default events) are not punished as heavily as with the MSE. Using the usual MSE-measure, the raw data with an elastic net (or lasso) still perform considerably worse.

As an additional robustness check for the predictive performance of our methods, we compute model confidence sets (Hansen et al., 2011) for the same forecasting combinations as before and for both the infeasible scenario and the "pure forecasting case". As suggested in Hansen et al. (2011), we use $B = 5000$ bootstrap replications, the $TMax$ test-statistic, and test-level $\alpha = 0.15$. Please see Appendix 3.5.1 for details. The displayed results are robust across different α test levels in the standard range $[0.1; 0.2]$. [15] Although the model confidence sets differ over the various forecasting schemes, we can identify models that consistently fall into the model confidence set. Please see Table 3.6 and Table 3.7 for full-sample selection and completely out-of-sample forecast results, respectively. We want to highlight that our proposed wFDR procedure always belongs to the model confidence set, together with the random forest procedure in the full-sample selection

[14] We did not include the Deep Knockoff procedures here since the sample size is significantly reduced for selections.

[15] Results are omitted here but are available upon request from the authors.

Table 3.6: Model Confidence Sets for $\alpha = 0.15$ and Different Methods With Full-Sample Selection and Principal Components

			Fixed		Annual		Daily	
Group-PCA	Selection Method	Post-Selection	TMax	P-Value	TMax	P-Value	TMax	P-Value
Selected Into All Model Confidence Sets								
✓	wFDR Knock.	Elastic Net	**−5.11**	1.00	-0.27	1.00	-2.53	1.00
✓	wFDR Knock.	OLS	-1.36	1.00	-0.74	1.00	-2.36	1.00
✓	MX Knock.	Elastic Net	-0.78	1.00	**−0.99**	1.00	**−2.95**	1.00
✓	MX Knock.	OLS	-0.61	1.00	-0.86	1.00	-2.94	1.00
✓	MX Knock. 2 Comp.	OLS	0.88	0.73	1.70	0.17	-1.47	1.00
Selected Into Two Model Confidence Sets								
✓	MX Knock.	Random Forest	1.22	0.51	-	-	-0.43	1.00
✓	MX Knock. 2 Comp.	Elastic Net	0.95	0.68	-	-	-1.32	1.00
Selected Into One Model Confidence Set								
✓	Deep Knock. Narrow	Elastic Net	-	-	-	-	0.97	0.83
✓	Deep Knock. Narrow	OLS	-	-	-	-	0.94	0.85
✓	Deep Knock. Wide	Elastic Net	-	-	-	-	0.91	0.87
✓	Deep Knock. Wide	OLS	-	-	-	-	0.90	0.87
✓	No Selection	Elastic Net	-	-	-	-	-0.31	1.00
✓	No Selection	Lasso	-	-	-	-	-0.14	1.00
✓	No Selection	Random Forest	-	-	-	-	-2.12	1.00
	Group Knock.	Elastic Net	-	-	-	-	0.19	1.00
	No Selection	Elastic Net	-	-	-	-	1.04	0.80
	No Selection	Lasso	-	-	-	-	1.24	0.66
	No Selection	Random Forest	-	-	-	-	-2.37	1.00
Selected Into No Model Confidence Set								
	Group Knock.	OLS	-	-	-	-	-	-

Notes: "TMax" depicts the test statistic and "P-Value" depicts the p-value for the TMax procedure in testing equal predictive ability in the model confidence procedure of Hansen et al. (2011) implemented with the R-package from Bernardi and Catania (2018). The TMax-values depict the average loss difference of that method compared to all other methods (negative value indicating smaller loss). P-values are based on $B = 5000$ bootstrap samples with rejection level set to $\alpha = 0.15$. Losses are squared losses calculated as in Table 3.4 and other details follow this table. The best model in each confidence set is marked in bold. "-" indicates that the model was not selected, i.e. had p-value smaller than α.

Table 3.7: Model Confidence Sets for $\alpha = 0.15$ and Different Methods for the Completely Out-Of-Sample Period With Selections up to 2012

			Fixed		Annual		Daily	
Group-PCA	Selection Method	Post-Selection	TMax	P-Value	TMax	P-Value	TMax	P-Value
Selected Into All Model Confidence Sets								
✓	wFDR Knock.	Elastic Net	0.40	0.81	-1.71	1.00	-1.74	1.00
✓	wFDR Knock.	OLS	0.01	1.00	-1.60	1.00	-1.66	1.00
✓	MX Knock.	Random Forest	0.06	1.00	-0.35	1.00	-3.32	1.00
✓	No Selection	Random Forest	**−0.61**	1.00	**−1.81**	1.00	**−4.04**	1.00
Selected Into Two Model Confidence Sets								
✓	MX Knock.	Elastic Net	-	-	1.43	0.34	0.42	0.99
✓	MX Knock.	OLS	-	-	1.44	0.33	0.48	0.98
✓	MX Knock. 2 Comp.	Elastic Net	-	-	1.43	0.34	0.37	0.99
✓	MX Knock. 2 Comp.	OLS	-	-	1.44	0.33	0.48	0.98
	No Selection	Random Forest	-	-	-0.52	1.00	-3.32	1.00
Selected Into One Model Confidence Set								
✓	No Selection	Elastic Net	-	-	-	-	1.51	0.48
✓	No Selection	Lasso	-	-	-	-	1.62	0.40
	Group Knock.	Elastic Net	-	-	-	-	-1.77	1.00
	No Selection	Elastic Net	-	-	-	-	0.01	1.00
	No Selection	Lasso	-	-	-	-	0.24	1.00
Selected Into No Model Confidence Set								
	Group Knock.	OLS	-	-	-	-	-	-

Notes: Compare Table 3.6 for detailed descriptions. Losses are squared losses calculated as in Table 3.5 and other details follow this table.

scenario. Over the full sample selection, the other model-X procedures can still compete, while in the "completely" out-of-sample case, they are outperformed by our proposed wFDR. This is also confirmed by the corresponding test statistics, where a lower value indicates better performance. More specifically, a negative value of the test statistic indicates that the average loss is smaller compared to all other methods in the confidence set, where our suggested method clearly outperforms the other procedures in general for the full-sample selection, while for the "completely" out-of-sample scenario, the raw random forest is slightly better when comparing the test-statistic. Using no or non-robust selection methods (i.e. lasso, elastic net) is always worse, and also the plain group knockoff does not perform well on our data set, which might be caused by our specific data structure, where we still have some strong correlations between groups.

3.4 Conclusion

In this paper, we demonstrate the benefit of connecting the flexibility of the knockoff framework with repeated subsampling and techniques controlling the proportion of false discoveries over the full spectrum of possible values. We are able to uncover important macroeconomic factors of corporate bond recovery rates while maintaining excellent forecasting performance. We employ a comprehensive set of distinctive knockoff machines, show that a transparent combination of their results yields optimal ensemble results, and consider the full grid of possible uncertainties in the methodology, which leads to a more stable selection.

With that, we identify important groups of variables and show their effect on recovery rates. Predictive power in various settings using linear models with just the identified groups is significantly higher than using the full set of variables in similar models. Furthermore, our procedure outperforms other model selection procedures (Sparse-step, stability selection, MC+[16] and performs similar to flexible machine learning methods. The latter are developed for prediction tasks but lack easy interpretation and identification of important factors, which is provided employing the proposed methodology.

For future research, the proposed methodology shows high-potential in other data-rich empirical finance environments such as e.g. asset pricing. In a separate paper, it would be of interest to derive conditions for FDR-level components and optimized forms of parsimonious weighting schemes to theoretically achieve and derive optimal in-sample or out-of-sample fits and respective statistical rates.

[16] See e.g. Nazemi et al. (2022).

3.5 Appendix

3.5.1 Methods

Deep Knockoffs

Even though the procedure of Candès et al. (2018) poses only few assumptions on Y and X together, namely that the observations are identically and independently distributed, there is the assumption that the distribution F_X of X is known beforehand. This procedure can be ineffective in producing reliable knockoff variables when the covariance structure of X is hard to replicate, e.g. when variables in X are highly correlated. This problem arises due to often conflicting requirements of X and $\tilde{X}$ to have a similar covariance structure but to be uncorrelated at the same time. Romano et al. (2020) propose to solve these issues by replacing the model-X algorithm by an artificial neural network. They define the covariance matrix of the combined $(X, \tilde{X})$ G and t:

$$G = Cov[(X, \tilde{X})] = \begin{bmatrix} G_{XX} & G_{X\tilde{X}} \\ G_{\tilde{X}X} & G_{\tilde{X}\tilde{X}} \end{bmatrix} . \tag{3.7}$$

We apply this approach to improve the creation of knockoffs and to provide a robustness check of the model-X knockoffs. In contrast to the model-X construction, a deep neural network is used to generate knockoff variables. We use different structures of neural networks and compare their performance. In the final analysis, we include one wide network (25 neurons per layer) and one narrow network (5 neurons per layer), each of them with six layers, and otherwise the same structure as in the implementation of Romano et al. (2020)[17]. The advantage of the Deep Knockoff procedure lies in the creation of the knockoffs. Since we use neural networks, we can model more complex relations and control specifically for higher moments in the knockoff distribution as well as the correlation of X and $\tilde{X}$.

Given X and a random noise matrix V, the network outputs knockoff copies $\tilde{X}$ that are then evaluated using a customized loss function. Also define X', $X'' \in \mathbb{R}^{n/2 \times p}$ as a random partition of X. This loss function J can be defined as follows (see Romano et al. (2020) for details) given M as a $p \times p$-matrix with zero diagonal and ones everywhere

[17] See `https://github.com/msesia/deepknockoffs` for details.

else, $\circ$ as element-wise multiplication of two matrices:

$$
\begin{aligned}
J_{\gamma,\lambda,\delta}(X,\tilde{X}) &= \gamma J_{MMD}(X,\tilde{X}) + \lambda J_{second-order}(X,\tilde{X}) + \delta J_{decorrelation}(X,\tilde{X}) \\
J_{MMD}(X,\tilde{X}) &= \hat{D}_{MMD}\left[(X',\tilde{X}'),(\tilde{X}'',X'')\right] + \hat{D}_{MMD}\left[(X',\tilde{X}'),(X'',\tilde{X}'')_{swap(S)}\right] \\
J_{second-order}(X,\tilde{X}) &= \lambda_1 \frac{\|G_{XX} - G_{\tilde{X}\tilde{X}}\|_2^2}{\|G_{XX}\|_2^2} + \lambda_2 \frac{\|M \circ (G_{XX} - G_{\tilde{X}\tilde{X}})\|_2^2}{\|G_{XX}\|_2^2} + \\
&\quad \frac{\lambda_3}{p} \left\| \frac{\sum_{i=1}^n (X_i - \tilde{X}_i)}{n} \right\|_2^2 \\
J_{decorrelation-order}(X,\tilde{X}) &= \|\mathrm{diag}(G_{X\tilde{X}}) - 1 + s^*_{ASDP}(G_{XX})\|_2^2 \ .
\end{aligned}
$$

$\lambda = (\lambda_1, \lambda_2, \lambda_3)$, $s^*_{ASDP}(\Omega)$ is a function returning the optimal $s^* = (s^*_1, \ldots, s^*_p)$ from the ASDP-procedure for model-X knockoffs given a covariance matrix Ω. $\hat{D}_{MMD}(X, Z\tilde{X})$ is the empirical version of the maximum mean discrepancy using a Gaussian kernel for comparing two matrices X and $\tilde{X}$. $(X,\tilde{X})_{swap(S)}$ stands for matrix $(X,\tilde{X})$ with entries of X and $\tilde{X}$ swapped in dimensions $S \in \{1,\ldots,p\}$, where each dimension j is contained in S with probability 0.5. This loss function contains three parts. $J_{second-order}(X,\tilde{X})$ measures the deviation from the first moment in λ_3, and the deviation from the diagonal (λ_1) as well as the off-diagonal elements (λ_2) in G. $J_{MMD}(X,\tilde{X})$ penalizes discrepancies in the two covariate-distributions in general, i.e. targeting higher moments. This is done in computationally efficient way by computing the MMD-distance on differently arranged versions of the two independent samples X', X''. Finally, $J_{decorrelation}(X,\tilde{X})$ is added to ensure that the knockoffs $\tilde{X}$ are not highly correlated with X. Otherwise, the algorithm could easily find an optimal trivial solution in just setting $X = \tilde{X}$. The degrees of influence of each of these parts are set by γ, λ, and δ, which should be set depending on the underlying data.

Group Knockoffs

We additionally employ the group knockoff filter from Dai and Barber (2016) as a robustness-check, since it is supposed to handle highly correlated variables better by imposing a group structure. The selection step using the lasso-signed max statistic is taken over a group-lasso regression, encouraging group-sparsity in the selection. W_j^{LSM} is changed accordingly so that groups of variables can are selected in the end. This is

simply done by replacing the individual coefficients by their group counterparts and thus recording at which λ those are included into the model. The construction of good knockoffs, however, is easier due to the imposed structure, and is an extension to what is done in Barber and Candès (2015). Intuitively, variables that are highly correlated should be in the same group, and the procedure should work well correlations among groups are low. In our case, this is not fully fulfilled, which is why we expect the performance to be lower than for the other methods.

Random Forest

We also use random forests (Breiman, 2001) both for prediction as a fully nonparametric machine learning benchmark with a tree-size of 2000. We can extract variable importance from this procedure based on mean variance reduction as a robustness check. This measures the mean reduction in mean squared error by splitting on a certain variable. To aggregate this on a group level, we take the mean over the reduction of all the variables in the respective group. We additionally extract p-values using the PIMP-procedure suggested in Altmann et al. (2010). There, we use 200 permutations of the response variable and measure the variable importance for each permutation to obtain 200 base-importances for each variable. We can then fit a distribution to these *null* importance to obtain a null distribution against which we compare the extracted true variable importance to obtain the p-value. We use a simple and fast nonparametric approach to obtain the p-values by simply measuring the fraction of null importances that exceed the true measured importance relative to the number of permutations. With that, we follow the suggestion of Altmann et al. (2010) and the subsequent implementation in the `R`-package `ranger`.

Model Confidence Sets

While our focus is on interpretation of data-driven selected model components, we also study the predictive ability of the resulting knock-off determined models. In this setting, we compute model confidence sets proposed in Hansen et al. (2011) and implemented in Bernardi and Catania (2018). Such model confidence sets provide practitioners with more robust statistical guidance on which models to apply rather then using simple

prediction errors. Intuitively, the procedure proceeds iteratively, testing whether all models have the same predictive ability, eliminating the worst model, until equal predictive ability cannot be rejected. Define the loss series as in Bernardi and Catania (2018), i.e $d_{ijt} = l_{it} - l_{jt}, i, j \in M = \{1, \ldots, m\}$, as the difference of the squared loss l_{it} and l_{jt} at time point $t = 1, \ldots, T$ for model i and j out of m available models. To construct the test statistic, we compute $d_{i\cdot t} = (m-1)^{-1} \sum_{j \in M \setminus i} d_{ijt}$, the average loss difference between model i and all other (remaining) models. We then construct the statistic $t_{i\cdot} = \dfrac{\overline{d}_{i\cdot}}{\sqrt{\widehat{var}(\overline{d_{i\cdot}})}}$, where $\overline{d_{i\cdot}} = \dfrac{1}{T} \sum_{t=1}^{T} d_{i\cdot t}$. $\widehat{var}(\overline{d_{i\cdot}})$ is the bootstrapped variance using the block-bootstrap with $B = 5000$ bootstrap samples and a block length k that is equal to the number of selected parameters in an auto-regression of the the loss difference series d_{ijt} using the Akaike information criterion to select the model order. The test statistic we use is $T_{max,M} = \underset{i \in M}{max}\ t_{i\cdot}$, and the test rejects if this value is larger than the $1 - \alpha$ quantile of the bootstrapped distribution of $T_{max,M}$.

3.5.2 Tables and Figures

Table 3.8: Most-Selected Groups Over Different Weighting Schemes for Completely Out-Of-Sample Selections

Rank:	1		2		3		4	
	Group	Mean Score	Group	Mean Score	Group	Mean Score	Group	Mean Score
Prob_unweight	14	0.648	12	0.631	11	0.596	5	0.534
Prob_exp	5	0.407	12	0.402	14	0.376	20	0.354
Prob_lin	12	0.506	14	0.493	5	0.462	11	0.458
Rank_unweight	14	17.460	12	15.747	11	15.032	15	14.636
Rank_exp	20	16.876	14	16.495	12	15.919	11	14.853
Rank_lin	14	16.990	20	16.014	12	15.779	11	14.951

Notes: In the columns, *Group* depicts the selected variable group, while *Mean Score* shows the (weighted) mean over all five procedures for the four most selected groups. The ranks for groups are rescaled linearly to lie between 1 and 20 (20 being the best score), while the selection probabilities lie between 0 and 1.

Table 3.9: Groups of Independent Variables

Group 1: Financial Conditions: Loans	
Total Net Loan Charge-offs to Total Loans for Banks	Nonperforming Total Loans (past due 90+ days plus non-accrual) to Total Loans
Nonperforming Loans to Total Loans (avg assets betw. USD 100M and 300M)	Net Loan Losses to Average Total Loans for all U.S. Banks
Loan Loss Reserve to Total Loans for all U.S. Banks	Nonperforming Commercial Loans (past due 90+ days plus non-accrual) to Commercial Loans
Group2: Monetary Measures: Savings	
Personal Saving	Personal Saving Rate
Gross Saving	
Group 3: Monetary Measures: CPIs	
Gross Domestic Product: Implicit Price Deflator	Consumer Price Index for All Urban Consumers: All Items Less Food
University of Michigan Inflation Expectation	Consumer Price Index for All Urban Consumers: Energy
Consumer Price Index for All Urban Consumers: Apparel	Consumer Price Index for All Urban Consumers: All Items
Consumer Price Index for All Urban Consumers: Medical Care	Consumer Price Index for All Urban Consumers: Transportation
Consumer Price Index for All Urban Consumers: All items less shelter	Consumer Price Index for All Urban Consumers: All items less medical care
Consumer Price Index for All Urban Consumers: Durables	Consumer Price Index for All Urban Consumers: Services
Consumer Price Index for All Urban Consumers: Commodities	
Group 4: Monetary Measures: Money Supply	
M2 Money Stock	Board of Governors Monetary Base, Adjusted for Changes in Reserve Requirements
M1 Money Stock	M3 for the United States
Group 5: Corporate Measures: Cash Flow and Profit	
Corporate Profits After Tax (without IVA and CCAdj)	Corporate Profits After Tax with Inventory Valuation and Capital Consumption Adjustments
Corporate Profits after tax with IVA and CCAdj: Net Dividends	Corporate Net Cash Flow with IVA
Group 6: Business Cycle: Unemployment	
Initial Unemployment Claims	Persons unemployed 15 weeks or longer, as a percent of the civilian labor force
Civilian Unemployment Rate	Continued Claims (Insured Unemployment)
Number of Civilians Unemployed for 5 to 14 Weeks	Number of Civilians Unemployed for 15 Weeks and Over
Number of Civilians Unemployed for 15 to 26 Weeks	Number of Civilians Unemployed for 27 Weeks and Over
Number of Civilians Unemployed for Less Than 5 Weeks	Average (Mean) Duration of Unemployment
Group 7: Business Cycle: Industrial Production	
Industrial Production Index	University of Michigan: Consumer Sentiment
Industrial Production: Business Equipment	Industrial Production: Consumer Goods
Industrial Production: Durable Consumer Goods	Industrial Production: Durable Materials
Industrial Production: Final Products (Market Group)	Industrial Production: Fuels
Industrial Production: Manufacturing (SIC)	Industrial Production: Materials
Industrial Production: Nondurable Consumer Goods	Industrial Production: Nondurable Materials
Industrial Production: Manufacturing (NAICS)	

Table 3.9: (Continued)

Group 8: Business Cycle: Private Employment	
Average Weekly Hours of Production and Non-supervisory Employees: Mfg	Civilian Employment
Civilian Employment-Population Ratio	Nonfarm Private Construction Payroll Employment
Nonfarm Private Financial Activities Payroll Employment	Nonfarm Private Goods - Producing Payroll Employment
Nonfarm Private Manufacturing Payroll Employment	Nonfarm Private Service - Providing Payroll Employment
Total Nonfarm Private Payroll Employment	Nonfarm Private Trade, Transportation, and Utilities Payroll Employment
Group 9: Business Cycle: Housing Market	
New Private Housing Units Authorized by Building Permits	Housing Starts: Total: New Privately Owned Housing Units Started
Housing Starts: Total: New Privately Owned Housing Units Started	New One Family Houses Sold: United States
Housing Starts in Midwest Census Region	Housing Starts in Northeast Census Region
Housing Starts in South Census Region	Housing Starts in West Census Region
New Private Housing Units Authorized by Building Permits in the Midwest	New Private Housing Units Authorized by Building Permits in the Northeast
New Private Housing Units Authorized by Building Permits in the South	New Private Housing Units Authorized by Building Permits in the West
Group 10: Business Cycle: Income	
Growth rate of Nominal Disposable Income	Real Disposable Personal Income
National income	Personal Income
Group 11: Stock Market: Index Returns and Volatilities	
S&P 500 Index return	S&P 500 Volatility 1m
CBOE DJIA Volatility Index	NASDAQ 100 Index return
CBOE NASDAQ 100 Volatility Index	Russell 2000 Price Index return
Russell 2000 Vol 1m	Wilshire US Small-Cap Price Index return
Wilshire Small Cap Vol	
Group 12: International Competitiveness: Exchange Rates	
Canada / U.S. Foreign Exchange Rate, Canadian Dollars to One U.S. Dollar	Japan / U.S. Foreign Exchange Rate, Japanese Yen to One U.S. Dollar
Switzerland / U.S. Foreign Exchange Rate, Swiss Francs to One U.S. Dollar	U.S. / U.K. Foreign Exchange Rate, U.S. Dollars to One British Pound
Real Broad Effective Exchange Rate for United States	
Group 13: International Competitiveness: Trade	
Real Trade Weighted U.S. Dollar Index: Broad	Trade Weighted U.S. Dollar Index: Major Currencies
Total Current Account Balance for the United States	Real Exports of Goods & Services
Real imports of goods and services	

Table 3.9: (Continued)

Group 14: Micro-level: Bond Yields and Interest Rates	
Bank Prime Loan Rate	1-Month AA Non-financial Commercial Paper Rate
10-Year Treasury Constant Maturity Rate	3-Month AA Non-financial Commercial Paper Rate
Term Structure	Effective Federal Funds Rate
Moody's Seasoned Baa Corporate Yield Relative to Yield on 10-Year Treasury	Moody's Seasoned Aaa Corporate Bond Yield
30-Year Conventional Mortgage Rate	Moody's Seasoned Baa Corporate Bond Yield
1-Year Treasury Constant Maturity Rate	5-Year Treasury Constant Maturity Rate
3-Month Treasury Bill: Secondary Market Rate	3-month Treasury Constant Maturity Rate
6-Month Treasury Bill: Secondary Market Rate	Moody's Seasoned Aaa Corporate Bond Minus Federal Funds Rate
Moody's Seasoned Baa Corporate Bond Minus Federal Funds Rate	3-Month Commercial Paper Minus Federal Funds Rate
Moody's Seasoned Aaa Bbb Spread	Size of High Yield Market in U.S. Dollars
Group 15: Micro-level: Bond Defaults in Industry	
Bond defaults within the industry (in percent)	
Group 16: Micro-level: High Yield Default Rate	
High Yield Default Rate, Trailing 12-month	
Group 17: Financial Conditions: Bank Credit and Debt	
Loans and Leases in Bank Credit, All Commercial Banks	Real Estate Loans, All Commercial Banks
Federal Debt: Total Public Debt	Total Consumer Credit Owned and Securitized, Outstanding
Excess Reserves of Depository Institutions	Commercial and Industrial Loans, All Commercial Banks
Total Borrowings of Depository Institutions from the Federal Reserve	Bank Credit of All Commercial Banks
Household Debt Service Payments as a Percent of Disposable Personal Income	Household Financial Obligations as a percent of Disposable Personal Income
Loans and Leases in Bank Credit, All Commercial Banks	
Group 18: Business Cycle: Real GDP	
Real Gross Domestic Product	Government Consumption Expenditures
Growth rate of nominal GDP	
Group 19: Micro-level: Producer Price Index	
Producer Price Index by Commodity Industrial Commodities	Producer Price Index by Commodity Intermediate Energy Goods
Producer Price Index by Commodity for Crude Energy Materials	Producer Price Index by Commodity for Finished Consumer Goods
Producer Price Index by Commodity Intermediate Materials	Producer Price Index for All Commodities
Group 20: Business Cycle: Inventories	
Capacity Utilization: Manufacturing	Change in Private Inventories
Capacity Utilization: Total Industry	Total Business Inventories

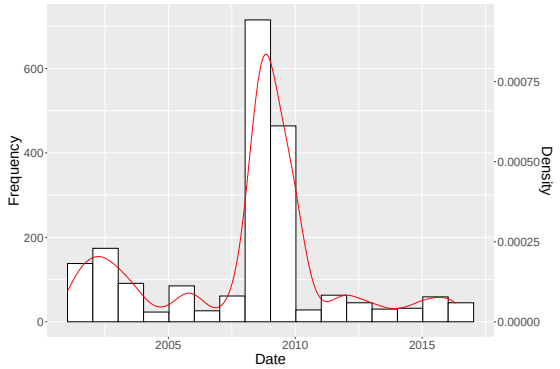


Figure 3.4: Default Frequency and Density (Red) Over Time for the Defaulted US Corporate Bonds From 2001 to 2016

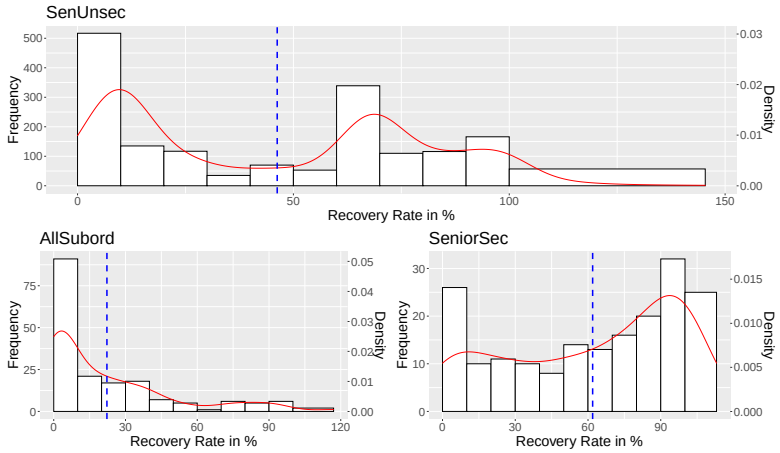


Figure 3.5: Recovery Rate Frequency and Density (Red) for the Defaulted US Corporate Bonds From 2001 to 2016

Notes: Mean Recovery Rates are depicted in dashed lines (Blue). Defaults are sorted by bond type, i.e. senior unsecured bonds (*SenUnsec*, $n = 1715$), all subordinate bonds (*AllSubord*, pooled because of insufficient data, $n = 178$, from subordinate bonds: $n = 158$ and senior subordinate bonds, $n = 21$), and senior secured bonds (*SenSec*). Please also notice the different scaling of the x-axis.

Table 3.10: Variable Importance From Random Forests Aggregated on Group Level

	Variance Reduction	P-Value
Group_15	4.533	0.005
Group_12	2.411	0.005
Group_11	1.678	0.008
Group_19	1.481	0.005
Group_2	1.433	0.085
Group_14	1.162	0.008
Group_13	1.046	0.027
Group_17	0.902	0.046
Group_3	0.870	0.016
Group_20	0.763	0.091
Group_4	0.602	0.007
Group_8	0.574	0.020
Group_6	0.535	0.008
Group_7	0.529	0.071
Group_5	0.393	0.139
Group_1	0.321	0.112
Group_9	0.298	0.140
Group_10	0.247	0.128
Group_18	0.138	0.320
Group_16	0.085	0.075

Notes: Variance reduction is the mean variance reduction (i.e. influence) of a variable in the random forest, measured by the mean reduction in mean squared error by splitting on this variable, averaged over all groups. For ease of presentation, we show values relative to the average of variance reduction taken over all variables (i.e. a value larger than 1 indicates higher importance). P-Value depicts the mean p-values obtained by the PIMP-procedure from Altmann et al. (2010) given 200 permutations.

Table 3.11: Mean Selection Probabilities for Each Procedure Over All Weighting Schemes

Method:	modelX_PCA	deep5_PCA	deep25_PCA	modelX2comp_PCA	Group	gKnock_Data
PC1.1	0.125	0.695	0.357	0.212	G_1	0.081
PC1.2	0.884	0.990	0.980	0.279	G_2	0.313
PC2.1	0.298	0.990	0.899	0.227	G_3	0.195
PC2.2	0.116	0.952	0.561	0.137	G_4	0.099
PC3.1	0.103	0.526	0.354	0.085	G_5	0.080
PC3.2	0.129	0.683	0.336	0.105	G_6	0.073
PC4.1	0.125	0.677	0.239	0.087	G_7	0.148
PC5.1	0.926	0.990	0.980	0.587	G_8	0.079
PC5.2	0.170	0.888	0.591	0.116	G_9	0.179
PC6.1	0.107	0.606	0.350	0.084	G_10	0.157
PC6.2	0.193	0.950	0.551	0.129	G_11	0.202
PC7.1	0.159	0.990	0.852	0.103	G_12	0.264
PC7.2	0.259	0.990	0.970	0.384	G_13	0.112
PC7.3	0.361	0.990	0.980		G_14	0.294
PC8.1	0.113	0.662	0.317	0.105	G_15	0.464
PC8.2	0.115	0.958	0.612	0.085	G_16	0.120
PC9.1	0.294	0.990	0.905	0.131	G_17	0.098
PC10.1	0.109	0.468	0.194	0.089	G_18	0.321
PC10.2	0.227	0.990	0.846	0.200	G_19	0.256
PC11.1	0.065	0.985	0.750	0.004	G_20	0.388
PC11.2	0.709	0.990	0.977	0.763		
PC11.3	0.310	0.990	0.871			
PC11.4	0.262	0.990	0.938			
PC12.1	0.141	0.990	0.889	0.215		
PC12.2	0.721	0.990	0.977	0.616		
PC13.1	0.111	0.722	0.386	0.093		
PC13.2	0.184	0.990	0.830	0.100		
PC13.3	0.306	0.990	0.976			
PC14.1	0.489	0.990	0.843	0.116		
PC14.2	0.793	0.990	0.980	0.848		
PC14.3	0.025	0.688	0.376			
PC15.1	0.361	0.990	0.783	0.422		
PC16.1	0.307	0.990	0.840	0.496		
PC17.1	0.109	0.535	0.268	0.084		
PC17.2	0.127	0.609	0.266	0.125		
PC17.3	0.119	0.880	0.376			
PC18.1	0.111	0.598	0.389	0.389		
PC18.2	0.208	0.990	0.905	0.264		
PC19.1	0.120	0.601	0.267	0.086		
PC20.1	0.372	0.990	0.980	0.182		
PC20.2	0.118	0.769	0.384	0.130		

Notes: Mean selection probabilities over the three weighting schemes for each procedure and variable. For ranking the variables, e.g. in the forecasting, the higher index is chosen first in case two probabilities are exactly the same (only relevant for the deep knockoff procedures).

Table 3.12: PCA-Weights for Groups of Our Proposed Procedure

	PC1	PC2	PC3	PC4
Group_14				
DCPN30	-0.267	0.046	0.088	-
DGS10	-0.217	-0.244	-0.22	-
DCPN3M	-0.267	0.043	0.108	-
TermStructure	-0.267	0.043	-0.008	-
FEDFUNDS	-0.266	0.033	0.059	-
BAA10YM	0.117	-0.27	0.463	-
DAAA	-0.176	-0.37	-0.109	-
MORTGAGE30US	-0.222	-0.265	-0.009	-
DBAA	-0.103	-0.433	0.249	-
DPRIME	-0.266	0.052	0.072	-
DGS1	-0.268	0.016	0.061	-
DGS5	-0.244	-0.16	-0.161	-
DTB6	-0.268	0.039	0.061	-
TB3MS	-0.268	0.027	0.037	-
AAAFF	0.203	-0.313	-0.157	-
BAAFF	0.194	-0.343	0.104	-
CPFF	0.002	-0.05	0.71	-
AaaBbbSpread	0.203	-0.313	-0.157	-
HYMSIZE	0.165	0.342	0.192	-
DGS3MO	-0.267	0.043	-0.008	-
Group_11				
SPRet	0.427	0.189	-0.053	0.053
VolSP500	-0.351	0.365	-0.292	-0.191
VXDCLS	-0.374	0.31	-0.258	-0.171
NasdaqRet	0.357	0.352	-0.208	-0.391
VXNCLS	-0.331	0.211	0.21	0.664
RussellRet	0.38	0.357	0.097	0.1
Russell2000Vol1m	0.154	-0.097	-0.833	0.479
WilshireRet	0.384	0.05	0.173	0.266
WilshireVol1m	0.049	-0.654	-0.155	-0.164
Group_12				
DEXCAUS	-0.51	0.131	-	-
DEXJPUS	-0.4	-0.561	-	-
DEXSZUS	-0.476	-0.222	-	-
DEXUSUK	0.305	-0.785	-	-
RBUSBIS	-0.51	0.047	-	-
Group_20				
CAPUTLB00004SQ	-0.568	0.134	-	-
CBI	-0.531	0.176	-	-
TCU	-0.564	0.173	-	-
BUSINV	-0.278	-0.96	-	-
Group_5				
CP	0.54	-0.072	-	-
CPATAX	0.545	0.193	-	-
DIVIDEND	0.418	-0.805	-	-
CNCF	0.487	0.556	-	-

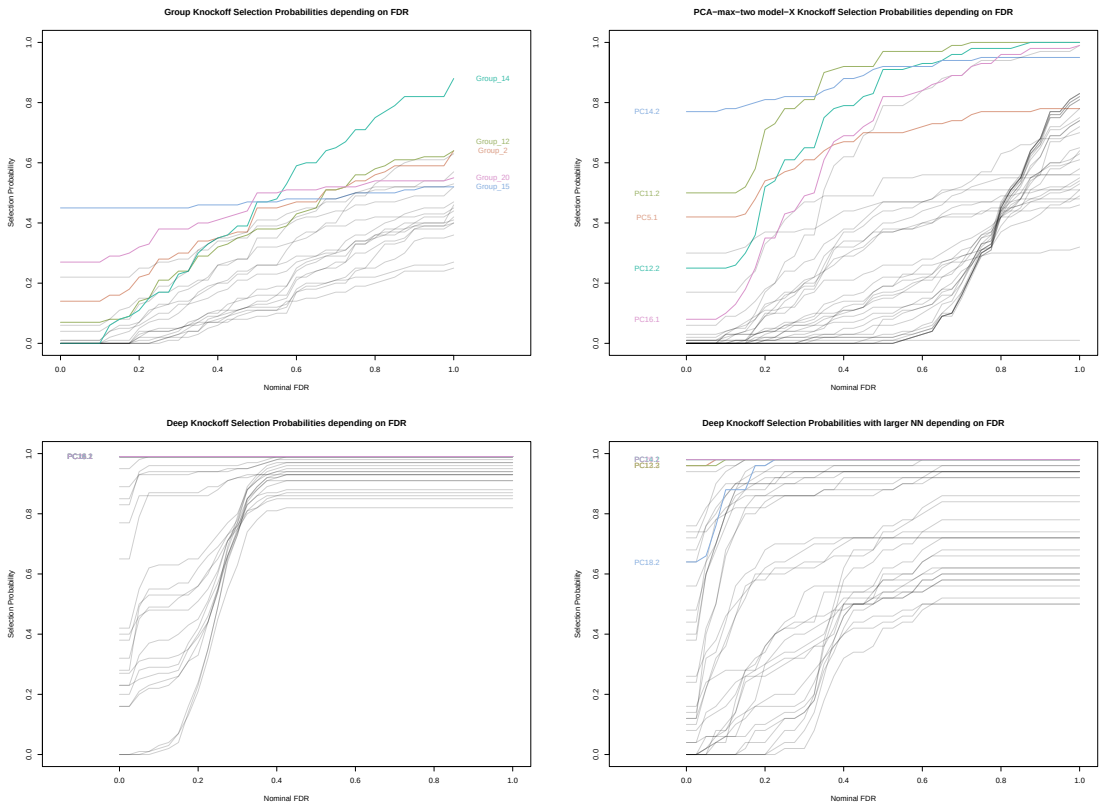


Figure 3.6: Selection Probabilities for Different Baseline Knockoff Procedures

Notes: The knockoff methods are the group knockoff (top-left), model-X knockoff (top-right) with a maximum of two PCs per group, and deep knockoffs using 5 neurons (bottom-left) or 25 neurons (bottom-right) per Layer. Selection Probabilities are obtained rerunning the full knockoff procedures using repeated subsampling of 90% of the data (100 iterations). Highlighted groups have the highest mean selection rank, i.e. the mean over the rank in each FDR-scenario. The group with the highest probability receives the highest rank (= 41 or = 20 for the Group knockoffs) and vice versa (= 1).

4 How Have German University Tuition Fees Affected Enrollment Rates: Robust Model Selection and Inference in High Dimensions

4.1 Introduction

In this paper, we study the causal effect of the introduction of a flat state-dependent tuition fee on university student enrollment behavior using official data for all 16 federal German states. In particular, we show how to derive a common federal average causal effect of tuition fees for limited administrative state-level data in the presence of a large amount of potentially influencing attributes also on the policy decision. In Germany, universities have generally been public and essentially free of charge but during the years 2006-2014 a maximum tuition fee of 1000 Euros per year was allowed. Only some states chose a tuition fee for their universities, and if they did they generally set it to the maximum level. Moreover, the implementation and timing of the fees, both, were no exogenous shock but driven policy decisions on the federal state level ("Bundesländer", denoted as states in the following) and thus varied among states. At the same time, however, major policy changes in different federal states also significantly impacted the cohort size of prospective university students.[1] This spatial time delay in the implementation of both tuition fees and different federal reforms induced substantial

[1]This comprises a decrease for the required compulsory years to high school graduation from nine to eight years of which the introduction varied on the state level, and the general German-wide abolishment of the 9 month compulsory military service for men in the age of 17-23.

migration effects which potentially impacted state-level student enrollment in addition to the many standard socio-economic state characteristics.

We thus suggest a stability post-double selection methodology (cp. Belloni et al. (2014a)) to robustly determine the causal effect in such a high-dimensional setting with many potentially influential controls and few observations with measurement problems. With a robust subsampling-augmented Lasso procedure (cp. Meinshausen and Bühlmann (2010)), we adaptively select the relevant controls not only in the outcome equation, but also crucially augment this set with the Lasso selection choices in an auxiliary propensity score equation. Given the strong correlation of the tuition fee decision and the control variables, this double-selection-type strategy ensures that underspecification and resulting biased estimates are not an issue. Overall, with these tailored data-driven techniques, we detect a significant negative effect of tuition fees inducing an up to 4.5 percentage point (pp) reduction in enrollment rates. Since the exact enrollment rate suffers from measurement problems, we show the stability of our results over a large grid of values. While spatial cross-effects have been ignored in the previous literature on German tuition fees (see e.g. Dwenger et al., 2012; Bruckmeier and Wigger, 2014; Mitze et al., 2015), we identify them as important drivers for enrollment rates by the Lasso, besides state specific factors such as the student-to-researcher ratio. We explicitly show that without Lasso pre-selection of variables, the signal to noise ratio of the problem is too low for detecting the correct magnitude of the effect. Generally, these insights and our methodological solution are highly relevant for all cases of policy evaluation, where implementation occurs in a spatially time-delayed manner, as for example environmental policies that target global warming or financial regulations in different countries. In addition, we believe that our empirical findings cannot only contribute to the active ongoing discussions on reintroducing tuition fees in Germany, but might also be of independent interest for other countries such as the United Kingdom, where fees are on the rise.

For the analysis we study the years 2005-2014 and all 16 federal states in Germany. We include a comprehensive set of 18 covariates, covering all potentially important controls of the national and international literature on tuition fee effects (e.g. Dynarski (2003); Kane (1994) and Baier and Helbig (2011); Dwenger et al. (2012); Bruckmeier and Wigger (2014); Mitze et al. (2015)). The variables are collected from different sources, but public data

on student enrollment behavior is only available on the state level and not on a university level, which is due to strict German data protection laws.[2] In addition to standard economic, social and educational factors from the literature on student enrollment rates, we also include specific effects for Germany which play a major role in the considered period. Particularly, policy changes such as the abolishment of mandatory military service or the heterogeneous introduction of a one-year reduced secondary education ("G8") in different states are key policies. Moreover, in addition to the above standard list of controls, we construct spatial variables that capture state cross-effects in the policy decisions for or against fees as the proportion of students migrating to each state from states with and without tuition fees based on their proximity. These are crucial to control for migration effects due to heterogeneous implementation and time delay of policies across states that could otherwise bias the estimated effect of tuition fees. We work with relative enrollment rates instead of absolute numbers as the dependent variable to ensure compatibility of effects across federal states of different population sizes. For correct ratios, however, we require the population size of all high school graduates affected by the introduction of tuition fees in a specific state. This quantity is hard to measure and thus prone to measurement errors as it consists not only of recent and less recent high school graduates from this specific state, but also of parts of cohorts from other states and abroad from where students migrate to study. We transparently treat this measurement ambiguity and thus provide results that are robust in this respect. Overall, the limitation to only state-level data results in a relatively small number of available observations where single observations could gain substantial influence on the overall result. Thus in total, we face a situation of many potentially influential but correlated covariates and relatively few observations with possible outliers due to data quality problems.

We tackle these challenges with a tailored subsampling-augmented variable selection technique in a fixed effects panel regression with many controls. The Lasso type double selection is key for avoiding underspecification in the outcome equation since the tuition fee policy treatment decision is strongly correlated with observed controls (cp. Belloni et al. (2014b,a) and Belloni et al. (2016) for a panel setup). In this, the data-driven

[2]Note that across states and universities, individual or household panel data from common sources such as e.g. the German SOEP is insufficient, incomplete and very unbalanced and cannot be employed for a general analysis. Please see Appendix 4.7.1 for details.

choice of covariates from the auxiliary propensity score equation is used to complement the Lasso-determined active set of relevant regressors in the outcome equation allowing for unbiased estimation of the causal effect. For both selection steps, we propose a subsampling based stability selection (see Meinshausen and Bühlmann (2010)) in order to mitigate correlation effects among covariates and measurement issues in the available small set of observations. In such cases, pure Lasso might have difficulties in correctly predicting the influence of each variable, which can lead to the choice of too many variables. We illustrate in a thorough simulation study for such challenging situations, that the suggested stability selection substantially improves on the robustness of the selection results in finite samples leading to augmented post-selection estimation results. Given the scarcity of the available public data and the complexity of the setting, the estimated specification in both the outcome and the auxiliary equation is set as linear which allows for the direct identification of the causal effect. Along with the usual HC3 standard errors (SEs) we additionally report design-based SEs in the simulation and results (Abadie et al. (2020) and Athey and Imbens (2022)).

Our set-up corresponds to the high-dimensional machine learning driven causal literature (see Belloni et al. (2014a, 2016) and Athey et al. (2019) for a survey as well as applications in labor Angrist and Brigham Frandsen (2019)) for the estimation of average treatment effects. With our aggregate state-level data, we can determine a common (average) causal effect of a policy or treatment as e.g. Rubin (1974) or Rubin (1977) in the standard low-dimensional potential outcomes framework. In our case, however, standard methods as e.g. simple difference-in-differences (Card and Krueger, 1994; Ashenfelter and Card, 1985), low-dimensional propensity score or matching techniques (see e.g. Rosenbaum and Rubin (1983) or for an overview on nonparametric, non-linear methods Imbens (2004)) or simple one-step LASSO variants thereof cannot adapt to the short available time span and few states in order to detect the tuition fee effect. Similarly, heterogeneous treatment effects as e.g. in Athey and Imbens (2016); Chernozhukov et al. (2018); Chang (2020); Athey and Imbens (2022) also require are much larger cross-section of e.g sub-state, university level data which is not publicly accessible in our case.

Up to our knowledge, the literature on student enrollment behavior generally works with only small sets of covariates on which there is no consensus and often subset selection

is only ad-hoc or based on heuristics. Therefore, we propose a data-driven statistical procedure in order to empirically identify relevant factors. Nevertheless, there are analyses on effects of tuition fees in various countries that mostly find significant effects only for certain subgroups of the population. Kane (1994), Noorbakhsh and Culp (2002) and McPherson and Schapiro (1991) find negative effects of tuition fees[3] for low-income groups or groups with African-American ethnicity for the US. More generally, Neill (2009) finds that an increase in tuition fees reduces enrollments significantly for the Canadian system. With the availability of individual data in the presence of much higher fees, but also an established scholarship system, US and Canadian studies can identify effects of tuition fees on enrollment that range between −2.5pp and −6.8pp. For countries where the situation is more comparable to the German system, and the particular case of Germany, previous studies generally cannot detect significant effects of tuition fees on enrollment rates (see e.g. for Germany Baier and Helbig (2011); Hübner (2012); Dwenger et al. (2012); Bruckmeier and Wigger (2014); Mitze et al. (2015), but also Huijsman et al. (1986) for the Netherlands and Denny (2014) for Ireland). This seems to be caused by the small number of included covariates, while missing out on the key ones according to our statistical selection technique. Variables possibly correlated with the tuition fee decision are mostly ignored, as well as state cross effects through differences in timing, which we show both to be relevant. Moreover, we cover the comprehensive list of all German tuition fee periods and states, which helps to increase precision of estimated effects in contrast to previous studied, who focused only on subperiods, specific states or subgroups. With mostly insignificant effects between −0.4pp and −2.69pp, the previous German studies seem to systematically underestimate the true impact of fees.

The remainder of the paper is structured as follows. A description of the data set and variables is presented in Section 4.2. It also contains the transparent construction of (a set of) response variables from the limited available information. Section 4.3 introduces the linear panel model and the Lasso-type selection methods featuring the stability double selection. In Section 4.4, a Monte Carlo simulation shows the advantages of these

[3] In the study of McPherson and Schapiro (1991), the authors find that the net costs (tuition fees minus student aid) have a negative impact, which is an even stronger argument.

methods with different distortions in a controlled environment. After discussing the main results of our empirical study in Section 4.5, we conclude in Section 4.6.

4.2 Data

We construct a panel from official, publicly available data on enrollment numbers and socio-economic and university-related covariates for the 16 German states ($n = 16$) in the years 2005 to 2014 ($T = 10$). We use a widespread set of potential controls for determining the effect of tuition fees which only existed in the years 2006-2014 in at least one state (see Figure 4.8 in Appendix 4.7.1 for an overview of the timing of fees in different states). The years 2005 and 2014 serve as a base for comparison before and after the introduction and complete abolishment of tuition fees[4]. Note that we are limited to state level aggregated data, since available individual or household type survey data from common sources such as e.g. the German Socio-Economic Panel (SOEP) is very unevenly distributed across states and universities and suffers from incompleteness (see Appendix 4.7.1 for details). Moreover, strict privacy protection laws prevent the dissemination of more granular official data beyond the federal state level.

4.2.1 Construction of the Response

As the response variable we study the enrollment rate $y_{i,t}$ of high school graduates into university in state i at the winter term (WT) of year t to $t + 1$ (denoted as $t/t + 1$) which we perceive as the most directly affected observable quantity by tuition fees.[5] As universities we denote all public general university type institutions comprising universities, specialized technical, arts and music universities but also universities of applied sciences (Fachhochschule) and cooperative state universities (Duale Hochschule).[6] Since the population size among German states varies substantially, relative enrollment rates $y_{i,t}$ ensure comparability of results across states. This is key for identifying a

[4] As the only state, Lower Saxony abolished Tuition fees only by the end of the summer term 2014, which is why we still use 2014 as a base for total abolishment of fees.

[5] The academic year starts with the winter semester usually beginning in September or October of year t and ending in February of year $t + 1$.

[6] More than 90% of universities in Germany are public.

meaningful average causal effect of tuition fees across states and in contrast to the simple absolute number of new enrollments (from anywhere) $NE_{i,t}$ in state i at WT of year $t/t+1$, where actual state-specific effects would vary by size. Though actual enrollment rates per state are not reported and can only be approximated.

The percentage $y_{i,t}$ is obtained as the quotient of the number of enrollments $NE_{i,t}$ in state i and the so-called eligible set $EHG_{i,t}$ of high school graduates for year t coming to or staying in state i, which can generally differ substantially from the own-state high school graduates $HG_{i,t}$ in i of this specific year. We set

$$y_{i,t} = \frac{NE_{i,t}}{EHG_{i,t}} \ , \tag{4.1}$$

where we model $EHG_{i,t}$ to consist of three main different groups, namely own i-specific high school graduates $HG_{i,t}$, potentially "affected" graduates $AHG_{j,i,t}$ from other German states and the number of new international enrollments in i, $NE_{i,t}^{(int)}$:

$$EHG_{i,t} = HG_{i,t} + \sum_{j \neq i} AHG_{j,i,t} + NE_{i,t}^{(int)} \ . \tag{4.2}$$

While respective enrollment numbers $NE_{i,t}^{(i)}$ from i in i, $NE_{i,t}^{(j)}$ from j to i and $NE_{i,t}^{(int)}$ of international students in i are publicly available for any state i in WT $t/t+1$, there is, however, no available direct data for the respective eligible quantities in (4.2). For the generally dominant from i to i component, this can be well approximated by its upper bound of the number of all high school graduates in i as in the German federal system, the "home state" of the high-school diploma is often part of the immediate choice set of university entrants. Since the share of international students remains stable at around 15% over the years due to effects such as language barriers in German undergraduate programs, we assume that the low amount of tuition fees in the international context has no effect and we therefore only use the lower bound $NE_{i,t}^{(int)}$ in the eligible set. Though for the eligible part of potential movers $AHG_{j,i,t}$ from j to i within Germany, extreme approximations by its lower bound of the number of enrollments $NE_{i,t}^{(j)}$ or the upper bound of all graduates $HG_{j,t}$ in j are too coarse. In particular in view of tuition fee interventions, it is clear that $AHG_{j,i,t}$ is affected, but unclear how. We therefore model it explicitly as a convex combination between the potential extremes.

$$AHG_{j,i,t} = \theta NE_{i,t}^{(j)} + (1-\theta) HG_{j,t} \ , \tag{4.3}$$

with $\theta \in [0,1]$. Of course, choosing θ too low, i.e. giving $HG_{j,t}$ too much influence, will yield $y_{i,t}$ values that are unrealistically low. An absolute lower boundary would be a mean enrollment of $\bar{y}_{0.90} = 0.25$, which is achieved at $\theta = 0.9$. Looking at the aggregated number of all new enrollments (not just first-time students) in all of Germany from German high schools over 2003-2014 divided by all high school graduations in Germany at that time in our data, we have a mean enrollment rate of around 0.72, which can serve as a very rough proxy for where to expect realistic values. If we only look at first-time enrollments, the rates have monotonically increased from 40% in 2009 over the years.[7] We therefore take $\theta = 0.98$ as a reasonable lower θ-boundary, which yields $\bar{y}_{0.98} \approx 0.4$. We then conduct our analysis transparently over a grid of θ-values in between 0.98 and 1 which we denote as admissible θs and which yield mean enrollment rates $\bar{y}_\theta \geq 0.4$. Figure 4.7 in Appendix 4.7.1 shows the mean enrollment rates over θ indicating the sensitivity of y with respect to θ in the considered range. With additional information on the number of new enrollments $NO_{j,t}$ with graduation in state j enrolling anywhere in Germany at t and non-public information on the number of postponers, we can also improve the approximation of $EHG_{i,t}$ yielding $EHG^*_{i,t}$ which we take as the benchmark case for our purely public empirical analysis (See Appendix 4.7.1 for the construction). In particular, for $\theta^* = 0.9927$, the empirical mean squared and mean absolute deviation of $EHG^*_{i,t}$ and $EHG_{i,t}$ over all i and t are minimized and both almost coincide. As a robustness check to our pure public data analysis, we also report results for a response $y^{extra}_{i,t} = \dfrac{NE_{i,t}}{EHG^*_{i,t}}$.

4.2.2 Covariates and Data Challenges

In the covariates, we model the treatment effect $d_{i,t}$ of a tuition fee as a dummy, with $d_{i,t} = 1$ indicating an existing tuition fee in state i in the winter term starting in year t and $d_{i,t} = 0$ otherwise.[8] Because of German laws, each state could strategically decide on the introduction and timing of fees.

[7] Data source: Federal ministry of education (BMBF) data web-space `http://www.datenportal.bmbf.de/portal/de/K253.html` Table 1.9.3

[8] In Germany, there were no fees for students studying for their first degree in public institutions from WT of 2014 and onward. Before that, the maximum amount for first degree studies was limited to €1000 per year. Almost all universities made use of the maximum amount, thus suggesting a dummy variable design.

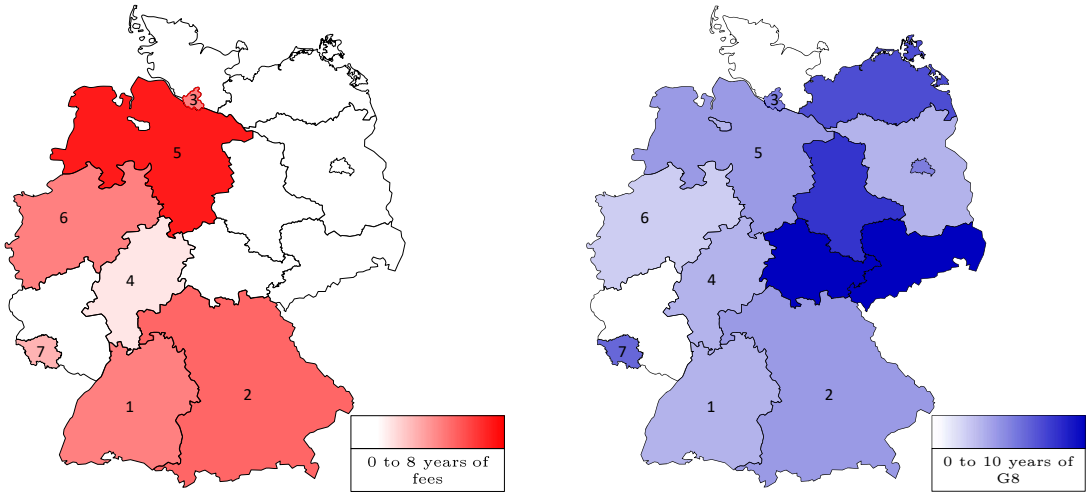


Figure 4.1: Overview of the Presence of Tuition Fees (Left) and the G8-Reform (Right) in the 16 German States Until 2015
Note: Darker colors represent longer presence of the respective variable.

For identification of the model given the different treatment timings, we use observable spatial controls $z_{i,t}$ that capture migration behavior to each state from other state groups aggregated in proximity and fee categories. These are key because of the heterogeneity of introduction and abolishment of tuition fees over states that can be seen in Figure 4.8 in Appendix 4.7.1. Additionally, there are many cases where fee-states border non-fee states, which is highlighted in Figure 4.1. We therefore construct the spatial controls as share of new enrollments in state i that obtained their high school diploma in another state group. For each state i, we measure the proportion of new enrollments from a specific state group (e.g. neighboring fee states) relative to all enrollments in i. The groups consist of fee states that have a shared border with i, fee-states without a shared border with i, non-fee states, and enrollments from outside Germany (*Migration.international*). For example, *Migration.neighbor.fees* measures the proportion of new enrollments from all fee states with shared border to i relative to all enrollments in i that year. A detailed description can be found in Table 4.6 in Appendix 4.7.1. Furthermore, to control for non-constant state specific effects, we employ 14 control variables $x_{i,t}$ using data from the socio-economic

panel (SOEP)[9] and Destatis[10], the Federal Statistical Office in Germany. A detailed description can be found in Table 4.5 and Table 4.4 in Appendix 4.7.5. Together with the spatial variables, we have a set of $p = 18$ potentially relevant covariates plus the binary variable of tuition fees. Among others, we capture socio-economic variables comprised of urbanization level, income, rent, life satisfaction, unemployment rate and university and student related controls on staff and graduation statistics, the student-to-researcher ratio and data on the funding of universities. In particular, this set of variables contains all types of relevant controls from similar, previous studies (e.g. Bruckmeier and Wigger (2014); Mitze et al. (2015)). Moreover, we include two variables on the G8-reform that reduced the time of secondary education from nine to eight years. The implementation of this major educational policy change was also heterogeneous across states and is illustrated in blue in Figure 4.1. This reform almost immediately substantially impacted the timing and the overall likelihood of much younger high school graduates to enroll to a university. We control for this effect with a dummy $G8_{i,t}$, where positive values indicate that the G8-reform was implemented in this state i, and additionally mark transition period years of double cohorts of G8 and G9 cohorts graduating by $DC_{i,t} = 1$.

We formally determine that our data is characterized by a few single observations in response and covariates which are highly influential. In particular, we compute the *DFFITS* for measuring the impact of each observation k on the resulting fit $\hat{y}$ and find that the fitted enrollment rates heavily change when specific single observations are dropped from the regression estimation. For covariates, these single observational effects are even more pronounced as measured by *DFBETAS* for the leverage of on observation k on the estimated linear effects, and thus largely impact model selection and estimation. For detailed results and definitions of the considered quantities, please see Figure 4.9 in the Appendix. In addition to high expected correlations between regressors, it further encourages the use of stability selection instead of using all data points just once.

[9] We use the SOEP-long version 31. More information at `https://www.diw.de/en/diw_01.c.519381.en/1984_2014_v31.html`; for the usage, see Wagner et al. (2007)

[10] More information at `https://www.destatis.de/EN`. Some variables were generated using data from Genesis-online database of Destatis accessible at `https://www-genesis.destatis.de`.

4.3 Model and Methodology

4.3.1 Model

The key goal of our study is to determine a finite sample precise estimate of the causal effect of tuition fees $\beta_{(0)}$ on enrollment rates y. For this, we work with federal state size-weighted enrollment rates rather than enrollment numbers to identify a federal causal effect in a linear panel set-up with spatial effects and many controls for the variety in states. We propose a model determination procedure with a stable but parsimonious data-driven selection of controls that is stable with respect to the correlation of the policy decision with state-specific controls. For our setting with limited data and potential measurement issues, this not only prevents cherry-picking of variables but also countervails biased causal effects for particular strong correlations of treatment and controls. Moreover, we illustrate how correct standard errors can be obtained quantifying the causal uncertainty when working with a complete population rather than a sample.

We use a two-equation linear panel model with fixed effects α_i, where the covariates in both equations consist of socio-economic variables $x_{i,t}$ and spatial factors $z_{i,t}$. In the outcome equation, for each admissible θ in (4.3), the focus is on the linear causal effect of the tuition fee dummy $d_{i,t}$ on enrollments $y_{i,t}(\theta)$ given the large set of controls $(x_{i,t}, z_{i,t})$.[11] The auxiliary propensity score equation is also linear in $(x_{i,t}, z_{i,t})$ and only serves as a correction device for data-driven model selection in the outcome equation due to correlation of $d_{i,t}$ and $(x_{i,t}, z_{i,t})$. Thus we use the following model specification for $i = 1, \ldots, n = 16$ states and $t = 1, \ldots, T = 10$ years

$$y_{i,t} = \beta_{(0)} d_{i,t} + \beta_{(1)}^{\mathsf{T}} \begin{pmatrix} x_{i,t} \\ z_{i,t} \end{pmatrix} + \alpha_i + \epsilon_{i,t}^{(1)} , \tag{4.4}$$

$$d_{i,t} = \beta_{(2)}^{\mathsf{T}} \begin{pmatrix} x_{i,t} \\ z_{i,t} \end{pmatrix} + \epsilon_{i,t}^{(2)} , \tag{4.5}$$

with $y_{i,t},\ \beta_{(0)},\ d_{i,t},\ \alpha_i,\ \epsilon_{i,t}^{(1)},\ \epsilon_{i,t}^{(2)} \in \mathbb{R}$ and $\binom{x_{i,t}}{z_{i,t}} \in \mathbb{R}^p$ with $p = 18$. The α_i are fixed effects comprising e.g. unobserved regional aspects such as climate conditions, culture, or the topography of a state which might generally be correlated with at least some of the covariates $(x_{i,t}, z_{i,t})$ such as e.g. rent or the urbanization level. Generally, decisions about

[11] For ease of exposition, we omit θ in the following in $y_{i,t}(\theta)$.

the implementation of tuition fees in each state were taken at least one or two years ahead of the implementation date, and were thus not influenced by actual enrollment numbers $y_{i,t}$. Hence the large set of controls including spatial factors for potential migration effects, the strict exogeneity conditions for both equations can be assumed as fulfilled, i.e. it holds that $\mathbb{E}[\epsilon_{i,t}^{(1)} \mid d_{i,1}, \ldots, d_{i,T}, x_{i,1}, \ldots, x_{i,T}, z_{i,1}, \ldots, _{i,T}, \alpha_i] = 0$, $\mathbb{E}[\epsilon_{i,t}^{(2)} \mid x_{i,1}, \ldots, x_{i,T}, z_{i,1}, \ldots, z_{i,T}] = 0$. Linearity in both equations does not only account for the scarce data situation but also yields $\beta_{(0)}$ as a causal effect. For identification of this common average tuition fee effect, we assume that tuition fee decisions in other states influence the propensity for tuition fees and the enrollment rate in state i only through the respective migration effects z.

In the following, we work with the standard fixed effects transformation of (4.4) and (4.5) removing α_i by demeaning:

$$\ddot{y}_{i,t} = \beta_{(0)}\ddot{d}_{i,t} + \beta_{(1)}^{\mathsf{T}}\begin{pmatrix}\ddot{x}_{i,t}\\ \ddot{z}_{i,t}\end{pmatrix} + \ddot{\epsilon}_{i,t}^{(1)} \ , \tag{4.6}$$

$$\ddot{d}_{i,t} = \beta_{(2)}^{\mathsf{T}}\begin{pmatrix}\ddot{x}_{i,t}\\ \ddot{z}_{i,t}\end{pmatrix} + \ddot{\epsilon}_{i,t}^{(2)} \ , \tag{4.7}$$

with $\ddot{y}_{i,t} = y_{i,t} - \overline{y}_i$ with $\overline{y}_i = \dfrac{1}{T}\sum_{t=1}^{T} y_{i,t}$ and similarly $\ddot{d}_{i,t}$, $\ddot{x}_{i,t}$, $\ddot{z}_{i,t}$, $\ddot{\epsilon}_{i,t}^{(1)}$, $\ddot{\epsilon}_{i,t}^{(2)}$.

4.3.2 Robust Model Selection and Post-Lasso Inference

The proposed model selection and estimation procedure is two-step, where in step one, covariates are automatically selected separately in the outcome and the auxiliary equation. In step two, the union of the two sets of pre-selected covariates is then used to identify the causal effect of interest. Moreover, in our situation of $\frac{nT}{p} = 8.89$, observations are so scarce relative to the dimensionality of the problem that plain OLS-type estimates are extremely imprecise. Thus for proper estimation of our main coefficient of interest $\beta_{(0)}$, we assume approximate sparsity, i.e., in fact only a few s_y (s_d) of the other p controls $x_{i,t}$ and $z_{i,t}$ are relevant for each state in the equation of y (d). We start from the reduced form of the main equation by plugging (4.7) into (4.6)

$$\ddot{y}_{i,t} = \phi^{\mathsf{T}}\begin{pmatrix}\ddot{x}_{i,t}\\ \ddot{z}_{i,t}\end{pmatrix} + \ddot{\eta}_{i,t} \ , \tag{4.8}$$

with $\phi = \beta_{(1)} + \beta_{(0)}\beta_{(2)}$ and $\ddot{\eta}_{i,t} = \ddot{\epsilon}^{(1)}_{i,t} + \beta_{(0)}\ddot{\epsilon}^{(2)}_{i,t}$. We use the Lasso (Tibshirani, 1996) as a data-driven tool to select the respective relevant covariates from an ℓ_1 penalized minimization problem. We obtain the Lasso estimates $\hat{\beta}_{(1)}, \hat{\beta}_{(2)}$ as

$$\hat{\beta}_{(1)} = \arg\min_{\phi} \frac{1}{2nT} \sum_{i=1}^{n} \sum_{t=1}^{T} \left[\ddot{y}_{i,t} - \phi^{\mathsf{T}} \begin{pmatrix} \ddot{x}_{i,t} \\ \ddot{z}_{i,t} \end{pmatrix} \right]^2 + \lambda_1 \sum_{j=1}^{p} |\phi^{(j)}| \ , \tag{4.9}$$

$$\hat{\beta}_{(2)} = \arg\min_{\beta_{(2)}} \frac{1}{2nT} \sum_{i=1}^{n} \sum_{t=1}^{T} \left[\ddot{d}_{i,t} - \beta^{\mathsf{T}}_{(2)} \begin{pmatrix} \ddot{x}_{i,t} \\ \ddot{z}_{i,t} \end{pmatrix} \right]^2 + \lambda_2 \sum_{j=1}^{p} |\beta^{(j)}_{(2)}| \ , \tag{4.10}$$

with regularization parameters $\lambda_1, \lambda_2 \geq 0$ that are estimated by cross-validation and $\phi = (\phi^{(1)}, \ldots, \phi^{(p)})^{\mathsf{T}}$ [12]. Note that we use the reduced form of the main equation (4.8) and therefore implicitly penalize the treatment also in (4.9). We use the lasso as a model selection device in both equations, where we denote the index set of selected covariates for (4.6) by S_y and for (4.7) by S_d. The causal effect can than be obtained from the post-selection equation using a union of both selected controls

$$\ddot{y}_{i,t} = \beta_{(0)}\ddot{d}_{i,t} + \tilde{\beta}^{\mathsf{T}}_{(1)} \begin{pmatrix} \ddot{x}^{S}_{i,t} \\ \ddot{z}^{S}_{i,t} \end{pmatrix} + \ddot{\epsilon}^{(1)}_{i,t} \ , \tag{4.11}$$

where $S = \hat{S}_y \cup \hat{S}_d \subseteq \{1, 2, \ldots, p\}$, and $\ddot{x}^{S}_{i,t}$, $\ddot{z}^{S}_{i,t}$ only contain elements of S. Note that the post-selection estimation in (4.11) is necessary in order to mitigate estimation biases from the penalized selection equations.

Instead of determining $\hat{S}_y$ and $\hat{S}_d$ as index set of elements in $(\ddot{x}_{i,t}\ddot{z}_{i,t})$ with non-zero $\hat{\beta}_{(1)}$ or $\hat{\beta}_{(2)}$ directly from (4.9) and (4.10)(see Belloni et al. (2014b)), we suggest a subsampling-based stability selection. We demonstrate in the Section 4.4 that this methodology also works for strongly correlated variables with measurement issues using the ideas and features of stability selection (Meinshausen and Bühlmann, 2010) in the Lasso selection steps (4.9) and (4.10). The procedure works as follows:

1. Generate C subsamples c of size n^* of the nT data points and obtain C estimates $\hat{\beta}^{(j)}_{(1,c)}$ and $\hat{\beta}^{(j)}_{(2,c)}$, $c = 1, \ldots, C$ for each coefficient $j = 1, \ldots, p$ in (4.9) and (4.10).

[12] In practice, there exist several techniques for solving this problem, while we use coordinate-descent algorithms (Friedman et al., 2007, 2010) provided in the *glmnet* package in R.

2. Compute for each variable j the relative inclusion frequencies $\hat{\Pi}_j^1 = \frac{1}{C}\sum_{c=1}^{1000} \mathbb{1}_{\{\hat{\beta}_{(1,c)}^{(j)} \neq 0\}}$ and $\hat{\Pi}_j^2 = \frac{1}{C}\sum_{c=1}^{1000} \mathbb{1}_{\{\hat{\beta}_{(2,c)}^{(j)} \neq 0\}}$.

3. Only include variable j in the model and thus in S if $\hat{\Pi}_j^1 > \pi_1$ or $\hat{\Pi}_j^2 > \pi_2$.

Note that as in Belloni et al. (2014a,b), S consists of variables either influencing the treatment $d_{i,t}$ or the response $y_{i,t}$. Hence the selection choice from the auxiliary equation (4.10) corrects wrong de-selection choices in the main enrollment equation (4.9) due to highly correlated control variables. In this sense it provides a robustification of the selection against underspecification and resulting biased estimates by double selection. In contrast to direct lasso in both selection equations, however, the proposed procedure reduces the risk of overspecification by the stability selection sub-sampling step. Typically, the index set S of the stability double selection is a subset of the standard double selected set and depends on the choice of sufficiently large π_1 and π_2 and the number of repetitions C. The stability post-double selection procedure yields a consistent $\beta_{(0)}$-estimator from (4.11), see (Belloni et al., 2014a; Meinshausen and Bühlmann, 2010). In contrast to standard lasso double selection, it also shows excellent finite sample performance in particular in settings with a very strong correlation of control variables in combination with single influential observations as in our data (see simulation study in Section 4.4).

For the empirical results and the simulation, we generally use $C = 1000$ and $n^* = 0.5nT$ in the algorithm above.[13] For a data-driven threshold choice, we set minimum thresholds $\pi_{1,\theta}^{min}, \pi_2^{min} > 0.9$ as lower bounds ensuring that we screen out irrelevant variables. Since the response values change with θ in (4.9), the corresponding minimum thresholds also depend on θ. The selection of effective thresholds is then performed over a grid of threshold values starting from the minima increasing the threshold level to the first points where small changes in the thresholds do no longer change the model. The algorithm for the threshold choice can be found in Appendix 4.7.2. In the simulation, we also report estimates with $\pi_{1,\theta}^{min} = \pi_2^{min} = 0.5$ and 0.7 for comparison. Moreover, for all statistical

[13] For the robustness checks using only the control year 2008 and 2014, we increase the subsample to $n^* = 0.8nT$ to deal with the small data set.

inference, we use the usual degrees of freedom (df) correction for fixed effects panel models.[14]

4.4 Simulation

We conduct a Monte-Carlo Simulation to show the importance of stability selection when it is hard to disentangle effects of different covariates. This can be further adapted to our data by including influential observations and by inducing strong correlation among covariates. Using $i = 1, \dots, n$, $t = 1, \dots, T$, and $g = 1, \dots, p$ with $T = 10$, $n = 16$, $N = nT$, and $p = 30$, we simulate a linear panel model of the following form[15]:

$$\begin{aligned} \tilde{y}_{i,t} &= \eta_0 d_{i,t} + \eta_1 \tilde{x}_{i,t} + \alpha_i + \sigma_1(d_{i,t}, x_{i,t})\epsilon_{i,t}^{(1)} \ , \\ d_{i,t} &= \eta_2 \tilde{x}_{i,t} + \sigma_2(x_{i,t})\epsilon_{i,t}^{(2)} \ , \end{aligned}$$

with coefficients depending on g: $\eta_0 = 0.5$, $\eta_1^{(g)} = \frac{5}{g}\mathbb{1}_{\{g \leq 10\}}$, and $\eta_2^{(g)} = \frac{5}{g-6}\mathbb{1}_{\{7 \leq g \leq 10\}}$ for $g \neq 6$, zero otherwise. The coefficients of covariates are up to 10 times higher than the coefficient of the treatment, since such large differences are also likely to arrive in our empirical application, where the expected treatment effect is relatively small. We generate the fixed effects[16] as $\alpha_i \sim \mathcal{N}(0, \sqrt{\frac{4}{T}})$ and $x_{i,t} \sim \mathcal{N}(0, \Sigma)$[17], with $\Sigma_{v,w} = 0.5^{|w-v|}$, v representing the rows and w the columns of Σ, $v \neq w$. For $v = w = 1, \dots, 10$, $\Sigma_{v,w} = 2$, and for $v = w = 11, \dots, 30$, $\Sigma_{v,w} = 6$. The errors are independently distributed as $\epsilon_{i,t}^{(1)} \sim \mathcal{N}(0,1)$ and $\epsilon_{i,t}^{(2)} \sim \mathcal{N}(0,1)$ with a heteroskedastic structure given by

$$\sigma_1(d_{i,t}, x_{i,t}) = \sqrt{\frac{(1 + \eta_0 d_{i,t} + \eta_1 x_{i,t} + \alpha_i)^2}{\mathbb{E}_N[(1 + \eta_0 d_{i,t} + \eta_1 x_{i,t} + \alpha_i)^2]}}, \ \sigma_2(x_{i,t}) = \sqrt{\frac{(1 + \eta_2 x_{i,t})^2}{\mathbb{E}_N[(1 + \eta_2 x_{i,t})^2]}} \ .$$

[14] The df of the residuals reduce from $df = nT - |S|$ to $df = n(T-1) - |S|$, which is due to the demeaning process. For each observation i, one degree of freedom is lost because of the error term $\epsilon_{i,t}$. The latter is now comparable to a parameter that needs to be estimated (see Wooldridge (2002)).

[15] This setup is similar the simulation in Belloni et al. (2014b) Belloni et al. (2016) but adapted to fit more closely to our application with highly influential observations and strong correlation among covariates.

[16] Note that we omit using a fixed effect in the second equation of this data generating process since using the demeaning framework here, such an effect disappears algebraically.

[17] $x_{i,t} = (x_{i,t}^{(1)}, \dots, x_{i,t}^{(g)}, \dots, x_{i,t}^{(p)})^\intercal$: for $g, k = 1, \dots, p$, $x_{i,t}^{(g)}$ represents a covariate that is standard normal with a correlation of $\rho = 0.5^k$ to $x_{i,t}^{(g+k)}$ and $x_{i,t}^{(g-k)}$, $1 \leq g - k \leq g + k \leq p$.

Given this structure, we distort the last 10% of observations by a vector $\gamma = (\gamma_1, \ldots, \gamma_p)^\mathsf{T}$ that is uniformly distributed with regard to a strength *inf*, where $inf = 0$ represents no distortion. We report mean values over 1000 replications for the absolute bias of estimators, the RMSE, the number of selected covariates, the true positive rate (TPR), and the false positive rate (FPR). More details on the distortion and the evaluation measures can be found in Appendix 4.7.4. We also report the rejection rate, which is based on conventional t-tests on the estimated $\hat{\eta}_0$ against the true η_0. For the t-tests and the $RMSE_{\eta_0}$, we generally use the classical heteroskedasticity consistent HC3 standard errors (MacKinnon and White, 1985). For comparison, we have also calculated design-based (Abadie et al. (2020)) and clustered SEs. The design-based SEs reflect that since the full population rather than a subsample is observed, uncertainty about the treatment effect does not result from sampling, but from uncertainty about the unobserved counterfactual. Details on the calculation can be found in Appendix 4.7.3 where we use that the post-double selection outcome and auxiliary equations are both linear.[18] Results for these design-based SEs are found in Table 4.8 in the Appendix and only differ considering the rejection rates and the $RMSE_{\eta_0}$, where the classical HC3 SEs are more conservative, resulting in smaller rejection rates and larger $RMSE_{\eta_0}$-values than their design-based counterparts.[19]

We report results from post-Lasso and post-double selection[20] as described in Section 4.3, using no subsampling at all and using the subsampling similar to stability selection with $\pi_{min} \in \{0.5,\ 0.7\}$. Additionally, we report the two extreme cases using all covariates without selection (Fixed Effects all) and using only the true influencing variables (Oracle).

Table 4.1 summarizes our simulation results. First of all, as expected, the proposed double selection procedure combined with stability selection performs best overall and is almost identical to the oracle procedure that knows the true active set. Using of $\pi_{min} = 0.7$ or $\pi_{min} = 0.5$ does not affect results much in most cases. The non-stable

[18] We can thus employ Abadie et al. (2020), Assumption 8 and Theorem 1. in the demeaned equation.

[19] The results for clustered SE are similar and available on request.

[20] We also ran a scenario using the cluster-robust penalty loadings of Belloni et al. (2016) for the lasso regression which did not yield superior results. We therefore do not report these additional calculation in the simulation and results section. Calculations are available from authors upon request.

Table 4.1: Simulation Results for Different Forms and Strengths of Influential Observations With HC3 Standard Errors

	Absolute $Bias_{\eta_0}$			$RMSE_{\eta_0}$			# Covariates			TPR			FPR			Rejection Rate		
Size of Distortion:	0	1	5	0	1	5	0	1	5	0	1	5	0	1	5	0	1	5
Distortion in the Active Set																		
PL Stab: 0.5	0.156	0.158	0.181	0.025	0.026	0.034	9.238	9.338	7.454	0.837	0.839	0.734	0.043	0.047	0.006	0.999	0.994	1.000
DB Stab: 0.5	0.088	0.087	0.081	0.024	0.024	0.021	10.669	10.629	10.099	1.000	1.000	1.000	0.034	0.032	0.005	0.068	0.062	0.058
PL Stab: 0.7	0.161	0.164	0.192	0.027	0.028	0.038	8.360	8.344	6.808	0.802	0.801	0.679	0.017	0.017	0.001	1.000	1.000	1.000
DB Stab: 0.7	0.087	0.086	0.081	0.024	0.023	0.021	10.190	10.199	10.021	1.000	1.000	1.000	0.009	0.010	0.001	0.066	0.059	0.056
Post Lasso	0.143	0.143	0.131	0.024	0.024	0.025	19.395	19.156	14.289	0.917	0.917	0.912	0.511	0.499	0.259	0.842	0.839	0.673
Double Selection	0.090	0.092	0.086	0.025	0.026	0.023	21.230	20.703	16.395	1.000	1.000	1.000	0.561	0.535	0.320	0.065	0.065	0.061
Fixed Effects All	0.092	0.094	0.089	0.028	0.028	0.026	30.000	30.000	30.000	1.000	1.000	1.000	1.000	1.000	1.000	0.051	0.058	0.045
Oracle	0.086	0.086	0.081	0.024	0.023	0.021	10.000	10.000	10.000	1.000	1.000	1.000	0.000	0.000	0.000	0.068	0.060	0.056
Distortion in the Inactive Set																		
PL Stab: 0.5	0.156	0.157	0.157	0.025	0.026	0.026	9.238	9.435	9.221	0.837	0.840	0.842	0.043	0.052	0.040	0.999	0.994	0.996
DB Stab: 0.5	0.088	0.087	0.087	0.024	0.024	0.024	10.669	10.750	10.585	1.000	1.000	1.000	0.034	0.037	0.029	0.068	0.066	0.066
PL Stab: 0.7	0.161	0.164	0.162	0.027	0.028	0.027	8.360	8.354	8.306	0.802	0.801	0.802	0.017	0.017	0.014	1.000	1.000	1.000
DB Stab: 0.7	0.087	0.086	0.086	0.024	0.023	0.023	10.190	10.251	10.153	1.000	1.000	1.000	0.009	0.013	0.008	0.066	0.061	0.061
Post Lasso	0.143	0.144	0.144	0.024	0.025	0.025	19.395	19.275	18.535	0.917	0.918	0.918	0.511	0.505	0.468	0.842	0.834	0.840
Double Selection	0.090	0.092	0.091	0.025	0.026	0.026	21.230	21.122	20.548	1.000	1.000	1.000	0.561	0.556	0.527	0.065	0.067	0.072
Fixed Effects All	0.092	0.094	0.094	0.028	0.028	0.028	30.000	30.000	30.000	1.000	1.000	1.000	1.000	1.000	1.000	0.051	0.062	0.061
Oracle	0.086	0.087	0.087	0.024	0.023	0.023	10.000	10.000	10.000	1.000	1.000	1.000	0.000	0.000	0.000	0.068	0.061	0.061
Distortion in the Response																		
PL Stab: 0.5	0.156	0.158	0.169	0.025	0.026	0.030	9.238	9.306	8.537	0.837	0.836	0.769	0.043	0.047	0.042	0.999	0.998	0.999
DB Stab: 0.5	0.088	0.088	0.107	0.024	0.024	0.036	10.669	10.788	10.777	1.000	1.000	1.000	0.034	0.039	0.039	0.068	0.062	0.065
PL Stab: 0.7	0.161	0.164	0.176	0.027	0.028	0.032	8.360	8.330	7.550	0.802	0.797	0.726	0.017	0.018	0.015	1.000	1.000	1.000
DB Stab: 0.7	0.087	0.088	0.106	0.024	0.024	0.035	10.190	10.237	10.238	1.000	1.000	1.000	0.009	0.012	0.012	0.066	0.063	0.054
Post Lasso	0.143	0.143	0.149	0.024	0.025	0.029	19.395	19.361	19.019	0.917	0.917	0.908	0.511	0.509	0.497	0.842	0.834	0.817
Double Selection	0.090	0.092	0.111	0.025	0.026	0.039	21.230	21.153	21.392	1.000	1.000	1.000	0.561	0.558	0.570	0.065	0.069	0.053
Fixed Effects All	0.092	0.095	0.114	0.028	0.028	0.043	30.000	30.000	30.000	1.000	1.000	1.000	1.000	1.000	1.000	0.051	0.052	0.042
Oracle	0.086	0.087	0.106	0.024	0.024	0.035	10.000	10.000	10.000	1.000	1.000	1.000	0.000	0.000	0.000	0.068	0.061	0.051

Notes: All values are based on Monte Carlo simulations with 1000 runs and 1000 repeated subsample steps ($C = 1000$). Rejection rates are based on t-tests with heteroskedasticity consistent standard errors (HC3, see MacKinnon and White (1985)). The remaining measures are means over the 1000 replication runs. PL Stab and DB Stab stand for post-Lasso and double selection with stability selection and the corresponding minum thresholds π_{min}. Oracle is similar to Fixed Effects All but using only true influencing covariates. *inf* indictates the strength of influential observations and is reported for each measure, while the form of influence (active/inactive set and response) is depicted in the rows.

versions often include up to twice as many covariates without much improvement on the TPR, but high increases in the FPR.

Taking a closer look at the different forms of distortion, we do not observe much change for high *inf*-values when we distort variables from the inactive set. As expected, when influential observations are only present in the noise variables, they do not affect the selection procedures much. When distorting the active set only, however, procedures with the post-Lasso select fewer (relevant) variables due to the added noise, which leads to a higher bias (for the stability cases), and increases *RMSE* values. The double selection procedures seem to be very robust against such distortions, with all measures remaining relatively unchanged. This is not surprising, since the double selection procedure helps to reduce such a bias by taking the second equation into account. Finally, distorting the response is interesting, since both relevant and irrelevant covariates are affected at the same time. Even with extremely high distortions, the double selection procedures keep a lower bias compared to the other methods and double selection with stability selection has very low FPRs, while selecting almost all variables from the active set. All in all, the simulation shows that only when we use stability selection, we can select the right variables without including too many noise variables. In our simulated model, where it is hard to distinguish between covariates and the treatment effect is relatively small compared to the effects of other covariates, the non-stable methods perform worse over all distortion scenarios[21]. Furthermore, we see that when some covariates explain the treatment well, but only have a moderate effect on the response (which is the case in the application), double selection outperforms the post-Lasso in terms of bias and rejection rate.

4.5 Empirical Results

4.5.1 Main Findings

In this section, we present the results of our empirical study. Generally, with only publicly available data and the proposed post stability double selection methodology, we find that

[21] Results are similar using a lower correlation among covariates. Additional simulations are available upon request.

tuition fees in Germany significantly reduced the enrollment rate by 3.8pp to up to 4.5pp on average over all possible cases of response variables. For all admissible values of θ, the procedure consistently identifies the same one university specific and one educational policy change control variable in x and the four spatial variables z as important drivers highlighting the importance of fee induced migration effects. Moreover, we find that during the considered period, other socio-economic factors only played a minor role. Given the transparency in θ and the data-driven stability double selection, we judge these findings are very robust.

Table 4.2: Estimates of the Causal Effect of Tuition Fees $\beta_{(0)}$ for Different θ-Values

	Double Selection + Stability			All Controls
Effects on $y_{i,t}$	$\theta = 0.98$	$\theta^* = 0.9927$	$\theta = 1$	$\theta^* = 0.9927$
Tuition Fees	**−4.310**	**−3.996**	**−3.808**	**−1.267**
(HC3)	*(1.243)*	*(1.372)*	*(1.593)*	*(1.229)*
(Design-based)	*(1.177)*	*(1.278)*	*(1.486)*	*(0.989)*
Student.to.researcher.ratio	−2.763	−2.931	−3.286	0.887
Double.Cohort	−1.732	−2.766	/	−6.294
Migration.neighbor.fees	/	46.443	86.812	31.254
Migration.rest.fees	59.847	83.003	134.509	71.765
Migration.international	−3.958	15.621	35.069	−21.623
Migration.no.fees	22.956	46.713	77.884	30.678
⋮	/	/	/	⋮

Notes: Response values are scaled to a percentage level. Standard errors in parentheses are calculated based on heteroskedasticity consistent infinite populations (HC3, see MacKinnon and White (1985)) or on treatment design and finite populations (Abadie et al. (2020)). Variables in blue appeared similarly in previous studies (not necessarily together).

Table 4.2 summarizes the post-selection estimation results. Most importantly, we find a significant negative causal effect over the whole grid of θ-values only when using post-double selection with repeated subsampling (Double Selection + Stability). The reference point $\theta^* = 0.9927$ from additional non-public information in (4.13) suggests in

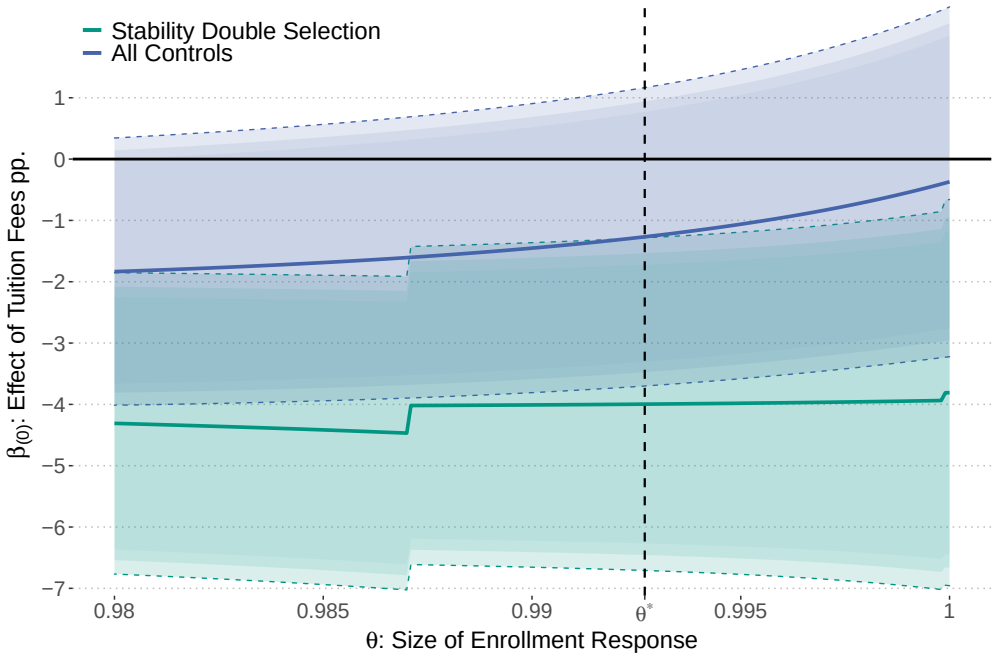


Figure 4.2: Estimates for the Causal Effect β_0 in (4.4) for Stability Double Selection

Notes: All effects are plotted over the grid of admissible θ in $AHG_{j,i,t}$ from (4.3)." All Controls" describes a linear fixed effects regression using all controls. We depict 95%, 92.5% and 90% CIs in shaded colors, which are calculated on HC3 standard errors (see MacKinnon and White (1985)).

fact that values very close to the right boundary of $\theta = 1$ are the most plausible, i.e. the number of effective enrollments of migrating students from j to i within Germany almost coincides with the number of potentially enrolling ones $EHG_{i,t}$ at θ^*. For such large θ-values in particular, using all controls in a plain panel OLS clearly underestimates the effect and thus leads to inflated p-values, which is illustrated in Figure 4.2[22]. Post-double selection Lasso without the stabilizing subsampling does not work as it leads to the same results as a pooled OLS with all controls. In those cases, the magnitude of the effect from tuition fees is roughly four times smaller than for the post stability double selection and the impact becomes insignificant. Across all admissible θ, only about a third of the controls are selected with our proposed procedure, which indicates that many plausible

[22]See Figure 4.10 in the Appendix for design-based errors.

controlling factors from the literature are in fact not relevant and dominated in this period of heterogeneous changes in educational policies across states.

Since the spatial variables are both among the most selected and have a high power in explaining variance, it is important to discuss two possible sources of endogeneity that are, however, not relevant in this case. First of all, the spatial variables are related to the outcome of enrollments, since they measure *migrations* from other states $-i$ relative on the total number of enrollments in state i. In our case, since we look at migration simultaneously to enrollment, we can rule out reverse causality that might be an issue if we looked at migration behavior in later years relative to enrollment behavior. Since this is not the case here, there is no reason why relative enrollment numbers would drive migration behavior in the same year, since it is simply not observed at the time of decision. Secondly, the spatial variables are related to the treatment and location. This split into two variables does not cause an endogeneity issue since it just serves to disentangle the effect in states that neighbor state i and have (not) implemented tuition fees vs. other states.

Looking more closely at Figure 4.2, we see that over the entire grid of admissible θ-values, only the double selection procedure with subsampling guarantees good performance, whereas with all controls the estimated effect for β_0 vanishes with θ approaching the upper bound 1. With an effect of tuition fees close to zero for the upper θ-boundary, and only half the size of the one by the stable double selection at the lower θ-boundary, the pooled OLS appears biased in detecting individual influences in this situation, where observations are scarce relative to the dimension of the model. This behavior is not surprising, as many irrelevant controlling factors that might be spuriously correlated with the response and the treatment are present without selection. This is more critical at the upper θ-boundary, where the variability of the response is higher. Furthermore, using the post-Lasso, even with stability selection, gives less stable and often insignificant results. The insignificance can be traced back to the lack of additional controls that are only added in the second step of the double selection procedure, whereas the rather unstable results can furthermore be accounted for by the difference in the selection procedure in the first step that includes the treatment in the equation. All this emphasizes the importance of using a post stability double selection as proposed.

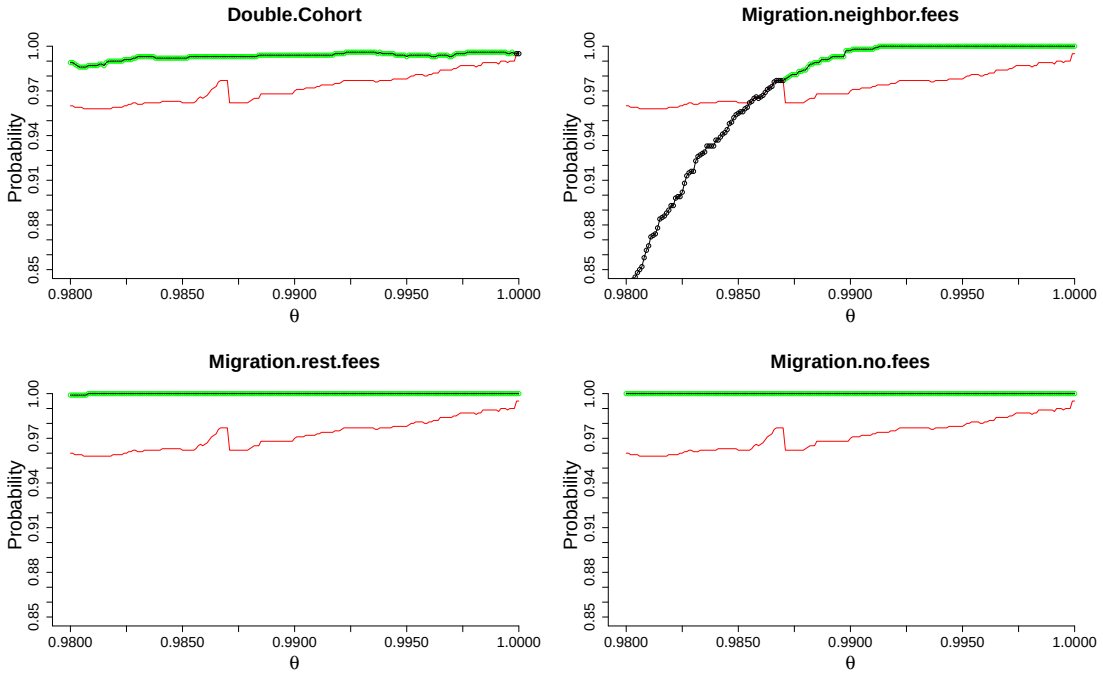


Figure 4.3: Controls With High Inclusion Probabilities in the First Step Depending on θ (X-Axis)

Green indicates that the respective variable is selected in the final model, which depends on the threshold that is depicted as a red line. The y-axis shows the selection probability from the Lasso step.

Figure 4.3 shows all controls that were selected in the main equation (4.9) (i.e. with $y_{i,t}$ as the dependent variable). We find the spatial variables to be highly relevant, which implies that mobility and migration effects played a major role for enrollments in the presence of heterogeneous timing and implementation of tuition fees and major educational policy decisions across states. In size, they largely contribute in explaining the variability of the enrollment rates. At the lower boundary of θ, only one of the four spatial variables *Migration.neighbor.fees* is included less often over different subsamples and is thus deselected by the stability selection for low θ-values. As there is only a small limited number of overall neighbors of each state, their impact on enrollments in state i is generally much smaller as from the aggregated rest of the country and thus more sensitive to a variation in the response variable.

Furthermore, the variable *Double.cohort* that indicates if there were two cohorts of high school students graduating in the same year, caused by the G8 reform reducing time to graduation, is identified as an important controlling factor. *Double.Cohort* has a negative sign, which at first might appear counter-intuitive, as with a double cohort, one would expect enrollment numbers of students to rise. For relative enrollment rates, however, a negative sign of double cohort seems justified, since universities did not double their admission numbers when there was a double cohort. Moreover, when the competition for universities is extremely high in a double cohort situation, fewer people might decide to actually compete and rather consider outside options or postpone university entrance with a gap year. Note that for the extreme boundary case ($\theta > 0.9998$), however, the variable is deselected, which can be attributed to the pre-dominance of the migration factors with large size effects at the extreme upper θ-boundary. Repeating the analysis with *Double.Cohort* in the extreme case for $\theta > 0.9998$, however, does not change results and only alters coefficient values in an minor insignificant way. This behavior can be expected when taking into account that the effect of *Double.Cohort* is relatively small compared to the other variables close to the upper boundary of θ.

In line with theory, the variables *Student.to.researcher.ratio* and the share of international enrollments *Migration.international* that are additionally selected in the auxiliary equation of the double selection procedure only have a minor direct influence on enrollment rates, while having a large impact on tuition fees. Thus, this socio-economic factor and the financial situation of universities drives the political decision for the introduction of fees. Overall, the double selection step is key yielding additional necessary variables for accurate estimation of β_0 (see Figure 4.2).

Generally, these findings show that spatial factors and the double cohort variable are crucial for identifying the effect of tuition fees on enrollments. In the existing empirical literature, however, they have been largely ignored yielding downward biased insignificant estimates. Moreover, the auxiliary equation and the stability double selection are key for detecting the magnitude β_0.

4.5.2 Robustness Checks

Apart from using all available data, we also analyze two subsets that either contain only periods with tuition fees (2006-2013) or that consist of the peak year 2008 of the presence of tuition fees and the year 2014 after their abolishment. Furthermore, we work with the alternative response variable $y_{i,t}^{extra} = \frac{NE_{i,t}}{EHG_{i,t}^*}$ constructed from additional non-public information in the eligible set $EHG_{i,t}^*$ in Equation (4.13) in the Appendix. Estimates of $\beta_{(0)}$ for these adaptations are summarized in Table 4.3. Results for design-based errors do not differ substantially, but are slightly less conservative and can be found in Table 4.9 in Appendix 4.7.5.

First, when comparing the effect with θ^*-response values over different time frames, we find that the main results prevail over the variation in the data set. The double selection is still the only reliable method, while post-Lasso and pooled OLS with all controls cannot capture the strength of the effect nor its statistical significance persistently. Post-Lasso generally de-selects too many relevant controls, yielding smaller effects in absolute values of tuition fees on enrollments. Omitting the first and last year from the data only causes mild changes in the amount of included controls, but the size of the estimate for β_0 from double selection decreases in absolute terms, probably due to fewer available observations. Though, in the extreme case of the smallest data set, where only two years with either "no fees at all" or "fees in seven states" are considered, the magnitude of the effect increases substantially. The results of the extra response $y_{i,t}^{extra}$ confirm the above observations. The size of the estimates for β_0 for different time frames and the amount of included controls mostly coincide with results for the response y_{θ^*}. In this case, however, the pure post Lasso stability selection estimate is much closer to the estimate of the stability double selection procedure in size and becomes even mildly significant.

In summary, we conclude that the effect is rather robust to changes of the time frame and double selection consistently identifies the effect, where the other methods mostly fail. While changes in the strength of the effect arise mostly in very high-dimensional situations (i.e. small data set), the effect is also identified using the additionally constructed $y_{i,t}^{extra}$. Comparing the strength of the effect to previous studies, which estimated (mostly insignificant) effects from -0.4pp to -2.69pp, we see that for almost all cases, our estimated effect lies rather between -3 and -4pp using double selection, and is always

Table 4.3: Estimates of the Causal Effect of Tuition Fees $\beta_{(0)}$ for θ^* in Different Time Frames With HC3 Standard Errors

	Data sets			No. of Variables
Tuition Fees	All	Fees	Small	All/Fees/Small
min MSD with θ^*:	*0.9927*	*0.9924*	*0.9934*	
All Controls	−1.267	−1.952	-	19/19/-
	(1.229)	(1.454)		
Post-Lasso Stability	−2.538	−2.599	−6.345**	4/4/3
	(1.354)	(1.358)	(1.698)	
Double Selection Stability	−3.996**	−3.180*	−16.468***	7/6/7
	(1.372)	(1.388)	(3.269)	
min MAD with θ^*:	*0.9927*	*0.9926*	*0.9945*	
All Controls	−1.267	−1.941	-	19/19/-
	(1.229)	(1.456)		
Post-Lasso Stability	−2.538	−2.599	−6.126**	4/4/3
	(1.354)	(1.363)	(1.745)	
Double Selection Stability	−3.996**	−3.185*	−17.133***	7/6/7
	(1.372)	(1.392)	(3.349)	
$y_{i,t}^{extra}$ with π_1/π_2:	*0.999/0.9*	*0.9/0.9*	*0.85/0.91*	
All Controls	−1.722	−2.213*	-	19/19/-
	(0.922)	(1.087)		
Post-Lasso Stability	−3.349*	−2.234*	−11.570***	3/9/2
	(1.369)	(0.942)	(1.519)	
Double Selection Stability	−3.920***	−2.198*	−15.021**	6/10/6
	(1.151)	(1.000)	(4.415)	

Notes: Response values are scaled to a percentage level. Standard errors in parentheses are heteroskedasticity consistent (HC3, see MacKinnon and White (1985)). *p<0.05; **p<0.01; ***p<0.001 indicate p-values from a t-test on significance from zero. θ^* is chosen according to minimum mean squared deviation (MSD) and minimum mean absolute deviation (MAD).

highly significant. On the contrary, using fixed effects with all controls and without selection yields estimates that appear to be downwards biased and closer to the lower bound found in other studies, while in almost all cases, this cannot identify significant effects.

4.6 Conclusion

In this article, we propose a stabilized double selection technique in order to identify the effect of tuition fees on enrollment rates from public state-level data in Germany. We show that such techniques are key for extracting size and significance of the causal effect for the special German situation. In this setting, where few observations coincide with varying implementation and timing of tuition fees and other educational policies across states and time, we are facing correlated covariates and influential observations, which require carefully chosen, tailored econometric techniques.

With our tailored post-Lasso approach, we are the first to find an overall significant negative effect of tuition fees in Germany. With the stability double selection we identify the relevant factors, which are crucial for political decision-making. In particular, previously neglected spatial migration effects and the major shift in educational policy by the G8 high school reform appear as key control variables for enrollment rates in the considered period. The detected effect is robust over a large grid of different response values and different subsets of the full data set. These empirical findings therefore contribute to the existing literature on education economics. In the active ongoing discussion about the reintroduction of tuition fees in Germany, the results might also be of political interest.

Moreover, this study strongly advocates the use of data-driven variable selection to choose relevant controls from a broad set of possible influencing factors. We explicitly show that standard fixed effects panel regressions without selecting variables fails to detect correct and precise effects for such small sample sizes relative to the dimensionality of the problem. Furthermore, appropriate statistical selection techniques determine and justify the relevance of chosen controlling factors, yielding an easily interpretable post-selection model that outperforms all ad-hoc choices. For future research, it would

be interesting to use the data-driven identification of relevant controls also for other countries, e.g. the United Kingdom or France, aiming for a comprehensive European study with increasingly relevant spatial cross-effects across country borders. This is particularly relevant given the reintroduction of fees for international students in parts of Germany, that could trigger such cross-effects.

4.7 Appendix

4.7.1 Data

Table 4.4: Description of Regression Variables and Socio-Economic Control Variables

Variable	Description	Source
Regression Variables		
$y_{i,t}$	Enrollment rate: new first year students in state i and winter term of year $t/t+1$ divided by affected graduates. Affected graduates consist of high school graduates in state i and year t, international first year students in state i and winter term of year $t/t+1$ and a weighted number of in-country migration from other German states ($\sum_{j \neq i} EHG_{j,i,t}$). $EHG_{j,i,t}$ is calculated using convex combinations of $NE_{i,t}^{(j)}$ and $HG_{j,t}$.	Own calculation from Federal Statistical Office (2014b): TAB-13 and Federal Statistical Office (2014d): TAB-06
$y_{i,t}^{extra}$	Similar to $y_{i,t}$, but using a different measurement for $EHG_{j,i,t}$ (see Section 4.2).	Own calculation from Federal Statistical Office (2014b): TAB-13 and Federal Statistical Office (2014d): TAB-06
$d_{i,t}$: Tuition.Fees	1 if tuition fees were present in state i in winter term of year $t/t+1$, zero otherwise	Mitze et al. (2015)
Socio-Economic Statistics		
log.Rent	Natural logarithm of average rent in households of state i in year t excluding heating or extra costs	SOEP: Hgen, Hgrent
log.Income	Natural logarithm of average income in households of state i in year t	SOEP :Hgen, Hghinc
Urbanization.level	Share of households living in cities in state i in year t	SOEP: Hbrutto, Regtyp
Life.Satisfaction	Average life satisfaction per person (0=Completely dissatisfied, 1= Completely satisfied) in state i	SOEP: pequiv, P11101
Unemployment.Rate	Unemployment rate in state i (0=0%, 1=100%)	Genesis
G8	1 if students graduated high school in 8 years in state i in year t, zero otherwise	Own research
Double.Cohort	1 if there was a double cohort of students graduating high school in state i in year t, zero otherwise	Own research
Mil.Service	1 if there was mandatory military service for male high school graduates in Germany in year t, zero otherwise	`https://www.gesetze-im-internet.de/wehrpflg/__2.html` (German)

Notes: All data sets used to obtain the variables can be found in the SOEP-database at `https://www.diw.de/en/soep` with the corresponding variable description and sample (SOEP: sample, variable) at `https://paneldata.org/soep-long`. For the Genesis Data, the tables are found at `https://www-genesis.destatis.de` in the menu under "Available Data" on the page "Tables". There, a search for a specific coding leads to the desired tables. The data is gathered by choosing the respective year.

Table 4.5: Description of University and Student Control Variables

Variable	Description	Source
Control Variables regarding Students and Universities		
Log.Third.Party.Funds.per.institution	Natural Logarithm of the quotient of third party funds for universities in state i in year t divided by the number of state accredited higher institutions, aggregated at a state level	Federal Statistical Office (2014a): 2.1.3; `www.hochschulkompass.de`
Log.Spendings.per.Student	Natural Logarithm of spendings of state i in year t per student, aggregated over all universities i	Federal Statistical Office (2014a): 1.1
Student.to.researcher.ratio	Number of students in state i in year t per scientific employee of higher institutions	Federal Statistical Office (2014c): ZUS-01
Habilitations	Share of habilitations at universities in state i to habilitations over all states	Federal Statistical Office (2014c): ZUS-07
Graduates	Share of graduates at universities in state i to relevant population	Federal Statistical Office (2014b): TAB-02
Women.Studying	Share of female students studying at higher institutions to all students	Genesis: Table 21311-0014

Notes: Data from the Federal Statistical Office are availible in the ".xls" format in German in the respective sheet (indicated by TAB or ZUS). They can be found at the bottom of the page `https://www.destatis.de/DE/ZahlenFakten/GesellschaftStaat/BildungForschungKultur/Hochschulen/Hochschulen.html` under "Ausgewählte Publikationen", using the name of the report and the reference. Reports from earlier years are in the report of the respective year. For the Genesis Data, the tables are found at `https://www-genesis.destatis.de` in the menu under "Available Data" on the page "Tables". There, a search for a specific coding leads to the desired tables. The data is gathered by choosing the according year.

Individual SOEP Data

Figures 4.4 and 4.5 give an overview on the lack of observations of individuals in the SOEP data set. On the x-axis, the number of observations for each state-year tuple is shown, whereas on the y-axis, the frequency of tuples with the specific number of observations is depicted. As can be seen in the histograms, there were many tuples with insufficient number of observations to represent a state. More specifically, 109 tuples out of 160 have less than 20 observations each (i.e. individuals) in Figure 4.4 (i.e. eligible high school graduates in state i and year t) and 119 tuples out of 160 have less than 10 observations each (i.e. individuals) in Figure 4.5 (i.e. first year students in state i and

Table 4.6: Description of Spatial Control Variables

Variable	Description	Source
Spatial Control Variables		
Migration.neighbor.fees	Enrolling students to state i in year t with high school diploma from fee neighbor states (i.e. states that share a border with i and that have tuition fees in the winter term of year t) minus enrollments to fee neighbor states by students with high school diploma from state i, both divided by all new enrollments in state i.	Federal Statistical Office (2014b): TAB-13
Migration.rest.fees	Enrolling students to state i in year t with high school diploma from fee non-neighbor states (i.e. states that do not share a border with i and that have tuition fees in the winter term of year t) minus enrollments to fee non-neighbor states by students with high school diploma from state i, both divided by all new enrollments in state i.	Federal Statistical Office (2014b): TAB-13
Migration.no.fees	Enrolling students to state i in year t with high school diploma from non-fee states (i.e. states that do not have tuition fees in the winter term of year t) minus enrollments to non-fee states by students with high school diploma from state i, all divided by all new enrollments in state i.	Federal Statistical Office (2014b): TAB-13
Migration.international	Share of new enrollments of international students (i.e. students that did not obtain their high school diploma in Germany) to state i in year t relative to all new enrollments in state i	Federal Statistical Office (2014b): TAB-13

Notes: Data from the Federal Statistical Office are available in the ".xls" format in German in the respective sheet (indicated by TAB or ZUS). They can be found at the bottom of the page `https://www.destatis.de/DE/ZahlenFakten/GesellschaftStaat/BildungForschungKultur/Hochschulen/Hochschulen.html` under "Ausgewählte Publikationen", using the name of the report and the reference. Reports from earlier years are in the report of the respective year. More details on the calculation and choice of spatial variables in text.

year t). This makes it necessary to use publicly available data aggregated on a state-level instead of individual data, since the latter cannot be representative for a state's specific

cohort (i.e. taking less than 20 observations to represent an entire cohort for the majority of tuples). first year students There were 11 state-year combinations with no observations

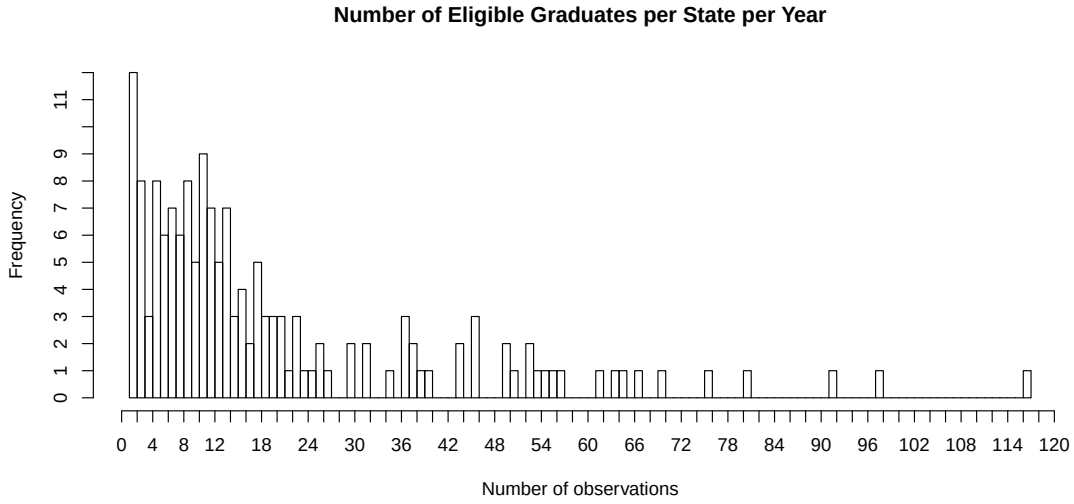


Figure 4.4: Histogram of the Number of Eligible High School Graduates in Each State and Year (i.e. 160 Tuples) in the SOEP Data Set *edubio*
Note: There was one state-year combination with no observations at all.

at all

Information on Control Variables from the SOEP

The control variables can be split up into socio-economic variables (Table 4.4), variables describing university statistics (Table 4.5) and spatial variables (Table 4.6).

Even though the number of households can vary, this cannot bias results since we measure shares of the population. In some cases, there is a substantial amount of missing values (i.e. "Rent", "Income"), which does not pose a large problem as enough data points are still available. These pre-chosen variables are all publicly available and are all potentially correlated with the outcome or the effect.

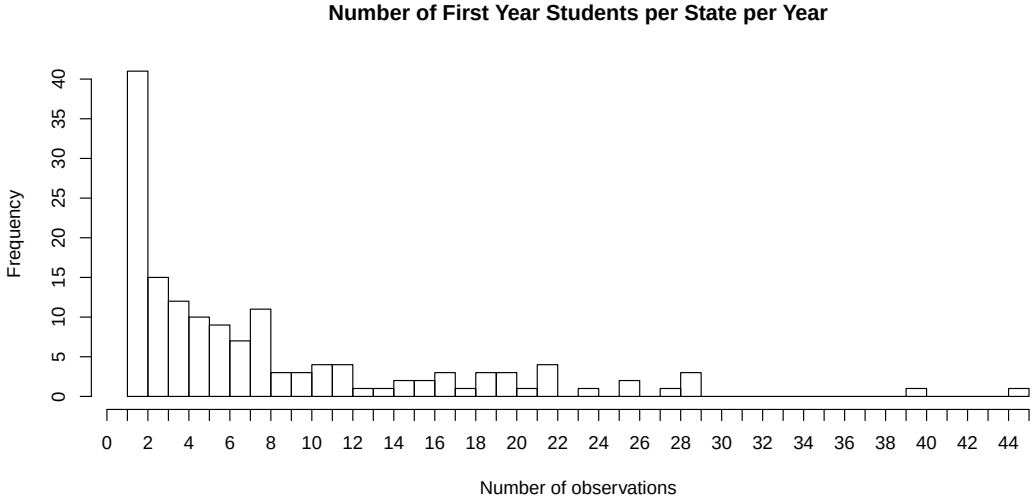


Figure 4.5: Histogram of the Number of First Year Students in Each State and Year (i.e. 160 Tuples) in the SOEP Data Set *edubio*
Note: There were 11 state-year combinations with no observations at all.

For the Destatis-data, the variables are already aggregated on a state level, while for the SOEP-data, the aggregation is done manually using the HID[23], which identifies a household over different subsamples and over time. All SOEP variables are therefore mean values (rent, income, life satisfaction).

Approximation of the Response With Non-public Data

With additional information on the number of new enrollments $NO_{j,t}$ with graduation in state j enrolling anywhere in Germany at t combined with $NE_{i,t}^{(j)}$ and $HG_{j,t}$ we can augment the approximation of $EHG_{i,t}$. Moreover, in order to additionally control for effects from postponers HG_{t-1}, HG_{t-2} in $EHG_{i,t}$, we employ extra non-public information[24] on the number of new enrollments $NE_{\tau,i,t}^{(j)}$ in state i in WT $t/t+1$ with high school diploma obtained in year τ. With this, we can obtain an alternative approximation $AHG_{j,i,t}^{*}$ of

[23]HID stands for Household-ID. For the variable "Life Satisfaction", data was available on a personal level. This does not make a difference since mean values are used.
[24]Provided by the Federal Statistics Office on request for a fee.

the number of high school graduates in j potentially moving to i at t

$$AHG^{*}_{j,i,t} = \max\left\{ \sum_{l=0}^{2} c_{i,j,t,t-l} HG_{j,t-l} \ , \ NE^{(j)}_{i,t} \right\} , \tag{4.12}$$

with share $c_{i,j,t,\tau} = \dfrac{NE^{(j)}_{\tau,i,t}}{NO_{j,t}}$ of enrollments from j to i within the cohort of $t-l$ relative to all enrollments from j in year t, approximating the potentially moving share of the graduates $HG_{j,\tau}$ (See Table 4.7 and Figure 4.6 for a (graphical) overview of involved sets and their role) .[25] We focus on numbers up to a time lag of $l = 2$ in $\tau = t-l$, which cover generally more than 75% of enrollments (on the German level), and use this graduation time specific information also for state i to get a refined approximation of $EHG_{i,t}$ by

$$EHG^{*}_{i,t} = HG_{i,t} + \sum_{l=1}^{2} c_{i,i,t,t-l} HG_{i,t-l} + \sum_{j \neq i} AHG^{*}_{j,i,t} + NE^{(int)}_{i,t} \ . \tag{4.13}$$

Table 4.7: Summary of Defined Quantities for the Response

Enrollments in state i from anywhere:	$NE_{i,t} = \sum_{j=1}^{n} NE^{(j)}_{i,t} + NE^{(int)}_{i,t} = \sum_{j=1}^{n} \Big(\sum_{\tau=t-\psi}^{t} NE^{(j)}_{\tau,i,t} \Big) + NE^{(int)}_{i,t}$
Enrollments from one state j to anywhere in Germany:	$NO_{j,t} = \sum_{\tau=t-\psi}^{t} NO_{\tau,j,t} = \sum_{\tau=t-\psi}^{t} \sum_{i=1}^{n} NE^{(j)}_{\tau,i,t}$
Eligible set of high school graduates in i:	$EHG_{i,t} = NE^{(int)}_{i,t} + HG_{i,t} + \sum_{j \neq i} AHG_{j,i,t}$
International Enrollments to i:	$NE^{(int)}_{i,t}$
High school graduates from i:	$HG_{i,t}$
High school graduates from j affected by enrollments in i:	$AHG_{j,i,t} = \theta NE^{(j)}_{i,t} + (1-\theta) HG_{j,t}$

Notes: ψ stands for the maximum number of years after which a high school graduate enrolled to a German university. Here, the index τ or rather ψ is set so that it includes all students from earlier cohorts.

[25] As it can happen that $NO_{j,t} > HG_{j,t-l},\ l = 0, 1, 2$, we ensure that $AHG^{*}_{j,i,t}$ is at least $NE^{(j)}_{i,t}$.

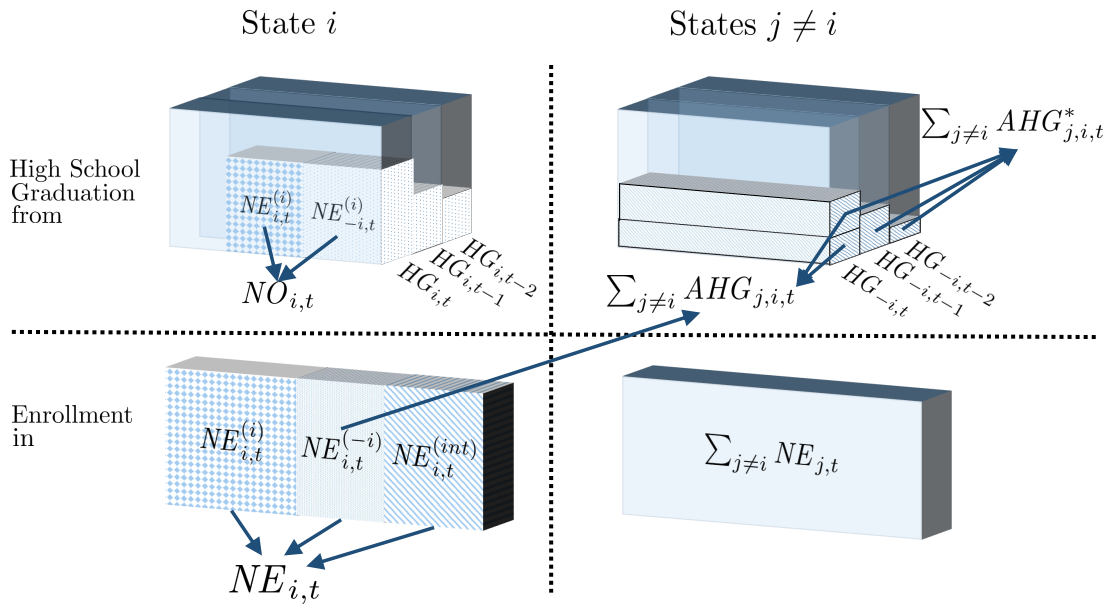


Figure 4.6: Illustration of the Composition of the Response Variable

Notes: High school graduates from different cohorts are depicted in the top boxes, new enrollments in the bottom boxes. The index $-i$ describes all states $j = 1, \ldots, 16$ except the ith state.

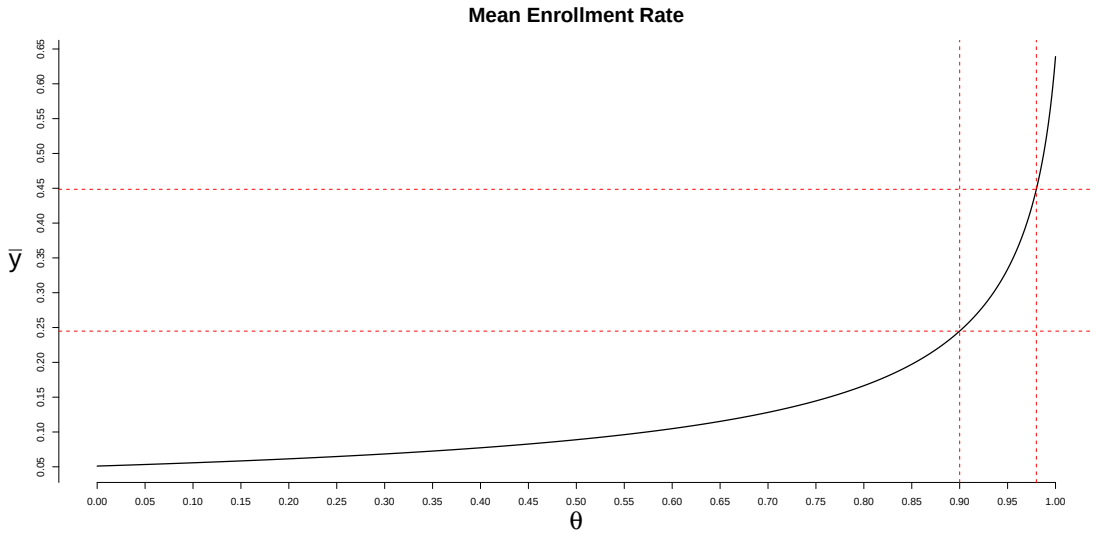


Figure 4.7: Illustration of Mean Values of the Response Variable for a Given Value of θ

Note: The red dotted lines show the lower boundaries $\theta = 0.9$ and $\theta = 0.98$ with their respective mean response value.

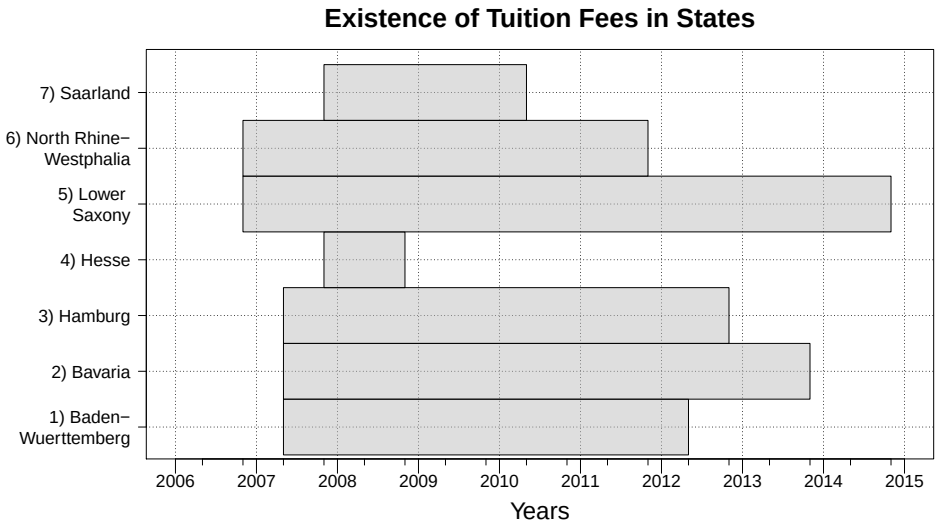


Figure 4.8: Overview of the Timing of Tuition Fees in German States (Presence in Gray)
Notes: The winter term (starting October) and summer term (starting April) are indicated with small ticks. States not listed had no tuition fees at all.

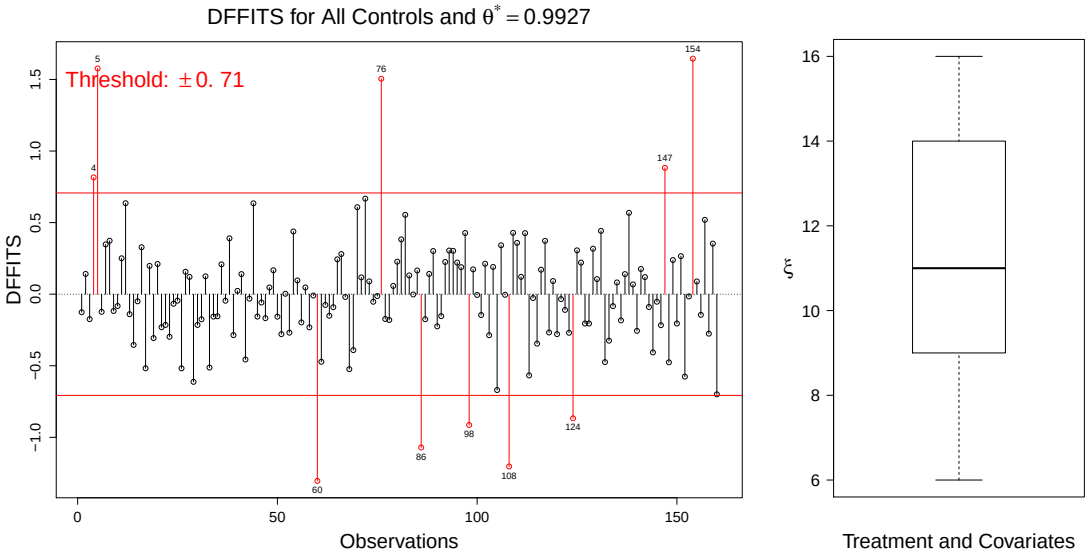


Figure 4.9: DFFITS and Boxplots of DFBETAS for Pure Double Selection

Notes: On the left we report for $\theta^* = 0.9927$ with all controls over all stacked observations k the $DFFITS_k = (\hat{y}_k - \hat{y}_{-k})\big/(s_k h_{kk})$, where $\hat{y}_{-k}$ is the prediction for point k without k being in the regression, s_k is the standard error of the regression without point k and h_{kk} is the leverage of point k. The threshold is $\pi_{DFFITS} = \frac{\sqrt{p}}{nT}$ and influential observations are marked in red. On the right, the boxplots displays $\xi_j = \sum_{k=1}^{N} \mathbb{1}_{\{|DFBETAS_{j,k}| > \pi_{DFBETAS}\}}$ over all components j with $DFBETAS_{j,k} = (b_j - b_{j,-k})\big/\left(\hat{V}(b)_{jj}\right)$, where $\hat{V}(b)_{jj}$ is the estimated variance of the OLS estimate b_j, and $b_{j,-k}$ is the same estimate without point k, and $\pi_{DFBETAS} = \frac{2}{\sqrt{nT}} = \pm 0.16$.

4.7.2 Algorithm for Threshold Choice

To obtain the thresholds for different θ-values in the full data set, we compute the thresholds automatically using the following algorithm:

1. For a given $\tilde{\theta}$, order the inclusion frequencies $\Pi^1_{\tilde{j}} \in \{\Pi^1_j,\ j = 1, \ldots, p\}$, $\tilde{j} = 1, \ldots, p$ and obtain $\Pi^1_{\tilde{1}}, \ldots, \Pi^1_{\tilde{j}}, \ldots, \Pi^1_{\tilde{p}}$, i.e. $\Pi^1_{\tilde{1}} \geq \cdots \geq \Pi^1_{\tilde{j}} \geq \cdots \geq \Pi^1_{\tilde{p}}$.

2. Compute the difference $\Delta_j = \Pi^1_{\tilde{j}} - \Pi^1_{\tilde{j}+1}$ for all $A = \{\tilde{j} : \Pi^1_{\tilde{j}+1} > \pi_{min}\}$, i.e. look at the distance between inclusion frequencies.

3. Choose the cutoff $\pi_{1,\tilde{\theta}}$ to lie at index $\hat{j} = \max\{\underset{\tilde{j} \in A}{argmax} \Delta_{\tilde{j}}\}$, and obtain $\pi_{1,\tilde{\theta}} = \Pi_{\hat{j}+1,1}$.

The algorithm chooses the cutoff at a large difference between ordered inclusion frequencies, i.e. where noise variables are distinguished by true influencing variables. The minimum threshold $\pi^{min}_{1,\theta}$ is set to ensure that the largest difference between inclusion frequencies does not occur between two noise variables (i.e. with very low inclusion frequency). We set this $\pi^{min}_{1,\theta},\ \theta \in [0.98, 1]$ in the following way to make sure noise variables are screened out:

$$\pi^{min}_{1,\theta} = \begin{cases} 0.945 & \textit{All} \text{ Data set and } \theta \in [0.98, 0.992] \\ 0.975 & \textit{All} \text{ Data set and } \theta \in (0.992, 1] \\ 0.98 & \textit{Fees} \text{ Data set and } \theta \in [0.98, 1] \\ 0.9 & \textit{Small} \text{ Data set and } \theta \in [0.98, 1] \end{cases}.$$

For the large data set, the minimum threshold is adapted once as $\bar{y}$-values rise strongly with θ, especially when θ is close to 1. There, all variables have higher inclusion frequencies which makes it necessary to adapt $\pi^{min}_{1,\theta}$. For π_2, we have no variation in θ and compute the thresholds in the same manner as above manually. We obtain

$$\pi_2 = \begin{cases} 0.9 & \textit{All} \text{ Data set} \\ 0.9 & \textit{Fees} \text{ Data set} \\ 0.93 & \textit{Small} \text{ Data set} \end{cases}.$$

4.7.3 Design-Based Standard Errors

In the following, we describe the detailed step-wise version of obtaining the design-based standard errors for the model in equation (4.11) as proposed in Abadie et al. (2020). We calculate such errors SE of the treatment effect $\beta_{(0)}$ in (4.11) as

$$SE(\beta_{(0)}) = \sqrt{(V^\mathsf{T}V)^{-1}G(V^\mathsf{T}V)^{-1}}\,, \tag{4.14}$$

where V is the scalar residual from regressing D jointly on X and Z, and G is the sample version of the variance V_ϵ of ϵ in (4.11). D, X and Z are the stacked vectors of $\ddot{d}_{i,t}$, $\ddot{x}_{i,t}$ and $\ddot{z}_{i,t}$, i.e. $D = (\ddot{d}_{1,1}, \ldots, \ddot{d}_{i,t}, \ldots, \ddot{d}_{n,T})^\mathsf{T}$, $X = (\ddot{x}_{1,1}, \ldots, \ddot{x}_{i,t}, \ldots, \ddot{x}_{n,T})$ and $Z = (\ddot{z}_{1,1}, \ldots, \ddot{z}_{i,t}, \ldots, \ddot{z}_{n,T})$. More specifically, we calculate them as follows:

1. Let $U = (X \quad Z)$ and calculate $V = D - \Lambda U$, where $\Lambda = \left(U^\mathsf{T}U\right)^{-1}U^\mathsf{T}D$ is the least squares estimator of regressing D on U.

2. Calculate the residuals $\hat{\ddot{\epsilon}}^{(1)}_{i,t}$ from the least-squares regression in equation 4.11 and obtain $\hat{V}_{\hat{\ddot{\epsilon}}} = \left(V_{1,1}\hat{\ddot{\epsilon}}^{(1)}_{1,1}, \ldots, V_{i,t}\hat{\ddot{\epsilon}}^{(1)}_{i,t}, \ldots, V_{n,T}\hat{\ddot{\epsilon}}^{(1)}_{n,T}\right)^\mathsf{T}$.

3. Calculate the least-squares estimator $\hat{\beta}_\epsilon$ of a regression of $\hat{V}_{\hat{\ddot{\epsilon}}}$ on U as $\hat{\beta}_\epsilon = \left(U^\mathsf{T}U\right)^{-1}U^\mathsf{T}\hat{V}_{\hat{\ddot{\epsilon}}}$. The fitted values of this regression serve as an estimate of $E[\hat{V}_{\hat{\ddot{\epsilon}}}]$, which is needed to calculate $Var[\hat{V}_{\hat{\ddot{\epsilon}}}]$.

4. Calculate $G = Var[\hat{V}_{\hat{\ddot{\epsilon}}}] = \left(\hat{V}_{\hat{\ddot{\epsilon}}} - U\hat{\beta}_\epsilon\right)^\mathsf{T}\left(\hat{V}_{\hat{\ddot{\epsilon}}} - U\hat{\beta}_\epsilon\right)$

5. Calculate $SE(\beta_{(0)}) = \sqrt{(V^\mathsf{T}V)^{-1}G(V^\mathsf{T}V)^{-1}}$.

4.7.4 Simulation

Using $i = 1, \dots, n$, $t = 1, \dots, T$, and $g = 1, \dots, p$ with $T = 10$, $n = 16$, $N = nT$, and $p = 30$, we simulate the following model

$$\begin{aligned} \tilde{y}_{i,t} &= \eta_0 d_{i,t} + \eta_1 \tilde{x}_{i,t} + \alpha_i + \sigma_1(d_{i,t}, x_{i,t})\epsilon_{i,t}^{(1)} \ , \\ d_{i,t} &= \eta_2 \tilde{x}_{i,t} + \sigma_2(x_{i,t})\epsilon_{i,t}^{(2)} \ , \end{aligned}$$

with coefficients depending on g: $\eta_0 = 0.5$, $\eta_1^{(g)} = \frac{5}{g}\mathbb{1}_{\{g \leq 10\}}$, and $\eta_2^{(g)} = \frac{5}{g-6}\mathbb{1}_{\{7 \leq g \leq 10\}}$ for $g \neq 6$, zero otherwise. The coefficients of covariates are up to 10 times higher than the coefficient of the treatment, since such large differences are also likely to arrive in our empirical application, where the expected treatment effect is relatively small. We generate the fixed effects[26] as $\alpha_i \sim \mathcal{N}(0, \sqrt{\frac{4}{T}})$ and $x_{i,t} \sim \mathcal{N}(0, \Sigma)$[27], with $\Sigma_{v,w} = 0.5^{|w-v|}$, v representing the rows and w the columns of Σ, $v \neq w$. For $v = w = 1, \dots, 10$, $\Sigma_{v,w} = 2$, and for $v = w = 11, \dots, 30$, $\Sigma_{v,w} = 6$. The errors are independently distributed as $\epsilon_{i,t}^{(1)} \sim \mathcal{N}(0,1)$ and $\epsilon_{i,t}^{(2)} \sim \mathcal{N}(0,1)$ with a heteroskedastic structure given by

$$\sigma_1(d_{i,t}, x_{i,t}) = \sqrt{\frac{(1 + \eta_0 d_{i,t} + \eta_1 x_{i,t} + \alpha_i)^2}{\mathbb{E}_N[(1 + \eta_0 d_{i,t} + \eta_1 x_{i,t} + \alpha_i)^2]}}, \ \sigma_2(x_{i,t}) = \sqrt{\frac{(1 + \eta_2 x_{i,t})^2}{\mathbb{E}_N[(1 + \eta_2 x_{i,t})^2]}} \ .$$

For the simulation, we generate each $\gamma_g \sim U[\frac{2}{3}inf, inf]$, where $inf \in \{0, 1, 5\}$ and $g \in \mathcal{D}$ depending on the scenario. In each scenario (i.e. different inf-values), we distort covariates either from the active set ($\mathcal{D} = \{j : \ |\eta_1^{(j)}| + |\eta_2^{(j)}| \neq 0\}$), the inactive set ($\mathcal{D} = \{j : \ |\eta_1^{(j)}| + |\eta_2^{(j)}| = 0\}$) or the response y. For distortion of covariates, we modify them to $\tilde{x}_{i,t} = x_{i,t} + \gamma, \ t = 10$. This means that $\gamma_g = 0$ for either $g > 10$ (inactive set) or $g \leq 10$ (active set). When y is distorted, we have $\tilde{y}_{i,t} = y_{i,t} + \zeta, \ t = 10$ and $\zeta \sim U[-inf, inf]$.

We report mean values over 1000 replications for the absolute bias of estimators $\hat{\eta}_0$ from η_0, the root mean squared error for η_0 with $RMSE_{\eta_0} = \sqrt{Bias^2_{\eta_0, \hat{\eta}_0} + Var_{\hat{\eta}_0}}$, the number of selected covariates, the true positive rate

[26] Note that we omit using a fixed effect in the second equation of this data generating process since using the demeaning framework here, such an effect disappears algebraically.

[27] $x_{i,t} = (x_{i,t}^{(1)}, \dots, x_{i,t}^{(g)}, \dots, x_{i,t}^{(p)})^\intercal$: for $g, k = 1, \dots, p$, $x_{i,t}^{(g)}$ represents a covariate that is standard normal with a correlation of $\rho = 0.5^k$ to $x_{i,t}^{(g+k)}$ and $x_{i,t}^{(g-k)}$, $1 \leq g - k \leq g + k \leq p$.

TPR=$\left(\sum_{g=1}^{p} \mathbb{1}_{\{\eta_1^{(g)} \neq 0\}} \mathbb{1}_{\{\hat{\eta}_1^{(g)} \neq 0\}}\right) \left(\sum_{g=1}^{p} \mathbb{1}_{\{\eta_1^{(g)} \neq 0\}}\right)^{-1}$, and the false positive rate FPR=$\left(\sum_{g=1}^{p} \mathbb{1}_{\{\eta_1^{(g)} = 0\}} \mathbb{1}_{\{\hat{\eta}_1^{(g)} \neq 0\}}\right) \left(\sum_{g=1}^{p} \mathbb{1}_{\{\eta_1^{(g)} = 0\}}\right)^{-1}$.

Additionally to the simulation results in the main paper, the following remarks can be made. When distorting the inactive set, using a higher minimum threshold reduces the FPR even more than in other cases, as the noise variables have more influence. When regarding post-Lasso, however, $\pi_{min} = 0.5$ seems to perform better in general, which can be explained by the post-Lasso not detecting all relevant covariates in the simulated data, where a lower threshold leads to the inclusion of more relevant variables compared to noise variables and improves the method here. For the double selection, only more noise variables are added since all relevant variables are already (almost) always detected. When distorting the response, bias and *RMSE* values go up in general for all procedures, but their relative performance compared to the oracle does not get worse. Comparing stability procedures to their non-stable counterparts, we see that the latter include up to twice as many covariates without much improvement on the TPR, but high increases in the FPR. This confirms the hypothesis that without stability selection, many irrelevant covariates are included in the model, which increases the bias and *RMSE*. The rejection rate is especially high for all post-Lasso procedures, which is not surprising given their high bias and relatively low standard errors that are a result of including fewer variables in the model. Small standard errors also affect the *RMSE* values, and in scenarios with high distortions in the response y, the post-Lasso has a similar *RMSE* compared to its double selection counterpart (regarding the stability procedures).

Taking a closer look at the different forms of distortion, we do not observe much change for high *inf*-values when we distort variables from the inactive set. As expected, when influential observations are only present in the noise variables, they do not affect the selection procedures much. When distorting the active set only, however, procedures with the post-Lasso select fewer (relevant) variables due to the added noise, which leads to a higher bias (for the stability cases), and increases *RMSE* values. The double selection procedures seem to be very robust against such distortions, with all measures remaining relatively unchanged. This is not surprising, since the double selection procedure helps to reduce such a bias by taking the second equation into account. Finally, distorting the

response is interesting, since both relevant and irrelevant covariates are affected at the same time. Even with extremely high distortions, the double selection procedures keep a lower bias compared to the other methods and double selection with stability selection has very low FPRs, while selecting almost all variables from the active set. All in all, the simulation shows that only when we use stability selection, we can select the right variables without including too many noise variables. In our simulated model, where it is hard to distinguish between covariates and the treatment effect is relatively small compared to the effects of other covariates, the non-stable methods perform worse over all distortion scenarios. Results are similar using a lower correlation among covariates. (Available upon request). Furthermore, we see that when some covariates explain the treatment well, but only have a moderate effect on the response (which is the case in the application), double selection outperforms the post-Lasso in terms of bias and rejection rate.

Table 4.8: Simulation Results for Different Forms and Strengths of Influential Observations With Design-based Standard Errors

	Absolute $Bias_{\eta_0}$			$RMSE_{\eta_0}$			# Covariates			TPR			FPR			Rejection Rate		
Size of Distortion:	0	1	5	0	1	5	0	1	5	0	1	5	0	1	5	0	1	5
Distortion in the Active Set																		
PL Stab: 0.5	0.155	0.158	0.171	0.025	0.026	0.030	9.284	9.333	8.131	0.839	0.838	0.784	0.045	0.048	0.015	0.997	0.997	0.994
DB Stab: 0.5	0.087	0.087	0.081	0.020	0.020	0.018	10.680	10.697	10.144	1.000	1.000	1.000	0.034	0.035	0.007	0.121	0.128	0.103
PL Stab: 0.7	0.161	0.163	0.181	0.027	0.028	0.033	8.379	8.341	7.384	0.804	0.801	0.730	0.017	0.017	0.004	1.000	1.000	1.000
DB Stab: 0.7	0.087	0.086	0.081	0.020	0.020	0.018	10.187	10.199	10.027	1.000	1.000	1.000	0.009	0.010	0.001	0.126	0.114	0.106
Post Lasso	0.144	0.143	0.121	0.023	0.023	0.021	19.377	19.193	16.813	0.917	0.918	0.935	0.510	0.501	0.373	0.886	0.866	0.610
Double Selection	0.091	0.091	0.086	0.020	0.020	0.019	21.272	20.694	17.371	1.000	1.000	1.000	0.564	0.535	0.369	0.169	0.167	0.135
Fixed Effects All	0.092	0.094	0.089	0.020	0.020	0.019	30.000	30.000	30.000	1.000	1.000	1.000	1.000	1.000	1.000	0.188	0.197	0.182
Oracle	0.086	0.086	0.081	0.020	0.020	0.018	10.000	10.000	10.000	1.000	1.000	1.000	0.000	0.000	0.000	0.121	0.109	0.105
Distortion in the Inactive Set																		
PL Stab: 0.5	0.155	0.157	0.157	0.025	0.026	0.026	9.284	9.403	9.249	0.839	0.840	0.842	0.045	0.050	0.042	0.997	0.994	0.996
DB Stab: 0.5	0.087	0.087	0.087	0.020	0.020	0.020	10.680	10.774	10.589	1.000	1.000	1.000	0.034	0.039	0.029	0.121	0.124	0.125
PL Stab: 0.7	0.161	0.163	0.162	0.027	0.028	0.027	8.379	8.369	8.300	0.804	0.801	0.802	0.017	0.018	0.014	1.000	1.000	1.000
DB.Stab: 0.7	0.087	0.086	0.086	0.020	0.020	0.020	10.187	10.234	10.168	1.000	1.000	1.000	0.009	0.012	0.008	0.126	0.122	0.119
Post Lasso	0.144	0.143	0.145	0.023	0.023	0.024	19.377	19.236	18.530	0.917	0.917	0.917	0.510	0.503	0.468	0.886	0.868	0.880
Double Selection	0.091	0.092	0.092	0.020	0.020	0.020	21.272	21.061	20.537	1.000	1.000	1.000	0.564	0.553	0.527	0.169	0.171	0.178
Fixed Effects All	0.092	0.094	0.094	0.020	0.021	0.021	30.000	30.000	30.000	1.000	1.000	1.000	1.000	1.000	1.000	0.188	0.195	0.200
Oracle	0.086	0.087	0.087	0.020	0.020	0.020	10.000	10.000	10.000	1.000	1.000	1.000	0.000	0.000	0.000	0.121	0.114	0.114
Distortion in the Response																		
PL Stab: 0.5	0.155	0.158	0.169	0.025	0.026	0.030	9.284	9.311	8.453	0.839	0.835	0.768	0.045	0.048	0.039	0.997	0.999	0.999
DB Stab: 0.5	0.087	0.088	0.108	0.020	0.020	0.031	10.680	10.799	10.786	1.000	1.000	1.000	0.034	0.040	0.039	0.121	0.125	0.111
PL Stab: 0.7	0.161	0.164	0.175	0.027	0.028	0.032	8.379	8.384	7.563	0.804	0.799	0.728	0.017	0.020	0.014	1.000	1.000	1.000
DB.Stab: 0.7	0.087	0.088	0.106	0.020	0.020	0.030	10.187	10.240	10.211	1.000	1.000	1.000	0.009	0.012	0.011	0.126	0.120	0.101
Post Lasso	0.144	0.144	0.149	0.023	0.024	0.027	19.377	19.334	19.020	0.917	0.916	0.908	0.510	0.509	0.497	0.886	0.877	0.842
Double Selection	0.091	0.092	0.111	0.020	0.021	0.031	21.272	21.164	21.497	1.000	1.000	1.000	0.564	0.558	0.575	0.169	0.170	0.143
Fixed Effects All	0.092	0.095	0.114	0.020	0.021	0.031	30.000	30.000	30.000	1.000	1.000	1.000	1.000	1.000	1.000	0.188	0.192	0.156
Oracle	0.086	0.087	0.106	0.020	0.020	0.030	10.000	10.000	10.000	1.000	1.000	1.000	0.000	0.000	0.000	0.121	0.115	0.102

Notes: All values are based on Monte Carlo simulations with 1000 runs and 1000 repeated subsample steps ($C = 1000$). Rejection rates are based on t-tests using a nominal level of 5% with design-based standard errrors (see Abadie et al. (2020)). The remaining measures are means over the 1000 replication runs. PL Stab and DB Stab stand for post-Lasso and double selection with stability selection and the corresponding minum thresholds π_{min}. Oracle is similar to Fixed Effects All but using only true influencing covariates. *inf* indictates the strength of influential observations and is reported for each measure, while the form of influence (active/inactive set and response) is depicted in the rows.

4.7.5 Additional Results

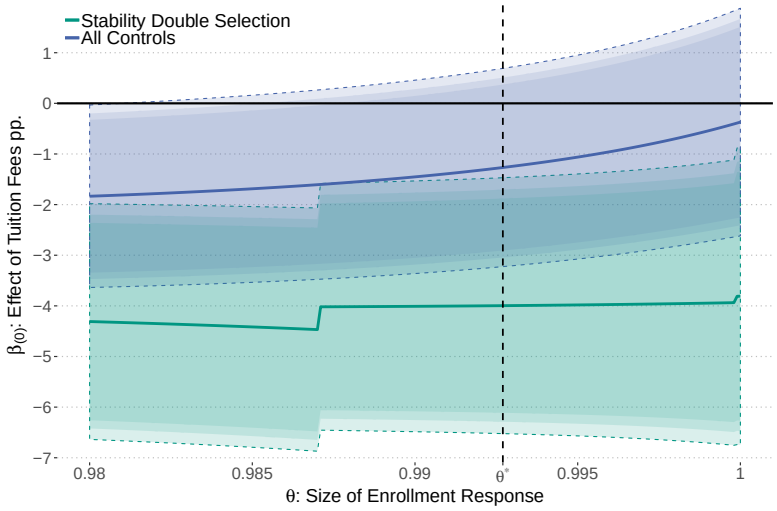


Figure 4.10: Estimates for the Causal Effect β_0 in (4.4) Using Design-Based Errors

Notes: Effects are plotted over the grid of admissible θ for stability double selection and using all controls in a linear fixed effects regression over the grid of admissible θ in $AHG_{j,i,t}$ from (4.3). We depict 95%, 92.5% and 90% CIs in shaded colors, which are calculated on design-based standard errors (Abadie et al. (2020)).

Table 4.9: Estimates of the Causal Effect of Tuition Fees $\beta_{(0)}$ for θ^* in Different Time Frames With Design-Based Standard Errors

	Data sets			No. of Variables
Tuition Fees	All	Fees	Small	All/Fees/Small
min MSD with θ^*:	*0.9927*	*0.9924*	*0.9934*	
All Controls	−1.267 (0.989)	−1.952 (1.167)	-	19/19/-
Post-Lasso Stability	−2.538 (1.303)	−2.599* (1.272)	−6.345** (1.495)	4/4/3
Double Selection Stability	−3.996** (1.278)	−3.180* (1.299)	−16.468*** (2.488)	7/6/7
min MAD with θ^*:	*0.9927*	*0.9926*	*0.9945*	
All Controls	−1.267 (0.989)	−1.941 (1.168)	-	19/19/-
Post-Lasso Stability	−2.538 (1.303)	−2.599* (1.277)	−6.126** (1.538)	4/4/3
Double Selection Stability	−3.996** (1.278)	−3.185* (1.302)	−17.133*** (2.549)	7/6/7
$y_{i,t}^{extra}$ with π_1/π_2:	*0.999/0.9*	*0.9/0.9*	*0.85/0.91*	
All Controls	−1.722* (0.770)	−2.213* (0.881)	-	19/19/-
Post-Lasso Stability	−3.349* (1.311)	−2.234** (0.823)	−11.570*** (1.317)	3/9/2
Double Selection Stability	−3.920*** (1.087)	−2.198* (0.877)	−15.021** (3.688)	6/10/6

Notes: Response values are scaled to a percentage level. Standard errors in parentheses are calculated based on treatment design and finite populations (Abadie et al. (2020)). *p<0.05; **p<0.01; ***p<0.001 indicate p-values from a t-test on significance from zero. θ^* is chosen according to minimum mean squared deviation (MSD) and minimum mean absolute deviation (MAD).

5 Predicting Property Prices Using Augmented Crime Data: Extracting Information From Satellite Images with Convolutional Neural Networks

5.1 Introduction

Cities around the world are growing as more people move to economically strong regions (Buhaug and Urdal, 2013). This trend may be best observed in prices of properties in big urban centers and it is thus an important task to be able to predict and to determine where such processes are happening. Given socio-economic evidence that crime plays a significant role in the determination of property values (Ihlanfeldt and Mayock, 2010; Gibbons, 2004), we use crime statistics combined with additional information from satellite images to predict house prices more accurately. Instead of taking the plain number of crimes directly, we construct crime features that contain the additional information from satellite images and use those (among other influencing factors) to predict property values in a linear model. With this, we combine economic intuition about crime with machine learning (ML) methods that are able to extract the additional information from satellite images. This approach outperforms both purely ML-driven models as well as purely linear models taking crime directly into account. Since these features are constructed using convolutional neural networks (CNNs), we quantify which part of input images contain additional information and what influence these parts have on the final property price prediction. We further investigate specifically what the

features, which are extracted from the the CNN constructions, represent and how they contribute to crime.

We show on the example of New York City (NYC) how such a model can be superior to classic approaches and furthermore generalize this model to Philadelphia. We do not re-train the CNN-model for this step and thus employ the same features as for NYC. This retains over 50% of the prediction power compared to NYC, which indicates that CNN-model acts as more general feature extractor for the task at hand. We choose NYC since it is unarguably one of the most important economic centers worldwide and has therefore seen a rise in housing prices over time (Sieg and Yoon, 2020; Haughwout et al., 2008). It is thus crucial to have measures that predict property prices and thus property values. ML-methods provide an often superior prediction performance for such tasks compared to classic linear models, since they can model more complex, non-linear relationships (see e.g. Mullainathan and Spiess (2017) for recent applications). While satellite images and crime statistics are readily available, house price data is too scarce to train a CNN directly, which we show explicitly in the out-of-sample prediction results. We therefore employ the following hybrid approach.

We combine the crime data with satellite images to extract proxy crime-features from a trained CNN, which are then used to predict property sales in a linear model. In NYC, crime statistics, are tracked by the NYC police with their exact GPS-coordinates. We therefore exploit this situation and map each crime to an image to construct our features. In this way, we are able connect the information from satellite images to crimes through an ML-model without losing the interpretability in the final prediction. To highlight how crimes and property sales are influenced by satellite images, we use the SHAP framework from Lundberg and Lee (2017), which helps us to efficiently assess changes in the response when input satellite images are varied both on the crime-level (i.e. on the model trained to predict crimes in NYC from satellite images) as well as on the property price-level (i.e. the full model predicting property prices).

With this transfer learning approach that yields the crime features, we built on work from Jean et al. (2016) that use similar ideas to predict poverty in Africa. In our case, crime data is available with the exact location, while we only have sufficiently accurate data on property sales on a zip-code level. Additionally, we also obtain the sales data

on finer address level and match them to images, leading to qualitatively similar results. Instead of using the (aggregated) crime numbers directly or taking the predicted value from the ML-model given the corresponding satellite image, we extract the information from within the model directly using principal components of neuron outputs of our trained CNN (similar to Jean et al., 2016). With this approach, we preserve more information than using the crime numbers directly for forecasting house prices. This extra information stems from the satellite images and is extracted non-linearly through the features, which makes the model generalizable and reduces overfitting, thus increasing predictive power. Our approach depends crucially on the structure of crime. Firstly, we crime needs to be influenced by the layout and spatial factors in different neighborhoods, implying predictability of crime by satellite images, which is clearly the case (cp. Najjar et al., 2017). Secondly, crime needs to be relevant for house prices, which has also been established in the literature on housing prices that identified crime as one major influence (Ihlanfeldt and Mayock, 2010; Gibbons, 2004).

To obtain these crime features, we use the VGG-16 architecture (Simonyan and Zisserman, 2015) pre-trained on ImageNet [1] as a baseline and fine-tune the network with subsequent pooling and dropout layers to adapt the CNN to satellite images and to find important latent features. The last fully-connected layer before the crime prediction consists of 750 neurons, corresponding to 750 raw features. These are reduced to 100 features using principal component analysis (PCA), and finally taken to predict housing prices in a ridge regression. The classic approach for such a problem would be to employ hedonic regression models (see e.g. Schwartz et al., 2014; Brunauer et al., 2010), usually assuming a linear relationship between factors and prices, which we add as a baseline model in our analysis. Furthermore, we additionally include controlling factors to both the final linear step in the feature learning model and the baseline model. While the baseline model is heavily reliant on those additional factors, the additional controlling factors are less important when the features learned in the ML-model are used. Using the SHAP framework of Lundberg and Lee (2017), we can explain predictions of the CNN for both the crime prediction task as well as for the full model predicting house prices. This approach leverages Shapley values (Shapley, 1953), initially introduced in

[1] First introduced by Jia Deng et al. (2009), see `http://image-net.org/`

game theory, which are adapted to the ML-setting. In this way, we can see the effect of certain parts of each input image on the output (i.e. crime numbers or house price) and find for example that the shadow of high-rise and park spaces have a substantial effect on both crime and prices.

The rest of the paper is structured as follows. In Section 5.2, we present the data used for the empirical results, while Section 5.3 introduces the methodology. The empirical results are shown in Section 5.4 before we conclude in Section 5.5.

5.2 Data

We collect the data for our analysis from four main sources. The satellite data was obtained from the *Google Maps Static API*[2] by placing a rectangular grid over New York City. All images are matched with their respective address (and zip-code) from the center of the image, using the longitude and latitude with the *Google Geocode API*[3]. We pre-process the images by removing the google watermark, as this logo might influence the learning of the CNN, by rescaling the image, and by augmenting the image space adding flipped versions of the image (see Figure 5.1 and Perez and Wang (2017)). The latter is used to counteract overfitting, since the position of a learned feature in the image should not matter and we therefore want to avoid a situation where the trained features are highly dependent on the chosen grid in the image. In total, this grid of images amounts to 6634 and thus to 26536 augmented images.

The NYPD Complaint Data[4] includes all valid felony, misdemeanor, and violation crimes reported to the New York City Police Department (NYPD), including exact location (longitude, latitude) and, amongst others, an NYPD three-digit internal classification code. We use the same crime classification as Vomfell et al. (2018) only taking into account crimes that pose serious threats to public safety. These crimes were violent crimes (murder,non-negligent manslaughter, robbery and aggravated assault) and property crimes (burglary, larceny-theft, motor vehicle theft and arson). They are classified in

[2] `https://developers.google.com/maps/documentation/maps-static/intro`, scale 1, zoom 17. This corresponds to a 400x400 pixel image, with size approximately 350x350 meters.

[3] `https://developers.google.com/maps/documentation/geocoding/intro`

[4] `https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i`

(a) Original image **(b)** Vertical Flip **(c)** Horizontal Flip **(d)** Flipped twice

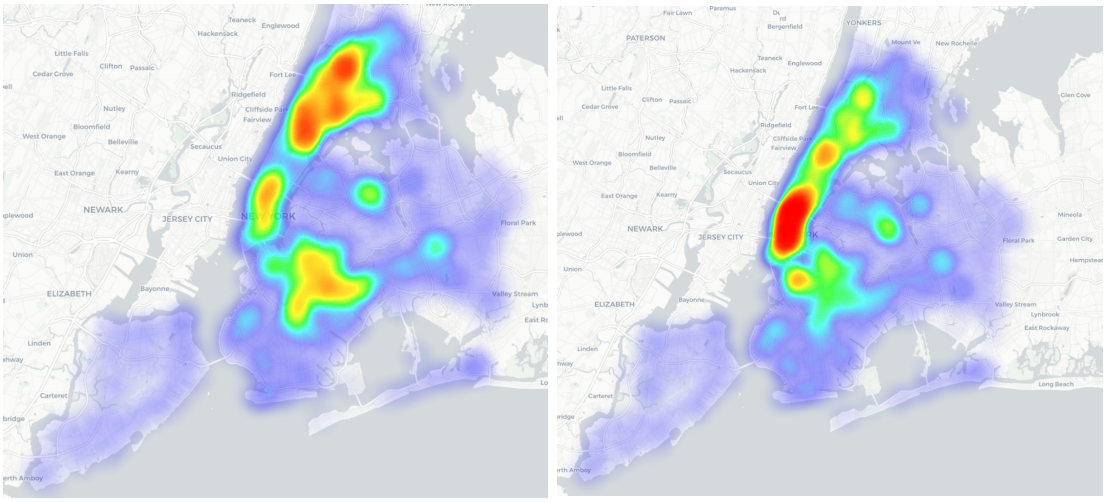
Figure 5.1: Example of Image Augmentation for One Image of the Data Set

Table 5.4 in Appendix 5.6.1. We use data from 2008-2017 and aggregate crime numbers to obtain the outputs for the training of the CNN, which gives us a total of around 5 million crime observations. Figure 5.2 shows that violent crime clusters (left) are most common in the Bronx, Harlem, the southern part of Manhattan and parts of Queens, while property crimes (right) are most common in the southern parts of Manhattan. Overall the distribution of both crime categories is quite similar with significant variation over all of New York City. Each crime is mapped to its closest images out of all images, minimizing the euclidean distance between the coordinates of the crime and the center of the images. This allows precise matching of every crime to each image and to a zip-code. In the results, we distinguish between two different measures of crime per image/zip-code. Firstly, we regard the number of crimes in each of the nine categories from Table 5.4 simultaneously ($9REG$). Secondly, as a baseline, we take the absolute number of crimes aggregated over all categories (REG) instead.

We obtain data on property prices for the year 2018 using the *New York City Sales Data* [5], which contains yearly sales data for properties in New York City, including additional attributes such as square footage and neighborhood. A first visualization of the data can be seen in Figure 5.3, where large heterogeneities are visualized. We compute the price per gross square foot p_{j_i} for each sold property j_i in zip-code or image i[6] and aggregate

[5] `https://www1.nyc.gov/site/finance/taxes/property-annualized-sales-update.page`

[6] We use the median of all prices on image i and the parts of the 8 images directly next to i. For this, we consider all images that lie within a Manhattan distance (i.e. sum-norm) of 1.5 times the length of each image.



(a) Violent Crimes **(b)** Property Crimes

Figure 5.2: Heatmaps of Violent and Property Crimes of a Subsample of 20,000 Observations

Note: Darker values of red indicate higher crime rates.

the price information on a zip-code or image level using the median price[7] of all sales, i.e. we obtain price $P_i = median\{p_{j_z} : z = i\}$ and use this price as our outcome variable Y_i. The respective outliers can be seen in Table 5.3 and in the histograms in Figure 5.14 in Appendix 5.6.1. We also include further potentially controlling factors that could influence house prices and are used in the literature (Mullainathan and Spiess, 2017; Dubin, 1998; Schwartz et al., 2014; Haughwout et al., 2008; Jim and Chen, 2009). These are mean values of the age of the property, the distance to the Empire State Building, the distance to (or number of in case of zip-codes) fire stations and subway stations, and the distance to John F. Kennedy International Airport and LaGuardia Airport, which is more commonly used for national flights. All distances are calculated based on data from *NYC Open Data*[8]. Furthermore, we include census data from the American Community

[7] We also use mean prices with observations larger than three standard deviations away removed from the sample. Results are similar and available upon request from the authors.

[8] `https://opendata.cityofnewyork.us/`

Survey 2015[9] (5-year estimates) for the zip-code level consisting of median household income and unemployment rate.

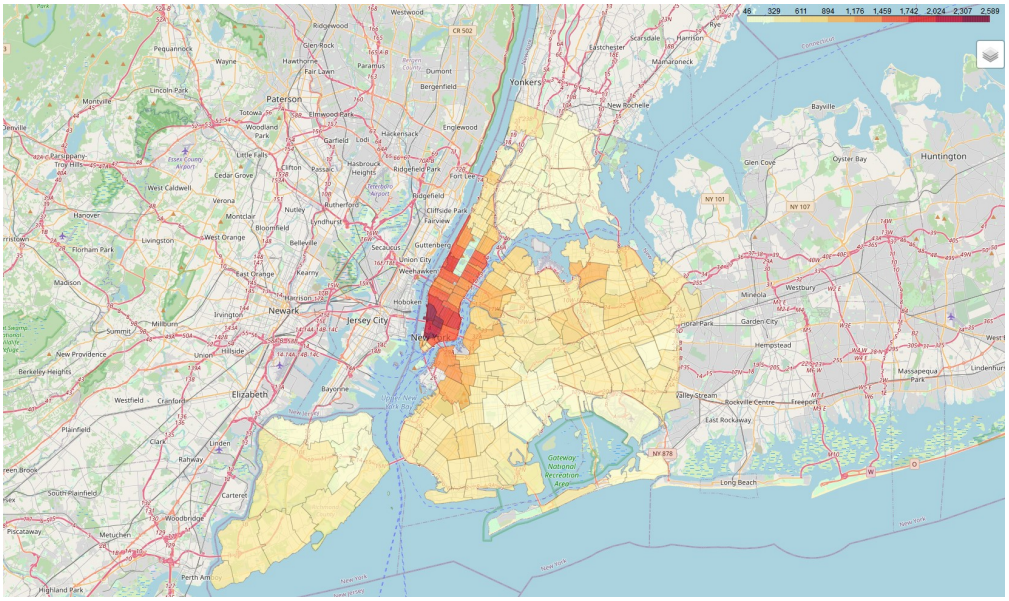


Figure 5.3: New York City Median Property Sales Prices per Square Foot by Zip-Code in 2018

5.3 Model and Methodology

5.3.1 Model

In our empirical analysis, we employ three different models and compare their predictive performance. The first model is the main model of interest, where we obtain non-linear crime features using satellite images and subsequently use these features in a linear setup. The second and third serve as benchmark models and either assume a linear relationship and use crime directly instead of the features, or are purely non-linear, where satellite images are used directly to predict property prices through a CNN. The main model is

[9] `https://data.census.gov`

given by

$$Y_i = \beta^\mathsf{T} X_i + \eta^\mathsf{T} \bar{F}_i^{FE} + \epsilon_i \tag{5.1}$$

with $Y_i \in \mathbb{R}$ as the median house prices in area i, where i is either the ith of the $n = 179$ zip-code areas of New York city or one of the RGB-400×400 satellite images ($n = 19092 = 4 \times 4773$)[10]. $\bar{F}_i^{FE}$ are the crime features, $X_i \in \mathbb{R}^d$ represents controlling factors such as the distance to the empire state building, the building year of the property, the distance to/number of subway stations or fire stations, the median household income, and unemployment rate (on zip-code level). $\beta \in \mathbb{R}^d$ and $\eta \in \mathbb{R}^{100}$ are coefficients of the model. We obtain $\bar{F}_i^{FE}$ by first predicting crime rates C_i with satellite images

$$C_i = g_{CNN}(S_p) + \epsilon_i^{(1)} \ , \tag{5.2}$$

where S_p stands for the satellite data [11] $S_p \in \mathbb{R}^{400 \times 400 \times 3}$, where in the case of using prices on an image level, $i = p$, and the number of crimes $C_i \in \mathbb{R}^c$, with c being the different categories of crimes. $g_{CNN}(S_p)$ represents the predictions of a trained CNN on image S_p, and when on zip-code level, we map each image to a zip-code. In a second step, we use the last (fully-connected) layer of the CNN before the crime prediction as features, for which we reduce the dimension from 750 to 100 using Principal Component Analysis, obtaining $\bar{F}_i^{FE}$. In the case where i equals the zip-codes, the raw outputs for each image are mean-aggregated on a ZIP-code level to obtain $\bar{F}_i^{FE}$. The PCA step is necessary to reduce the number of features relative to the number of observations (179 ZIP Codes) in the final linear model. Additionally, we use ridge regression within the predictive model for Y and the remaining 100 features to reduce overfitting. We also compare this approach to standard ordinary least squares (OLS). Alternatively, we evaluate a scenario where $C_i^{(abs)} \in \mathbb{R}$ is the absolute number of crimes in i[12]. In the two benchmark models,

[10] The increased sample size is due to image augmentation, which is explained in more detail in Section 5.2. The image size is smaller than for training of the CNN, since there are differences between the availability of property data and crime data.

[11] The value of each pixel w_{jkl} can vary between 0 and 255 and is rescaled to 0-1 using min-max scaling, i.e. $\tilde{w}_{jkl}^{(i)} = \dfrac{w_{jkl}^{(i)} - \min_{j,k=1,\dots,400} w_{jkl}^{(i)}}{\max_{j,k=1,\dots,400} w_{jkl}^{(i)} - \min_{j,k=1,\dots,400} w_{jkl}^{(i)}}$ to mimic the input data of the pre-trained VGG-16.

[12] We also analyze another scenario where we replace C_i by two dummies representing high, medium and low-crime areas (i.e. zip-codes). Since results do not differ from absolute results, we do not present

we first employ $Y_i = \beta^\mathsf{T} X_i + \gamma^\mathsf{T} C_i + \epsilon_i^{(2)}$ as a linear baseline model, i.e. using crimes and covariates directly instead of using the features. Secondly, we take $Y_i = f_{CNN}(S_p) + \epsilon_i^{(3)}$ as a purely nonparametric model that uses a CNN directly to predict Y only using the satellite data S_p.

5.3.2 CNN Architecture and Transfer Learning

In our transfer learning model, we employ a VGG-16 (Simonyan and Zisserman, 2015) architecture. We choose this CNN over others such as AlexNet (Krizhevsky et al., 2012) or ResNet (He et al., 2016), since VGG-16 offers a good combination of prediction accuracy and training time in challenges such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). As a comparison, we also employ the faster ResNet-18 in the direct CNN-model.

We take the 13 pre-trained convolutional layers and five max pooling layers of the VGG-16 and combine them with two fully-connected layers (each with 750 neurons) and 3 dropout layers for training to counteract overfitting, which is visualized in Figure 5.4. In the beginning, only the two fully-connected layers are trained, unfreezing the two last convolutional layers only in a second training step in which the learning rate was reduced to allow for a finer adaption of the already pre-trained weights of the convolutional layers. We assess the performance by randomly splitting the sample in training and validation sets for the CNN, and a test set for the final models using the out-of-sample-R^2 ($OOS - R^2$) on the test set. The loss curve for the training and validation data set can be found in Figure 5.16 in Appendix 5.6.1. For the transfer learning approach, we proceed as follows:

1. Obtain $\hat{C}_i$ from (5.2) and obtain $\hat{F}_p^{FE}$ as the last layer of the CNN for each satellite image S_p.

2. Apply PCA on $\hat{F}_p^{FE} \in \mathbb{R}^{750}$, which reduces the dimensionality from 750 to 100, and obtain $\tilde{F}_p^{FE} \in \mathbb{R}^{100}$.

these results here. Formally, we replace C_i by $\left(C_i^{(high)}, C_i^{(medium)}\right) \in \{0,1\}^2$, and low crime areas serve as the base. We define low, medium, and high-crime areas as absolute crime numbers below their 33%-quantile,between the 33%- and 66%-quantile, and above the 66%-quantile.

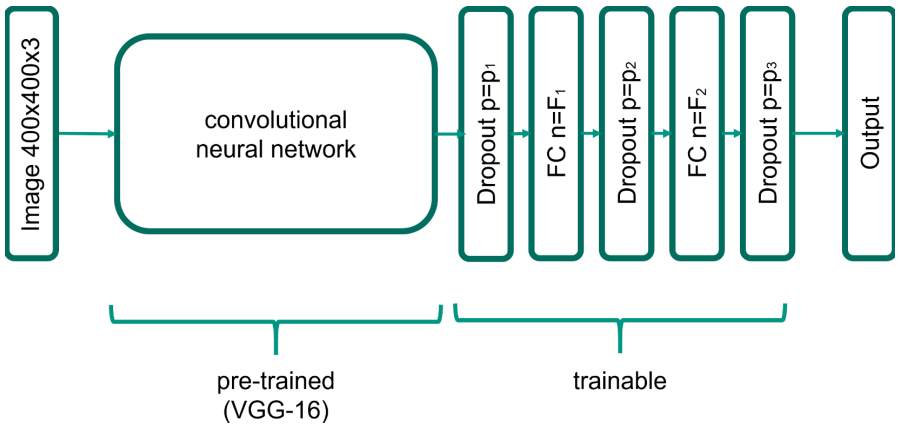


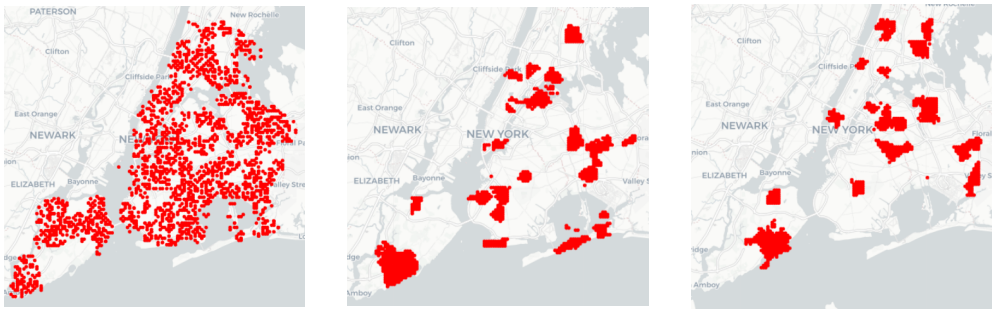
Figure 5.4: Architecture of the Feature Extraction Models

3. For the case of image price data where $i = p$, set $\bar{F}_i^{FE} = \tilde{F}_p^{FE}$. Else, compute the mean of the transformed features $\tilde{F}_p^{FE}$ for each i (important for the zip-code level, where $n = 179$), i.e for $\mathcal{I}_i = \{p : S_p \text{ lies in area } i\}$, calculate $\bar{F}_i^{FE} = \frac{1}{|\mathcal{I}_i|} \sum_{p \in \mathcal{I}_i} \tilde{F}_p^{FE}$.

4. Use this feature representation in a final ridge/OLS regression on the model $Y_i = \beta^\mathsf{T} X_i + \eta^\mathsf{T} \bar{F}_i^{FE} + \epsilon_i^{(2)}$. For the ridge regression models, we use 10-fold cross-validation for hyperparameter tuning.

Furthermore, we split our data on crimes and the satellite images into random subsets. We first split off 15% of the data for testing the predictive power of our models. The remaining 85% are used for training of both the CNN and the subsequent ridge regression models. Due to the high computational requirements of training the CNN, we refrain from employing a cross-validation scheme here and validate the training taking 15 percentage points (pp.) from the 85% of the remaining data. In particular, this validation data set is then used for the evaluation of the loss-functions in the CNN. The rest of the 70% of the full data is our training data set for the CNN. Since for the ridge regression model, we employ 10-fold cross-validation, we take both the training and the validation data for estimating the parameters (i.e. using 85% of the data). Figure 5.5 plots the area of satellite images remaining in each randomly chosen subset.

To understand how the input images drive both the crime rates and the house prices, we employ the SHAP framework of Lundberg and Lee (2017). For the crime rate effects,



(a) Train Data (70%) **(b)** Validation Data (15%) **(c)** Test Data (15%)

Figure 5.5: Graphical Representation of the Three Subsets

we use DeepSHAP, which builds on the DeepLift approach of Shrikumar et al. (2017) and is specifically designed to handle neural networks. To explain effects on house prices (i.e. the full model), we select the model agnostic version Kernel-SHAP that is based on linear LIME (Ribeiro et al., 2016).

This model agnostic Kernel-SHAP linearizes the model and takes an approximated weighted least squares approach to quantify the effect of changes in the outcome compared to a baseline of images. The Kernel-SHAP procedures therefore chooses weights so that the measured effects represent Shapley values. Since this is still computationally too expensive, the contribution of each covariate is simplified by assuming the independence of other covariates. Intuitively, if there is only a small uncertainty over which change caused an effect (e.g. a small number of pixels is changed from the baseline image), the weights are high and vice versa. To reduce complexity of the model (recall that the input size is 400x400x3), we build clusters using an adapted version of the SLIC algorithm (Achanta et al., 2012) to identify 30 clusters on each image that can be compared against a baseline.

To highlight the performance of the CNN, we also compute Shapley values that show the contribution of images to crime directly. For this approach, we can rely on the faster, model-specific DeepSHAP, which is based on DeepLift. DeepLift linearizes the predictions locally around each neuron/layer and then backpropagates these changes through all layers. DeepSHAP changes DeepLift so that the weights from the linearization

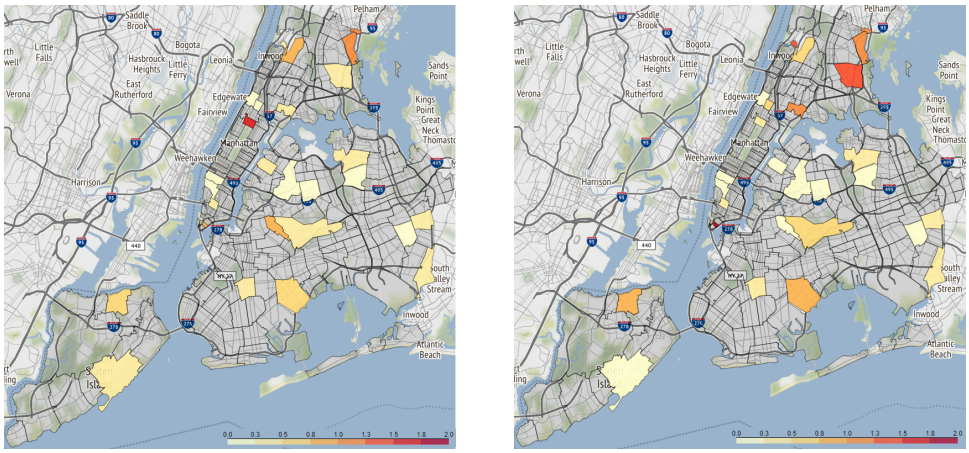
are obtained in a way that mimics Shapley values. For a detailed overview, we refer to Lundberg and Lee (2017).

5.4 Empirical Study

Our main analysis in Section 5.4.1 is conducted on New York City. Leveraging satellite images in our crime feature model, we predict house prices on a zip-code and image level. In Section 5.4.2, we show that crime can be explained well by the features based on the CNN in Section 5.3, and visualize which parts of images are important in predicting crime and prices. Finally, we demonstrate how the trained crime-feature model can be easily generalized to Philadelphia in Section 5.4.3.

5.4.1 Predictive Model Results for New York City

We show the superiority of our approach from Section 5.3 evaluating the out-of-sample R^2 in a variety of settings. We distinguish between different measures of crime (*9REG*, *REG*), using additional controlling factors X_i, and predicting prices on a zip-code level or image level. As described above, the baseline model uses linear regression with crime numbers C_i directly. The results are summarized in Table 5.1.



(a) Hybrid model (9REG): $\frac{\hat{\epsilon}_i^{9REG}}{Y_i}$ **(b)** Baseline model: $\frac{\hat{\epsilon}_i^{Base}}{Y_i}$

Figure 5.6: Relative Absolute Error on Zip-Codes in Test Data Set

Table 5.1: Predictive Power Comparison of Different Models on Zip-Code and Image Level

	With X_i		Without X_i	
Out-of-Sample-R^2	Zip code level	Image level	Zip code level	Image level
CNN (VGG/ResNet)	0.557/0.289	-/0.340	0.557/0.289	-/0.340
9REG:				
Hybrid Model	**0.765**	**0.566**	0.642	**0.365**
Baseline	0.716	0.546	0.570	0.336
REG:				
Hybrid Model	0.748	0.528	**0.765**	0.312
Baseline	0.701	0.447	0.179	0.106

Notes: Out-of-sample-R^2 for models with and without additional covariates X_i. CNNs are only trained on images. This table shows results using OLS on image level data and ridge regression for zip-code level data. The best model in each scenario is marked in bold.

We find that all feature extraction models outperform both the linear models as well as the CNN directly trained on predicting prices using the out-of-sample R^2. Furthermore, the linear model only performs similarly when paired with additional controlling factors. Those covariates might be hard to obtain in general and might not be available on the preferred level. However, the prediction performance employing the extracted features is less dependent on additional external covariates, which suggests that these features contain additional information in comparison to crime. Additionally, we obtain a rough proxy for crime that generalizes well in cities with similar crime structure (see Section 5.4.2).

Figure 5.6 highlights the spatial performance of the different methods. It shows absolute errors[13] $\hat{\epsilon}_i^{base}$ of the baseline and $\hat{\epsilon}_i^{9REG}$ of the hybrid model relative to the true median house price Y_i colored from white (low) to red (high). As expected, the hybrid model performs significantly better in most regions. Both models have their largest errors in

[13]We show results for the zip-code level model for reasons of clarity, without covariates X_i to compare the features against crime, and using 9REG, where the baseline model does not fully break down to provide a fair comparison between the two.

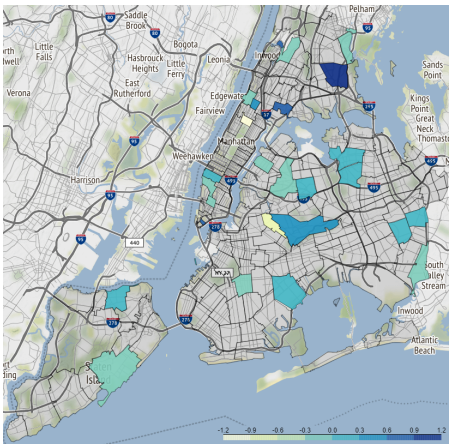


Figure 5.7: Difference Between the Absolute Error of the Baseline Model ($\hat{\epsilon}_i^{base}$) and the Hybrid Model ($\hat{\epsilon}_i^{9REG}$) Relative to the Real Median House Price

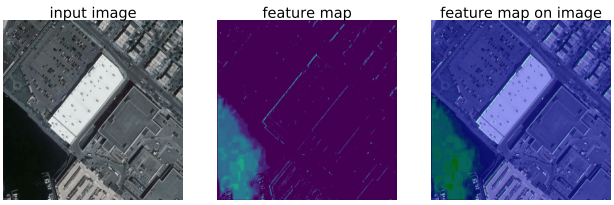
Notes: In more detail, the formula is as follows: $\epsilon_i^{diff} = \frac{\hat{\epsilon}_i^{base} - \hat{\epsilon}_i^{9REG}}{Y_i}$. Darker colors thus represent better performance of the hybrid model.

Manhattan, where house prices are generally much higher. Still, the hybrid model is able to adapt better to this while maintaining a lower general level of error throughout the other zip-codes.

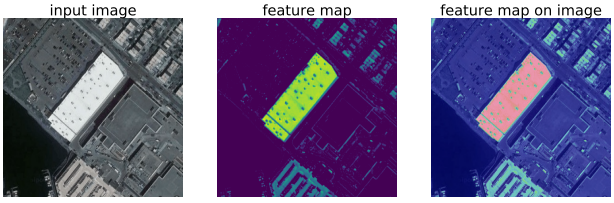
Figure 5.7 visualizes the relative difference in the absolute error of the baseline and hybrid model for each zip-code area. Clearly, the hybrid model outperforms thee baseline for the majority of zip-code. Only for very few areas, the baseline model performs similarly or relatively better, which is mostly caused by areas that are very hard to predict. A direct, per borough comparison of absolute zip-code errors can be found in Figure 5.17 in Appendix 5.6.1. We furthermore run additional out-of-sample methods such as random forests and boosting procedures that show improvement especially when paired with additional controlling factors, see Table 5.6 and 5.7 in the appendix for image- and zip-level, respectively. In the next section, we explore the features and predictability through images more using the model agnostic and model specific methods from Lundberg and Lee (2017).

5.4.2 Explaining Image Features Using SHAP

In this section, we highlight the performance of the hybrid model by shedding light on the information on house prices as well as of crime activity that is extracted by features. Using Shapley values, we can visualize the combined effect of changes in image input (to an average of baseline images) on house prices and crimes.



(a) Filter That Detects Water



(b) Filter That Detects Buildings

Figure 5.8: Selected Low-Level Filters of the CNN Applied on the Same Image

First, we highlight how the CNN extracts information on a low level. Figure 5.8 visualizes how selected learned filters of the CNN are able to distinguish different structures such as buildings or water on the same image. This feature of CNNs can be efficiently achieved by adapting a pre-trained CNN architecture and fine-tuning it on the problem at hand, i.e. satellite images. Going further, the trained CNN extracts the same information on completely different images, which is highlighted in Figure 5.9. There, a specific filter detects large building structures by their shadow, which generalizes well in different environments. For example, the leftmost image of Figure 5.9 has different light conditions compared to the other two images, and is also from another borough (The Bronx vs Manhattan on the other two images).

To quantify the impact of input pixels towards the number of crimes in each of the 9 crime classes specified in Table 5.4, we use DeepSHAP. As an example, Figure 5.10

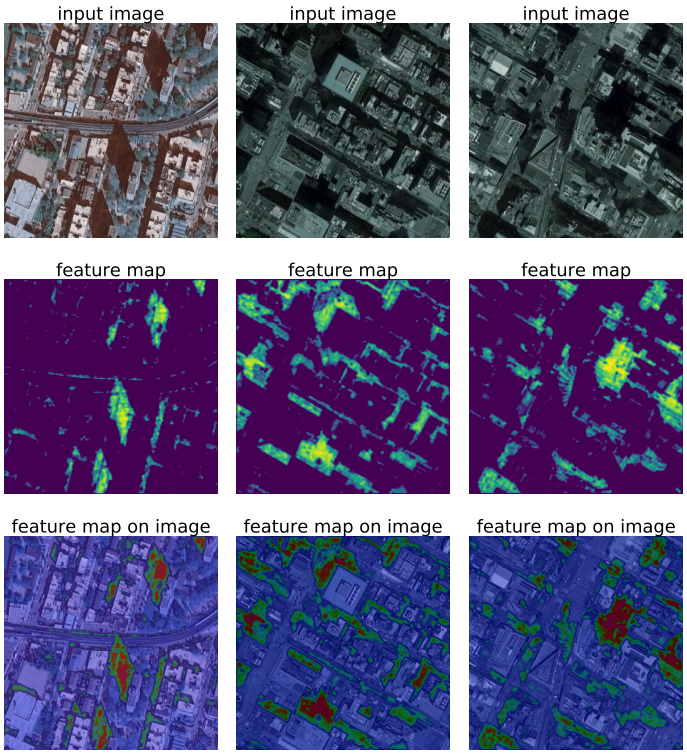


Figure 5.9: Results of Filter 1 in Layer 4 (Middle and Bottom) of the Trained CNN Detecting Shadows of Buildings for Three Different Images (Top)

shows the effect of each pixel on the nine crime classes on an image around Time Square in Manhattan. A red color indicates that having this pixel in the image increases crime activity compared to a baseline image composed of 100 randomly selected images. We can see that the area around Time Square seems to contribute to higher crime in Classes 3 and 4, which represent crimes connected to theft and less strong in Class 6, which involves offenses against public order. This seems reasonable when looking at Time Square as a public place attracting many tourists. Another example where both directions and other crime classes are highlighted can be found in Figure 5.15 in Appendix 5.6.1, which shows a more residential neighborhood.

In a similar fashion, we also analyze the effect of changes in input images on the house prices, i.e. including the CNN, the dimensionality reduction using PCA and the final linear/ridge regression. Since this encompasses more complexity and is computation-

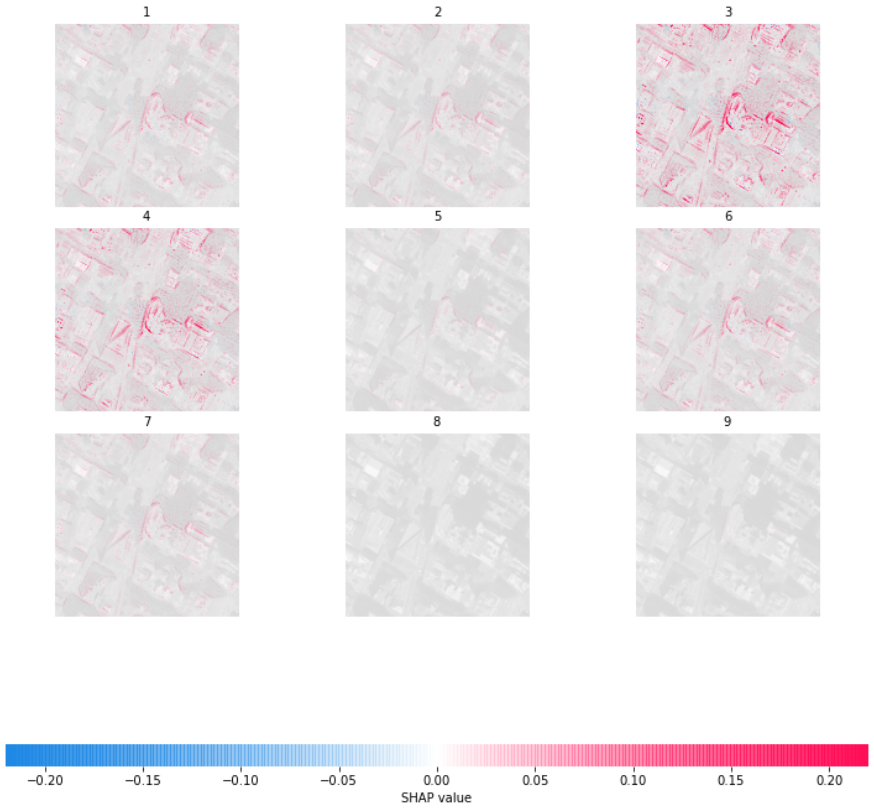


Figure 5.10: Contribution of Input Pixels To the Number of Crimes in the 9 Output Crime Classes Using DeepSHAP at Time Square

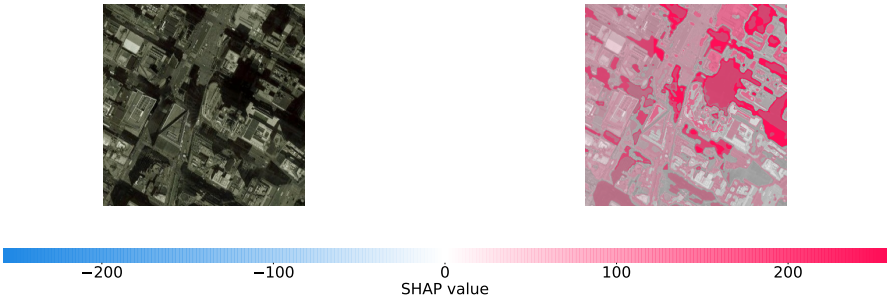


Figure 5.11: SHAP Values Highlighting the Contribution of Building Shadow To Price (Right) at Time Square (Left)

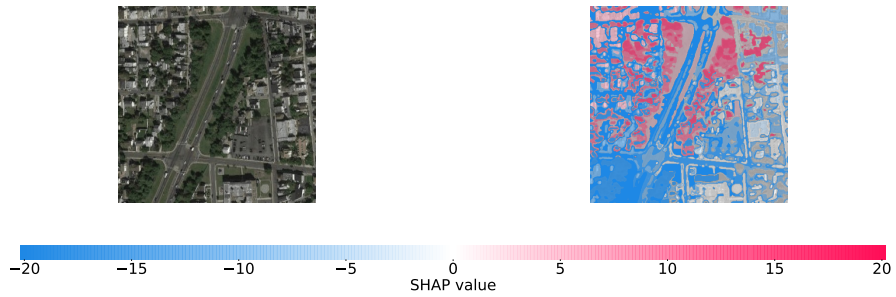


Figure 5.12: SHAP Values Highlighting the Contribution of Green Areas To Price (Right) in a Residential Neighborhood (Left)

ally expensive, we compare changes not of pixels but of clusters of similar pixels that are obtained using SLIC (see Section 5.3 for more information). In Figure 5.11, the shadow of buildings is perceived as a high indicator of price, again for the Time Square. Unsurprisingly, this area of Manhattan has mostly positive Shapley values, given that they reflect the contribution compared to an average image price. Such price differences, however, are also detected in lower price areas, consequently with smaller Shapely values. This indicates that the model correctly extracts more information from images, which can be visualized intuitively with Shapley values. In Figure 5.12, this is highlighted as the model detects a positive effect of green areas/parks on house prices, especially when paired with residential areas, while large roads have a strong negative impact on price.

All in all, the features extracted by the hybrid model have an economic interpretation for house prices, which is reflected in their strong predictive performance on the latter. In addition to that, the contributions of features for each image can be easily highlighted, which can be used for understanding the impact of individual images and areas in NYC. Finally, the predictive performance of the hybrid model beats the benchmark of using crime directly in the majority of cases, which is why it would be interesting to expand the model of extracting a proxy for crime to other cities where such a benchmark (i.e. geo-coded crime numbers or additional covariate information) is not available. The next section focuses on such a task, namely predicting house prices for Philadelphia.

5.4.3 Predictive Performance for Philadelphia

In this section, we extend our hybrid model to Philadelphia, which is done using image data obtained again from the *Google Maps Static API* for Philadelphia and extracting the features using this data with the CNN trained on NYC image data and crimes. We then use price data obtained from Real Estate Transfers 2018-2019 in Philadelphia[14], which, after removing missing values and data errors (e.g. unrealistically low transfers $\leq$ 100 USD), amount to roughly 86,000 transfers coded with exact GPS coordinates. We use the assessed fair market value as a measure of property price and regress it on the features, again with prices on each image (and its 8 surrounding images) aggregated via either median price or mean weighted by inverse euclidean distance to the image center.

Table 5.2: Predictive Power of Trained Hybrid Model on Data from Philadelphia

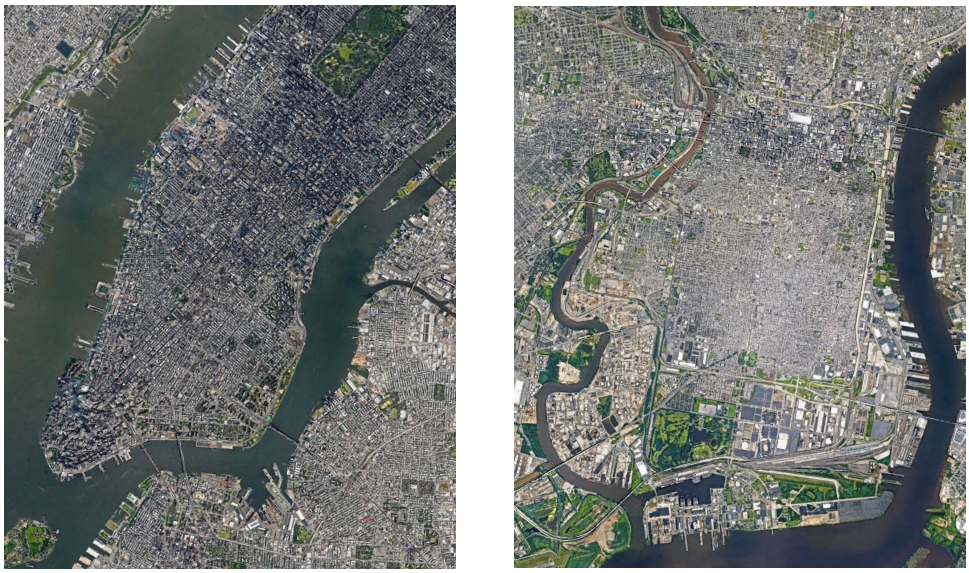
Out-of-Sample-R^2	Ridge	OLS
9REG:		
Median	0.192	0.183
Weighted Mean	0.231	0.222
REG:		
Median	0.001	-0.064
Weighted Mean	0.001	-0.061

Notes: Out-of-sample-R^2 for the CNN model trained on NYC crimes (9REG or REG) and images pairs. Final estimations on Philadelphia price data are done with either ridge regression or OLS on an image level. Aggregation of prices on an image is done as in the main analysis either using the median or the mean weighted by inverse euclidean distance to the center of the image.

We find that using only the extracted features, we obtain an out-of sample R^2 of around 0.2 depending on aggregation scheme and using the CNN which was trained on predicting the nine most common crime categories, which is summarized in Table 5.2. This is roughly 10pp worse to the results in NYC, which is impressive given that the CNN was trained only on crimes and images from NYC and not Philadelphia. Interestingly, using only the absolute number of crimes does not generalize as well, which indicates that

[14] Available at `https://www.opendataphilly.org/dataset/real-estate-transfers`

the nine categories are necessary to give the CNN enough flexibility. Importantly, the crime structure is similar in the two cities, with crime in Philadelphia being significantly higher on a relative scale, which can be seen in Table 5.5 in Appendix 5.6.1[15]. Crime rates for the different categories are 36% to 492% higher in Philadelphia compared to NYC. Comparing city characteristics, Philadelphia shares some similarities with NYC. Apart from the similar age structure and growth rate[16], they also have a similar city layout, both centers being surrounded by rivers and big water surfaces, and a high-rise formation in the center, which can be seen in Figure 5.13[17].



(a) Center of New York City **(b)** Center of Philadelphia

Figure 5.13: Satellite Images Of Centers of New York City and Philadelphia From 18km

[15]See also at `https://ucr.fbi.gov/crime-in-the-u.s/2017/crime-in-the-u.s.-2017/tables/table-8/table-8-state-cuts`

[16]For more information, see the quick facts on the Census Bureau website`https://www.census.gov/quickfacts/fact/table/philadelphiacitypennsylvania,newyorkcitynewyork/HSD410218`

[17]Images from Google Earth (`https://earth.google.com/`) on 28/10/2020.

5.5 Conclusion

In this paper, we demonstrate how to leverage neural networks and image data to improve existing models for predicting property prices. Using state-of-the-art image recognition techniques, we extract information on crime from satellite images to construct new proxy-features for crime that predict property prices well. With these features, we capture more information than with crimes itself while maintaining good interpretability in a final linear model stage. For the task of predicting house prices, we are able to both outperform baseline linear methods which take into account crime information directly as well as the benchmark of employing image information directly within a nonparametric CNN.

We show that our extracted features have interpretable structures for crime by interpreting the underlying CNN and its predictions. These features are more robust to changes in including additional covariates, while being easily adaptable to different cities as well. For the case of Philadelphia, we show the robustness of the feature extraction by only taking into account new satellite images to generate new features without costly retraining a CNN. The feature keep good predictive performance in a simple linear model step. For future research, it would be interesting to extend the model to forecast on a time series level. Allowing for changing structure, however, would be harder as one would need more satellite images, especially high resolution images that are time-coded. These are especially hard to obtain on a consistent scale.

5.6 Appendix

5.6.1 Tables and Figures

Table 5.3: Descriptive Statistics for the NYC Property Prices

	Min	1%	Median	99%	Max	Quantile at 3500 USD
The Bronx	2.56	31.05	248.60	411.11	27050.75	1.00
Brooklyn	2.12	30.14	486.59	1100.89	21600.00	1.00
Manhattan	1.09	68.36	1503.01	2645.96	27608.67	0.96
Queens	3.10	45.29	437.69	768.45	512500.00	1.00
Staten Island	10.55	69.66	338.95	494.73	1875.00	1.00

Notes: Values in column one to five represent property prices per gross square foot at the respective quantile. In the last column, the respective quantile at price 3500 USD is indicated, which is the cutoff for the histograms in Figure 5.14. All values are rounded.

Table 5.4: Overview of the 9 Crime Classes

leading internal code number	Type of offenses	5 most occurring crimes (no. of occurrences)
1	Offenses involving physical injury, sexual conduct, restraint, intimidation	Assault 3 & related offenses (521470) / Felony assault (190216) / Miscellaneous penal law (84016) / Sex crimes (39049) / Rape (13014)
2	Offenses involving damage to and intrusion upon property	Criminal mischief & related offenses (493471) / Burglary (171449) / Criminal trespass (59178) / Arson (12083) / Miscellaneous penal law (4003)
3	Offenses involving theft	Petit larceny (824298) / Robbery (182646) / Petit larceny of motor vehicle (696)
4	Offenses involving theft (grand)	Grand larceny (421768) / Grand larceny of motor vehicle (84956) / Possession of stolen property (26953) / Unauthorized use of a vehicle (14634) / Other offenses related to theft (11744)
5	Offenses against public health and morals	Dangerous drugs (316692) / Miscellaneous penal law (5593) / Gambling (2109) / Sex crimes (1263) / Prostitution & related offenses (824)
6	Offenses Against Public Order, Public Sensibilities and the Right to Privacy	Harassment 2 (601329) / Offenses against public order sensibility and privacy (259058) / Sex crimes (14277) / Miscellaneous penal law (5612) / Offenses related to children (1269)
7	Offenses against public administration and public safety / provisions relating to firearms, fireworks, pornography equipment and vehicles used in the transportation of gambling records	Dangerous weapons (119193) / Offenses against public administration (101718) / Theft-fraud (49720) / Forgery (47961) / Frauds (32036)
8	Other	Administrative code (10983) / Other state laws (non penal law) (4586) / Nys laws-unclassified felony (4359) / Alcoholic beverage control law (841) / Agriculture & markets law-unclassified (346)
9	Vehicle and Traffic Regulations	Intoxicated & impaired driving (69048) / Vehicle and traffic laws (60628)

Notes: Code numbers as in the internal classification of the NYPD. Types of offenses are matched to the penal law extracted from the New York Senate penal laws (`https://www.nysenate.gov/legislation/laws/PEN/P3`).

Table 5.5: Crime Statistics for Philadelphia and NYC Published by the FBI in 2017

Crime per 100K in 2017	Philadelphia	New York
Population	1,575,595	8,616,333
Violent Crime	948	539
Murder and manslaughter	20	3
Rape	75	28
Robbery	382	162
Aggravated assault	470	346
Property crime	3,063	1,449
Burglary	418	129
Larceny-theft	2,297	1,253
Motor vehicle theft	348	67
Arson	26	/

Table 5.6: Additional Predictive Power Results With OOS-R^2 on Image Level

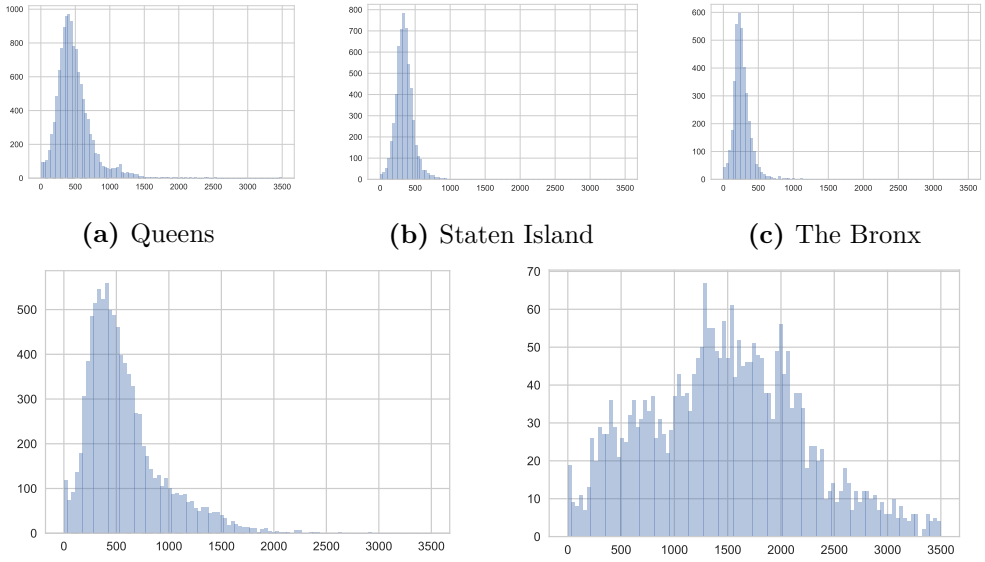
	With X_i		Without X_i	
	9REG	REG	9REG	REG
Crimes_lm.OOS-R2	0.546	0.447	0.336	0.106
Crimes_ridge.OOS-R2	0.339	0.173	0.292	/
Crimes_rf_feat.OOS-R2	0.777	0.709	0.443	-0.460
Crimes_boost_feat.OOS-R2	0.770	0.769	0.389	0.102
Features_lm.OOS-R2	0.566	0.528	0.365	0.313
Features_ridge.OOS-R2	0.393	0.328	0.377	0.312
Features_rf_feat.OOS-R2	0.732	0.723	0.344	0.248
Features_rf_raw.OOS-R2	0.553	0.578	0.287	0.226
Features_boost_feat.OOS-R2	0.790	0.786	0.391	0.255
Features_boost_raw.OOS-R2	0.795	0.781	0.352	0.204

Notes: The first part denotes whether original crime features (*Crimes_*) or extracted features (*Features_*) were used. The second part shows the method that was used for prediction, where *lm* stands for a linear model, *ridge* for a ridge regression where tuning parameters are selected using 10-fold cross validation, *rf* for a random forest with 500 trees, and *boost* for XGBoost using decision trees with a learning rate of 0.08, maximum depth between 4-6 for each tree, a minimum required child weight between 100-200 (corresponds to the number of instances in a node for this task), and 50 weak learners (tuned by CV to prevent overfitting). For some random forest and XGBoost models, we also use the raw 750 features from the CNN (*raw*) instead of PCA-extracted 100 features (*feat*). The ridge model ran into numerical problems for the case using only one the absolute number of crimes without covariates.

Table 5.7: Additional Predictive Power Results With OOS-R^2 on Zip-Code Level

	With X_i		Without X_i	
	9REG	REG	9REG	REG
pca_ridge_median.OOS-R2	0.765	0.748	0.644	0.765
baseline_ols_median.OOS-R2	0.716	0.701	0.570	0.179
Random_Forest_median.OOS-R2	0.647	0.655	0.497	0.510
XGB_median.OOS-R2	0.792	0.799	0.606	0.684

Notes: All methods are run on median aggregated prices per zip-code (n=152). XGBoost is run with 250 weak learners, a maximum tree depth of 2, and learning rate of 0.08. Random forests are trained on 500 trees.



(a) Queens **(b)** Staten Island **(c)** The Bronx

(d) Brooklyn **(e)** Manhattan

Figure 5.14: Histogram of Property Prices per Gross Square Foot in the Five Boroughs of New York City

Note: Each histogram is cut off at a price of 3500 USD, which is over the 99% quantile of the whole price data and over the 96% quantile of each borough's prices.

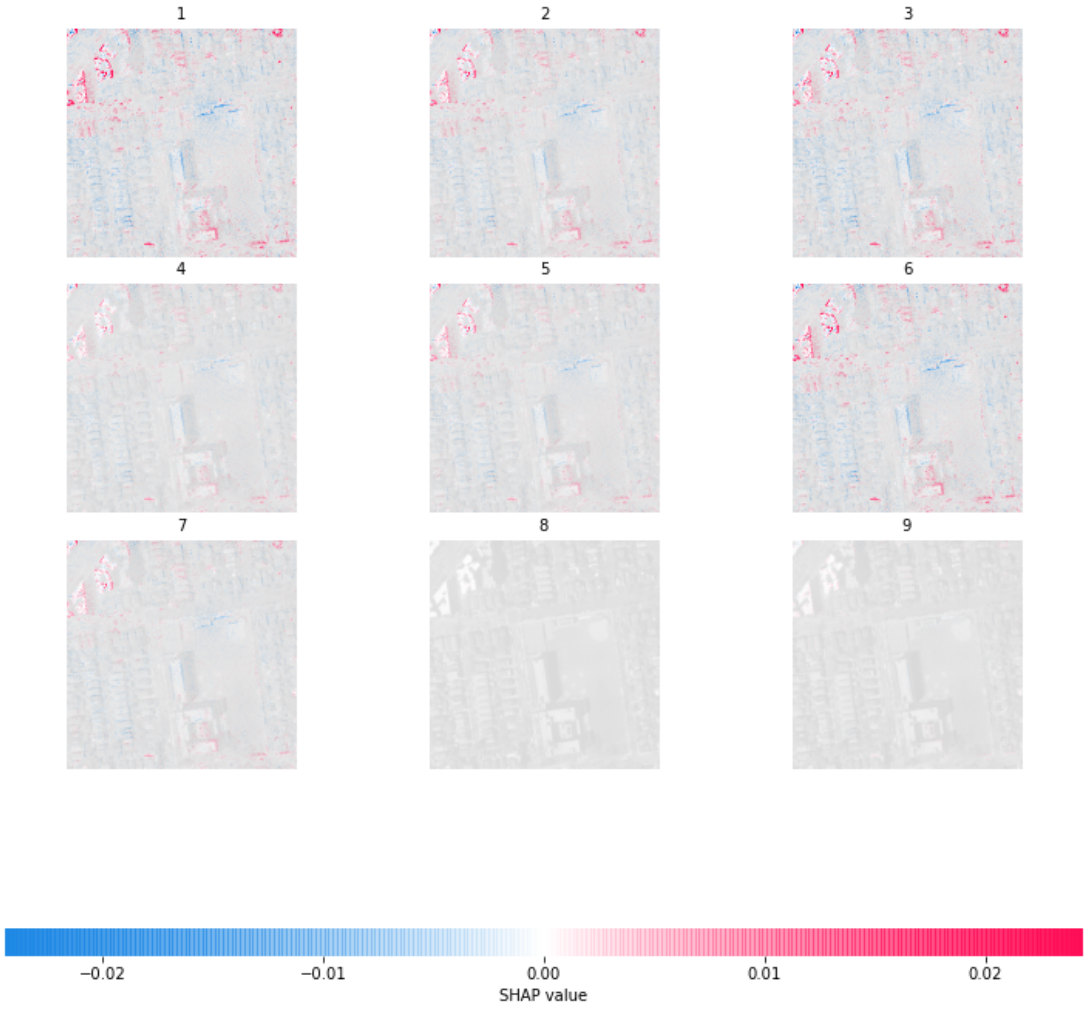


Figure 5.15: Contribution of Input Pixels Towards the 9 Output Crime Classes Using DeepSHAP in a Residential Neighborhood

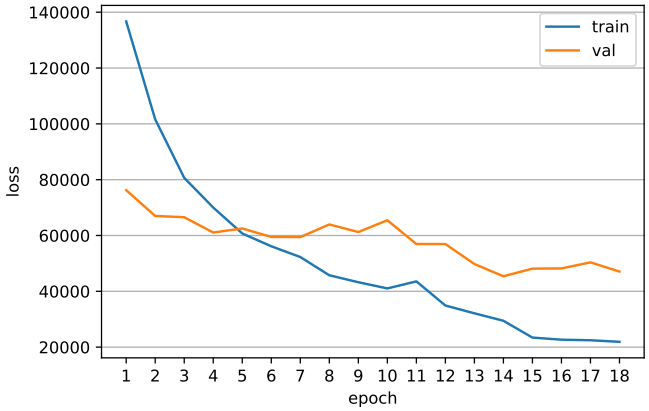


Figure 5.16: Training and Validation Loss for the *CNN*-Model Directly Trained to Predict Prices

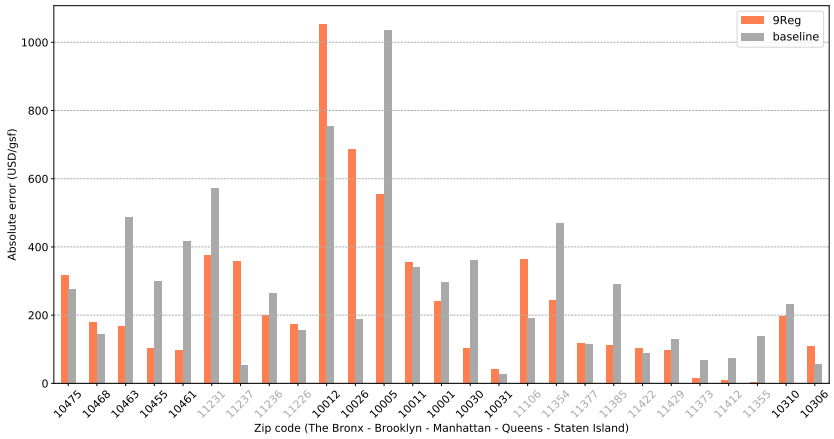


Figure 5.17: Comparison of Absolute Error on Test Set

Note: The shaded labels on the x-axis represent a switch of boroughs.

6 Identifying Important Factors of Property Prices Using Satellite Images and Crime Data

6.1 Introduction

In the past years, research has produced a vast number of new machine learning methods, which are often measured on their predictive performance. Particularly, artificial neural networks have experienced a rise in popularity, especially in image recognition, while now also being more frequently used across disciplines such as economics and finance (see e.g. Hájek, 2011; Jean et al., 2016; Hartford et al., 2017; Lee, 2018). They are easily applied and model non-linear relationships without explicitly having to specify a model. This has helped them to gain considerable popularity, but has the drawback that it is often not well understood how (and if all) the input factors actually influence the outcome of interest. Recent advances to solve these issues are often summarized as interpretable machine learning, for example using Shapley values (Shapley, 1953) by Lundberg and Lee (2017) and Aas et al. (2021).

We investigate house prices in New York City (NYC) applying so-called "features" that are predictive for crime in a generalized additive model (GAM) framework. We assess the contribution of each of these features clearly and show that they have an economic interpretation that is not limited to crimes. While crimes itself have been shown to be endogenous for house prices, we leverage the power of convolutional neural networks (CNNs) and use information from satellite images to obtain features that serve as a rough proxy for crime. This approach has first been proposed by Deuschel et al. (2022), whose model we improve by adding more interpretability while retaining excellent predictive

performance. Intuitively, we use a CNN that is trained to predict crimes from satellite images, which are both available in large numbers and on a fine scale. House prices, on the other hand, are not available in sufficient numbers, which makes it infeasible training a CNN directly to predict them. We then extract features from the above trained CNN on crimes that are (amongst other covariates) used to explain house prices in a semiparametric GAM. One the one hand, employing the features mitigates endogeneity issues that would arise using crime directly. On the other hand, the GAM can handle non-linear features easily while retaining interpretability through the additive structure of the model.

In more detail, the features we extract have lower dimension compared to Deuschel et al. (2022) and thus have a higher information density. This is possible by using a more recent, state-of-the art Inception-ResNet architecture Szegedy et al. (2017). Since we employ GAMs in the final model instead of linear or ridge regressions as in Deuschel et al. (2022), we can use the raw features that model the non-linearity directly and do not need to rely on principal components analysis, which facilitates interpretation[1]. We furthermore show how the features correspond to crime and how they spatially act in comparison to house prices. Using Shapley values taking into account the dependence structure of the features, we can quantify the influence of features on house prices graphically as well as empirically.

Employing machine learning (ML) methods in economics and finance has gained popularity in the recent years (Mullainathan and Spiess, 2017; Athey and Imbens, 2019; Gu et al., 2020), while recently, there have also been approaches using ML-methods in the house pricing context (Yoo et al., 2012; Rischard et al., 2020). More generally, recent research on house prices includes e.g. Anselin and Lozano-Gracia (2007) and de La Paz et al. (2022), where crime has been shown to be an important predictor.

The rest of the paper is structured as follows. Section 6.2 introduces the data we use in our analysis, while Section 6.3 covers the methodology and the model for house prices and crimes. In Section 6.4, we describe in detail the features and how they are extracted from the CNN (Section 6.4.1), before presenting the main model results analyzing its

[1]In the Appendix, we additionally report on effects using principal components in Figures 6.16 and 6.17. They are, however, mostly linear and information largely lies in the first two principal components.

properties in Section 6.4.2 and showing predictive power results in Section 6.4.3. Section 6.5 concludes.

6.2 Data

In our analysis, we obtained data from four main sources (using the same data set as Deuschel et al. (2022)). The satellite data was taken from Google using the Maps Static API[2], while we removed the google watermark and rescaled the image. The data we use corresponds to all areas that lie in the zip-code range of NYC, which means that we remove images that only contain water and images from other nearby urban areas (e.g. New Jersey). As in Deuschel et al. (2022), we also use flipped and rotated versions of each image for training of the neural network, which is useful since it counters overfitting Perez and Wang (2017). In total, we have $n = 6634$ observations for the final models and $n_{aug} = 26536$ augmented images for training of the CNNs.

For our dependent variable of property sales in NYC, we work with official data from 2018 published by the NYC Department of Finance [3]. It contains all tax-relevant properties sales in NYC including square footage, address, and other relevant factors such as building year of the property. For our analysis, we divide the raw price by the square footage of the property to obtain comparable results for properties of differing size. Here, we discard prices with either zero gross-square-footage or missing information in other variables, such as the age of the property, as well as observations with an unrealistically low price per square foot of less than 10 USD/sqf. This is in line with Rischard et al. (2020), who use similar data and deselect all prices lower than approx. 20 USD/sqf. Furthermore, we map the prices to each image by taking the median over all prices in a certain range of the center of each image in the following way. For image i, we use the number of prices that lie in a Manhattan distance of 1.5 times the length of an image. More formally, let $dis_{ki} = |lat_k^{raw} - lat_i^{cent}| + |lon_k^{raw} - lon_i^{cent}|$ be the distance of raw price pr_k^{raw} to image center $cent_i$ with corresponding lat-lon coordinates. We then use the median of set $D_i = \{pr_k^{raw} : dis_k i < 1.5 len_{image}\}$, where len_{image} is the

[2] `https://developers.google.com/maps/documentation/maps-static/intro`, scale 1, zoom 17. This returns an RGB-image with 3x400x400 pixels, with rounded size of 350x350 meters.

[3] `https://www1.nyc.gov/site/finance/taxes/property-annualized-sales-update.page`

length of one image in lat-lon format (which corresponds to roughly 350 meters), as the price $pr_i = median(D_i)$ of image i. Since prices have a highly skewed distribution with large outliers, we decide to use the log-price per square foot as our dependent variable, i.e. we use $pr_i^{log} = log(pr_i)$. Panel (b) of Figure 6.1 shows the cleaned prices before log-transformation[4], where we see quite heavy right tails, which largely results from higher prices in Manhattan. We furthermore investigate the number of prices per image, which can vary considerably depending on different areas indicating that some observations could have a higher amount of information for our analysis. Panel (a) of Figure 6.1 visualizes this property, where for some observations (i.e. images), our dependent variable is composed of only very few price observations (up to only one price), while for a few large outliers, we observe a large number of more than 750 prices per image, with a median of 66 prices per image.

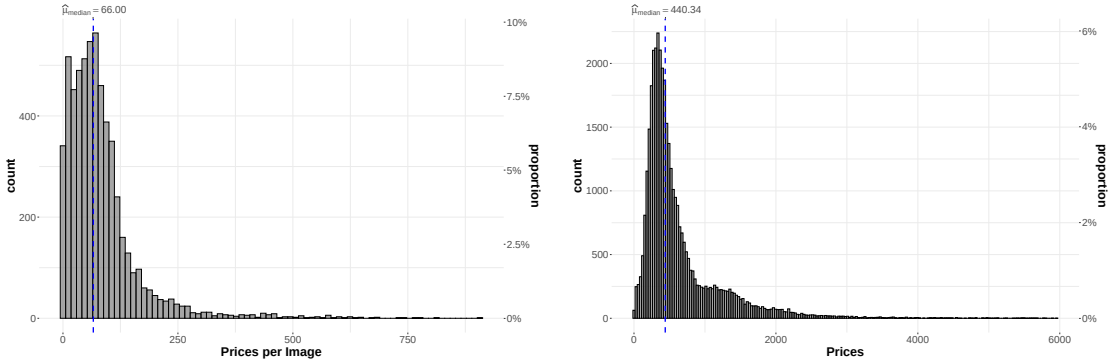
The crime data include all crimes recorded by the NYC Police Department from 2008-2017 and can be found in more detail in Table 6.3. We analyze nine crime classes that include very different types of crimes, from traffic violations and theft to armed robberies, assault, and murder. This amounts to around 5 million observations ($k = 1, \dots, K$) of crimes over the full, pooled, time period. The crimes have a time-stamp and exact geo-location, making it easy to link crimes to an image which we obtain by assigning crime k to its closest image i by taking the Euclidean distance. We then count the number of crimes in each picture for each category for our crime measures $cr_1, \dots, cr_9$.

Additionally, we employ the same covariates as in Deuschel et al. (2022) for an image level, which includes the distance of each image center[5] to the next subway station, to the Empire State Building, to the next fire station, to LaGuardia Airport, and John F. Kennedy International Airport (JFK)[6], and the mean of the age of the properties on each image (in years).

[4]We leave out the 79 highest prices that would distort the plot, which go up to over 27 USD/sqf).

[5]Here, the distance is defined as the Euclidean distance on the level of lat-lon coordinates. 0.01 in this unit corresponds to roughly 1km in NYC.

[6]We use data on the distance based on *NYC Open Data* (`https://opendata.cityofnewyork.us/`).



(a) Histogram of Number of Sales per Image

(b) Histogram of Cleaned Sales Prices

Notes: Properties with prices smaller than 10 USD/sqf are excluded. For visibility reasons, prices larger than 6000 USD/sqf (79 most extreme prices) are not plotted here.

Figure 6.1: Descriptive Statistics for House Prices

6.3 Methodology

To estimate the effect of crime and different covariates on house prices, we use generalized additive models (GAM), with which we can model the expected non-linear effects of crime. It is not feasible, however, to use crime numbers directly, since they have been repeatedly shown to be endogenous for prices (see e.g. Ihlanfeldt and Mayock (2010) for an overview). Instead, we extract features x_j that are predictive for crime from image data using convolutional neural networks (CNNs). We model the log price per square foot y_i in the following model:

$$y_i = a + \sum_{j=1}^{J} s_j^{(x)}(x_{ji}) + \sum_{h=1}^{H} s_h^{(z)}(z_{hi}) + \epsilon_i \ , i = 1, \ldots, n \tag{6.1}$$

$$x_j = (x_{j1}, \ldots, x_{jn}) = G_j(CNN_v, img_1, \ldots, img_n) \ , \tag{6.2}$$

where $x_j, \ j = 1, \ldots, J$ are extracted features that are predictive for crime, $z_h, \ h = 1, \ldots, H$ are covariates that are predictive for house prices, a is an intercept, and ϵ_i is a mean-zero error term. $s_j^{(x)}$ and $s_h^{(z)}$ are non-linear functions that we estimate in Section 6.4 using penalized cubic splines as basis functions, with maximum degree of freedom $df = 10$. Computationally, we use penalized iterative reweighted least squares (PIRLS)

for estimation of the spline coefficients, while the optimal penalty parameters for the shrinking of the complexity of the splines (i.e. degrees of freedom) are computed using the fast restricted maximum likelihood (fREML) algorithm (see Wood et al. (2015, 2017); Li and Wood (2020) for details) implemented in the `R`-package `mgcv` (Wood et al., 2017).

The function G_j is a function that returns the jth n-dimensional feature vector from the last fully-connected layer before the output of CNN_v. This vector consists of the value of the jth feature for each of the $i = 1, \ldots, n$ images. The crucial part of this feature-extracting function is obviously the neural network CNN_v. This idea was originally developed in Deuschel et al. (2022), who, however, focus more on the predictive ability of the model. We employ different architectures which always consist of the Inception-ResNet Szegedy et al. (2017) with two additional fully connected layers. As input, we use RGB-satellite images with dimension 400x400x3, and train the network to simultaneously predict the number of crimes $cr_1, \ldots, cr_9$ (see Table 6.3 for details) on an image. We alter the number of neurons in the last fully connected layer which corresponds to the feature. For our main analysis, we will use $v = 20$ neurons in the last layer, and 750 neurons in the second last layer. We expect this to be large enough to capture important drivers of the 9 crime classes after being trained. Figure 6.2 visualizes our employed architecture, which is similar to Deuschel et al. (2022). Note that during the training, we use dropout layers after the pre-trained network and after the two fully connected layers to counteract overfitting. We furthermore train the top-most layers of the pre-trained network in a final step.

Having extracted the features, the final GAM estimating the intercept a and the functions $s_j^{(x)}$ and $s_h^{(z)}$ using the The PIRLS fitting procedure can be described as follows. For ease of notation, stack both x_j and z_h together to $Q = (x_1, \ldots, x_J, z_1, \ldots, z_H) \coloneqq (q_1, \ldots, q_V) \in \mathbb{R}^{n \times V}$ and do so similarly for $s_j^{(x)}(x_j)$ and $s_j^{(x)}(z_h)$ and obtain $s_Q = (s_1^{(x)}(x_1), \ldots, s_J^{(x)}(x_J), s_1^{(z)}(z_1), \ldots, s_H^{(z)}(z_H) \coloneqq (s_1, \ldots, s_V)$. s_v can now be expressed as the sum of basis functions b_{cv} with coefficients β_{cv}, where for $q_v = (q_{v1}, \ldots, q_{vn})^\mathsf{T} \in \mathbb{R}^n$, b_{cv} returns $b_{cv}(q_v) = (b_{cv}(q_{v1}), \ldots, b_{cv}(q_{vn}))^\mathsf{T} \in \mathbb{R}^n$. Then, we obtain for $s_q(q_v)$:

$$s_q(q_v) = \sum_{c=1}^{C_v} b_{cv}(q_v)\beta_{cv} \coloneqq Q_v\beta_v \ , \tag{6.3}$$

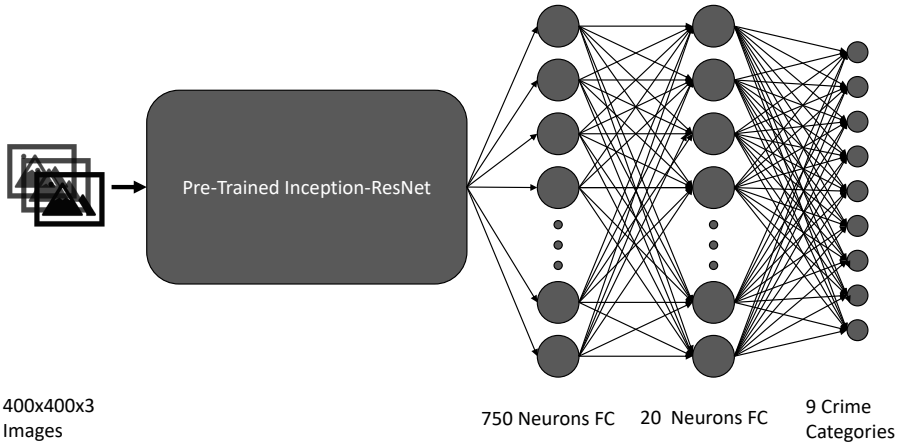


Figure 6.2: Structure of the Employed Neural Network Architecture

Note: Starting with an Inception-ResNet pre-trained on the ImageNet data, we train two fully connected (FC) layers, one with 750 neurons and a second one, our feature layer, consisting of 20 neurons.

where $Q_v \in \mathbb{R}^{n \times C_v}$ contains the values of q_v evaluated at the basis functions and $\beta_v = (\beta_{c1}, \ldots, \beta_{cV})^\mathsf{T}$. For $\tilde{Q} = (a, Q_1, \ldots, Q_V) \in \mathbb{R}^{n \times P}$, $P = 1 + \sum_{v=1}^{V} C_v$, β as the stacked vector of 1 (for the intercept) and all β_v, and $y = (y_1, \ldots, y_n)^\mathsf{T}$, the algorithm minimizes:

$$\min_{\beta} \|y - \tilde{Q}\beta\|^2 + \sum_{v=1}^{V} \lambda_v \beta_v S_v \beta_v \ , \tag{6.4}$$

with S_v as the identity matrix of dimension C_v. To compute the penalty parameters λ_v, we use performance iteration in combination with fREML, which is much faster for large data sets (see e.g. Wood et al. (2015) and Wood et al. (2017)) selecting λ_v in each iteration of the algorithm.

To analyze the stability of the GAM estimation in Section 6.4, we will refit the final model B times using subsampled observations. The main comparison for this will be the estimated effective degrees of freedom (EDF) of our estimated model, which will be a

result of the maximum dimension for each spline and the optimal shrinkage parameter $\lambda_i j$ for basis function i in covariate j. More specifically, the EDF can be calculated as the trace $tr(F)$ of a matrix F, where F is the hat-matrix of the for the model:

$$F = (\tilde{Q}^{\mathsf{T}}\tilde{Q} + S_{opt})^{-1}\tilde{Q}^{\mathsf{T}}\tilde{Q} \ , \tag{6.5}$$

where $S_{opt}^{-1} = \sum_{v=1}^{V} \lambda_v^* S_{v0}$ at the estimated λ_v^*, where S_{v0} is a zero padded version of S_v with dimension $P \times P$. To obtain the EDF for the final variables q_v, we simply have to sum the diagonal elements at the respective positions, i.e. for the first variable x_1, we sum from $p = 2$ to C_1. Tests for the significance of EDF from zero are likelihood ratio tests testing $E[\hat{\beta}_k] = 0$ based on a Bayesian Random Effects model, and use $tr(2F - FF)$ instead. For details, see Wood (2013).

Finally, we also employ the Kernel-SHAP from Lundberg and Lee (2017) to explain contributions of the features and covariates on single observations and predictions, but use the extension from Aas et al. (2021) accounting for the dependence structure of variables. This Kernel-SHAP is model-agnostic as it linearizes the model for Shapely-Value computation. In our prediction setup, each Shapley value ϕ_v contributes linearly to the final prediction. Following Aas et al. (2021), write $E[y] = f(Q)$ for the estimated GAM, and define for a new value Q^*:

$$f(Q^*) = \phi_0^* + \sum_{v=1}^{V} \phi_v^* \ , \tag{6.6}$$

where $\phi_0^* = E[f(Q)]$ and ϕ_v^* is the Shapley value, which intuitively can be described as the weighted contribution that a variable v adds to the model output $f(Q^*)$, given all possible subsets of variable combinations. For more details on the implementation, see especially Aas et al. (2021), Lundberg and Lee (2017), Shapley (1953), or Deuschel et al. (2022).

6.4 Results

In this section, we present our approach of extracting features via CNNs that are predictive for crimes, but in comparison to the latter, are not endogenous for house prices. We use generalized additive models to show the impact of the features on house prices and

further investigate spatial heterogeneities in the influence of variables on prices using Shapley values. Finally, we show that this methodology retains excellent forecasting performance in various setups.

6.4.1 Crime Endogeneity and Feature Extraction

In the literature on house prices, crime has repeatedly come up as an important influencing factor. Intuitively, people would value property in a relatively safe area higher, thus indicating that crime should be included in house-pricing model. However, there have been various studies showing that crime can be endogenous through various channels (Anselin and Lozano-Gracia, 2007; Ihlanfeldt and Mayock, 2010; de La Paz et al., 2022). Specifically, Ihlanfeldt and Mayock (2010) characterize the endogeneity through several channels, which act in both positive and negative ways: Crime could be positively correlated with price, if, for example, higher valued neighborhoods attract more crime (higher payoff), have higher reporting rates, or are less crowed possessing features that make crime easier (large windows, secluded property). On the other hand, wealthier neighborhoods can afford more security measures, or lower priced neighborhoods could attract more crime in general, which again correlates crime negatively with price.

We train a CNN to predict crime from satellite images and extract features which describe the general nine categories of crime classified by the New York City Police Department. With this, we think that we are less affected by the endogeneity issue of crime. The satellite images are static and crime information is only used up to one year before our price data starts. This gives way to the argument that crime changes more dynamically than the surrounding shape of the city and thus, this time delay secures exogeneity. Specifically, we only use the satellite image of a certain area to extract the features, which can be assumed as static and inelastic in comparison to crime in our short time horizon, while the latter can change more rapidly. The properties of an image (e.g. whether we have high rise, parks, commercial property) are mapped in a nonparametric way by the CNN to the features we extract, which makes them act as a sort of instrument, although not in the classic way.

More specifically, we construct the features as the output of the last fully connected layer of our CNN with 20 neurons in the last layer. For each input image, we therefore

obtain 20 feature values that are strictly positive (or zero), which is due to the ReLU-activation[7] function, which maps all negative values to zero. The features thus contain information about certain image properties that are predictive for crime. Intuitively, these properties also have an impact on house prices but do not suffer from the endogeneity problems stated above. Interestingly, more than half of the features are zero, indicating that the nine crime categories can be represented by only a few characteristics of the images. There are some images with particularly high feature values. We select 19 of these images, mostly from Manhattan and the Bronx, and explain their influence more in detail in Section 6.4.2. See also Figure 6.8 in the Appendix for a detailed location of these on a map of NYC. Figure 6.3 depicts Shapley values using the Kernel-SHAP from Lundberg and Lee (2017) for the feature values given the image input for three of the selected images. This is done by separating each image into clusters of size 40 using an adapted version of the SLIC algorithm (Achanta et al., 2012), and then using each cluster as a covariate for which the contribution to the feature values can be depicted. This means that we can identify regions on an image that contribute a lot to the values of the features. We can see that the contributions of pixel-clusters vary depending on the feature that is used, which indicates that our feature selection works as intended, i.e. identifying different classes of crimes by using the properties of the image. We can see, for example, that roads, the shadows of buildings, rooftops, parks, and trees are important factors in the composition of the features.

To see how this translates into the nine crime categories, we look at the final predictions of the CNN-model. Figure 6.4 shows the mean prediction of each (non-zero) feature for each crime category, i.e. the mean contribution of a feature to a certain crime category [8]. We can see that there is considerable heterogeneity in the mean prediction, both across categories (as expected since crime counts differ per category) and features, which shows that the features are working as intended, i.e. having distinct influences over different crime categories. For violent crimes, which can be roughly summarized by category 1,4, and 5 (see also Table 6.3 for details), we see the strongest influences in Features 10,17, and 18, while for property damages (category 2), Features 1,13, and 18 are most

[7] The rectified linear unit function is given as $f_{relu}(x) = \max(0, x)$.

[8] see Figure 6.14 in the Appendix for more details on the distribution of features and 6.15 for the pure weights.

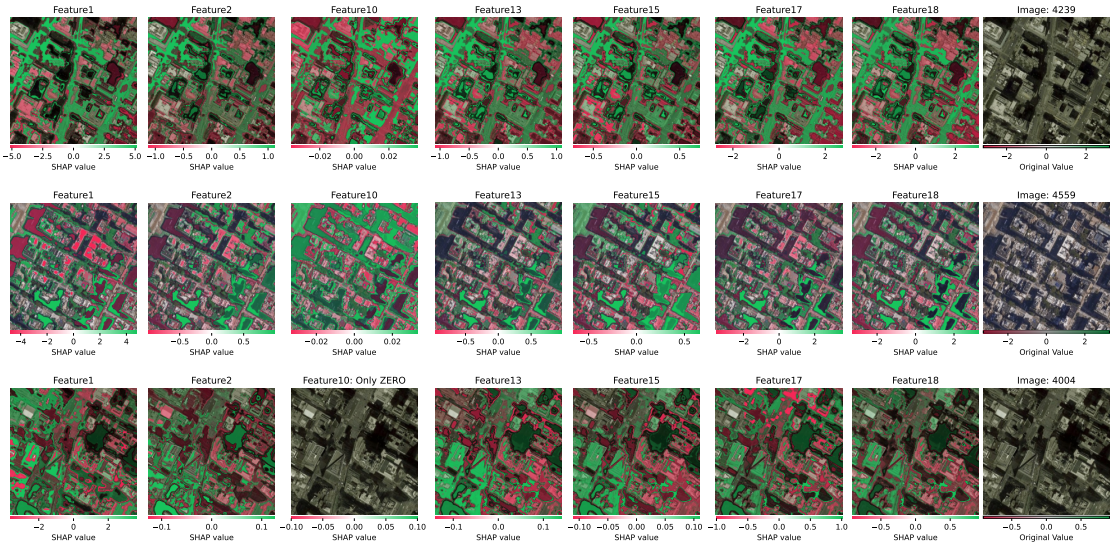


Figure 6.3: Illustration of Feature Shapley Values for Selected Images

Note: The top image is located at the southern west corner of Central Park, the middle image is located at the Upper East Side, and the bottom image is located at Times Square in Midtown Manhattan.

important. For little theft and safety (categories 3 and 7), again Features 1,13, and 18 have high predictions, although to a lesser extent as Feature 17, which is highest here. For traffic related crimes (category 9), the group of Features 1,13, and 18 has a negative impact, while Feature 17 again has a positive effect.

Looking more feature specific effects, Features 1,13, and 18 often act together, and are positive for theft, public safety, and property damages, while having negative influence on traffic related crimes. This could be evidence that these features are detecting image structures that are simultaneously predictive for the above categories, indicating that they do not appear in the same images as most traffic related crimes. This points to the conclusion that crimes related to public safety, theft, and property damages appear together and spatially distant to traffic crimes. One explanation for this would be that traffic related offenses occur in areas with high traffic, while property crime or offenses against public safety occur rather in more remote, residential areas with less traffic. For violent crimes, on the other hand, Features 10 and 17 often predict high positive values for violent crime and simultaneously negative values for public order. Again, this points

towards the conclusion that crimes related to public order, which includes harassment, intrusion of privacy, or offenses related to children, occur in different areas than violent crimes such as assault, grand theft, burglaries, arson, and felony assaults. To further investigate the location of crime, we inspect the predictions of each feature on a spatial grid of the city and highlight where different categories of crime are most prevalent. We find that features indeed predict different crime hotspots as expected and the CNN-model using only satellite image can thus approximate crime quite well.

To visualize this, we plot heatmaps of the 1000 highest feature predictions, i.e. the feature values multiplied by their weights for each crime category in Figures 6.5 and 6.6. We can see that apart from the absolute predicted numbers that vary over each category as with the underlying crime data, the hotspots of crime vary quite substantially per category. Although they are mostly being focused around Manhattan (especially Midtown Manhattan and Uptown Manhattan north of Central Park) and the Bronx, where the most crimes were reported in general, we see some patterns over the different categories. While violent crime in general seems to be most heavily prevalent in the Harlem area and in certain areas in Midtown Manhattan, Grand Theft and crime against Public Health and Morals are more concentrated only in Harlem. This seems reasonable, as very busy areas as Midtown Manhattan with large businesses are likely more controlled by police forces, making them less prone for offense such as Grand Larceny in comparison to more remote areas in Harlem or the Bronx. The latter two also have the highest concentration of General Theft and offenses regarding Vehicle and Traffic. Property crime and offenses regarding Public Order and Privacy, on the other hand, are predicted to be highest in the Bronx, which is in line with our interpretation that violent crime and property crime are occurring mostly in different, detached areas. We cannot fully verify that traffic offenses are spatially detached from property, theft, and public safety crimes as indicated above. This might especially be caused by traffic offenses such as intoxicated and impaired driving or other violations against traffic laws, which are obviously correlated to the amount of traffic and therefore also occur in more busy business-heavy areas (e.g. Midtown Manhattan). This all highlights the high crime density in the Bronx and Harlem and also explains why the features itself suffer to some degree from collinearity, which we investigate more in detail in Section 6.4.2.

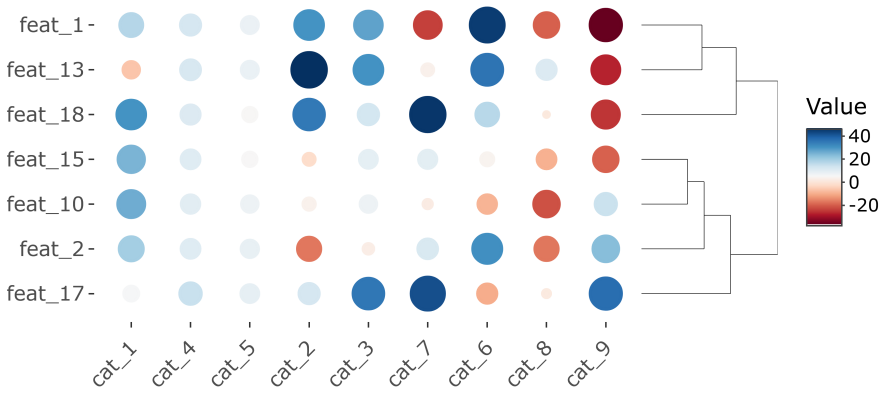


Figure 6.4: Mean Prediction of All Non-zero Features From InceptionResnetV2-20 CNN for Each of the Nine Final Crime Categories

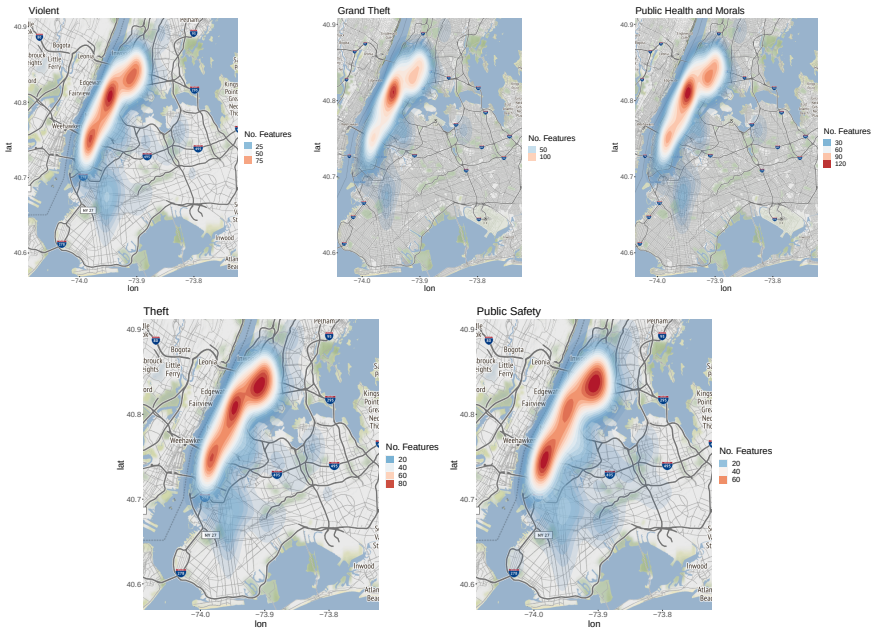


Figure 6.5: Heatmap of Highest Predictions per Feature for Crime Categories Concerning Violent Crimes, Theft, and Safety

Note: The scaling is different on each subfigure since the number of crimes varies strongly in each category.

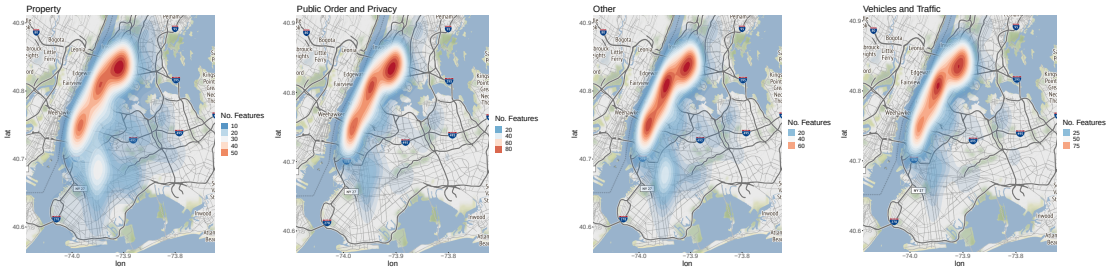


Figure 6.6: Heatmap of Highest Predictions per Feature for Crime Categories Concerning Property, Public Order, Traffic, and Other Crimes
Note: The scaling is different on each subfigure since the number of crimes varies strongly in each category.

6.4.2 Model Estimation and Interpretation

We now present the results of our main model for interpreting the features. We employ a GAM regressing the log house price per square foot on the features from our architecture with 20 features and the covariates from Section 6.2. The main estimates for the functions of the GAM can be found in Table 6.1, where the estimated degrees of freedom are depicted. While the covariates are all highly non-linear, mainly Features 13,15,17, and 18 have non-linear effects based on Table 6.1. Feature 1, 2, and 10 seem to have mainly linear effects. Although the effect for Feature 10 in Figure 6.8 has some non-linear shape, the confidence bands are quite wide. In Table 6.1, we additionally test for the non-linearity of the effects by first estimating a linear model on the full data, and subsequently using the residuals (which contain all information not yet explained by a linear function) as an outcome in a GAM and employ the same tests as in the full model. These tests are likelihood ratio tests as in Wood (2013) and are all highly significant for the full model apart from Feature 2, suggesting there is no significant effect. Figure 6.7 suggests that this again caused by an extremely high variance in the estimates. On the other side, our the p-values of the tests from the residualized model suggest that the effect of Feature1 and 10 are linear. All in all, the results from Table 6.1 indicate that using a GAM instead of a linear model is highly appropriate in our application.

This becomes even clearer when looking at Figures 6.7 and 6.8, which depict the effects of each variable on our price variable over the full range of possible values. There, we can

Table 6.1: Estimated Degrees of Freedom and P-values for the 20-Feature GAM

	Full Model				Residualized Model	
	EDF	Ref.df	F-Stat	p-value	F-Stat	p-value
year_built	7.90	8.68	24.95	0.000	32.95	0.000
dist_jfk	8.79	8.95	120.59	0.000	122.93	0.000
dist_esb	8.36	8.60	495.64	0.000	172.93	0.000
dist_lag	8.45	8.61	116.50	0.000	79.95	0.000
dist_fdp	4.08	5.07	6.19	0.000	6.42	0.000
dist_sub	7.74	8.56	43.05	0.000	29.92	0.000
feat_1	1.04	1.07	6.35	0.013	0.14	0.750
feat_2	1.44	1.75	0.16	0.778	3.16	0.082
feat_10	2.61	3.39	2.73	0.036	0.81	0.502
feat_13	2.76	3.54	7.78	0.000	2.92	0.023
feat_15	3.35	4.27	16.58	0.000	10.78	0.000
feat_17	4.90	5.96	6.29	0.000	6.35	0.000
feat_18	2.20	2.86	4.88	0.002	5.38	0.001

Notes: Estimated effective degrees of freedom (EDF) are calculated according to Section 6.3 and based on the fit of the PIRLS estimator. P-values and test statistics (F-Stat) are for likelihood ratio tests as in Wood (2013) with the null hypothesis that the expectation of functional coefficients are zero and are calculated against Ref.df. The latter are computed as $tr(2A - AA)$, see Section 6.3 for details. The residualized model represents a GAM fit on the residuals of a linear model and therefore tests for the non-linear component of the covariates.

see the non-linearities in the variables very clearly. First of all, note that to be able to identify the functional for each variable, we impose that $\sum_{i=1}^{n} s_j^{(x)} = 0$ for all $j = 1 \ldots, J$, and $\sum_{i=1}^{n} s_h^{(z)} = 0$ for all $h = 1 \ldots, H$. This can be seen at the plots that are always centered around zero. The effects are plotted given that all other terms are set to zero, with shaded areas depicting two standard errors above and below the estimate taken from the Bayesian posterior distribution of the coefficients (see Marra and Wood (2012) for details on the Bayesian Estimation).

The covariates have mostly non-linear effects that seem in line with economic intuition. Properties more recently build have a positive impact on price, with a dip around the

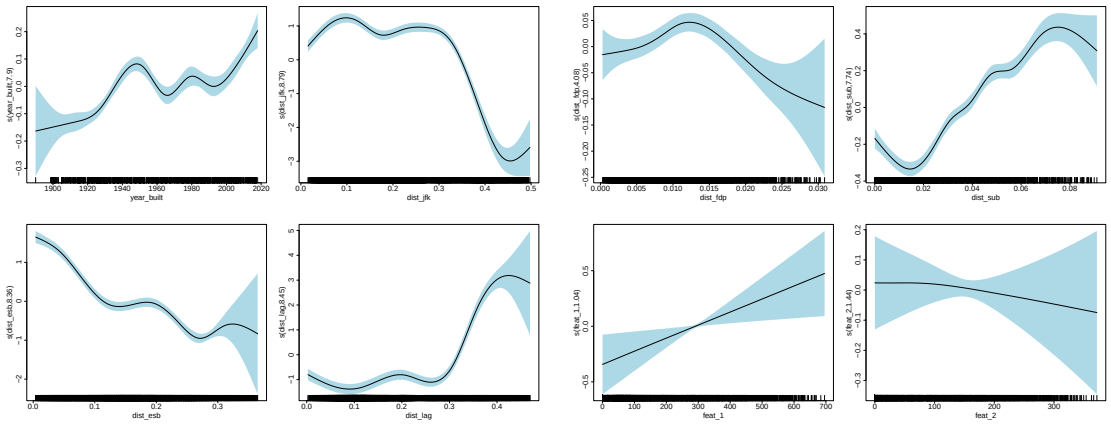


Figure 6.7: Marginal Effects for the GAM Within the 20-Feature InceptionResNetV2 CNN

Notes: GAM with cubic splines trained on covariates and all non-zero features. Black lines show GAM-estimates and shaded areas depict two standard errors above and below using Bayesian variance estimates. Black lines at x-axis are rug plots showing the distribution of data points. Note that the axes are scaled differently to visualize the non-linearities.

1960s, which might be attributed to the type of buildings that were constructed in the post-war period after the second world war. Similar, the closer properties are to the Empire State Building (ESB), the more valuable they are, with a plateau between 0.1 and 0.2 (corresponding roughly to 10-20km airline distance), and then subsequently decreasing. This seems appropriate judging that areas that the furthest away are mostly in Staten Island, while areas in 10-20km of the ESB cover most inner parts of Brooklyn, Queens, and The Bronx. Distance to the JFK, on the other hand, is mostly positive, and dominated by properties on the very outside of NYC in Staten Island or The Bronx (which is far away from JFK) that are lower priced. We see, however, that the effect is increasing for properties up to 10km away from the airport, which is sensible as properties too close to an airport might be unattractive due to noise or infrastructure such as large highways. The distance to fire departments and subway stations have much lower effects in general, while again, being too close to a fire department is less positive, as possible noise complaints play a role. The largest effect is seen in in the middle of the distribution, and goes down when the distance becomes too large, indicating a possibly higher risk of fire damage of a delayed arrival of the fire rescue service. The effects of Features 1,13, and

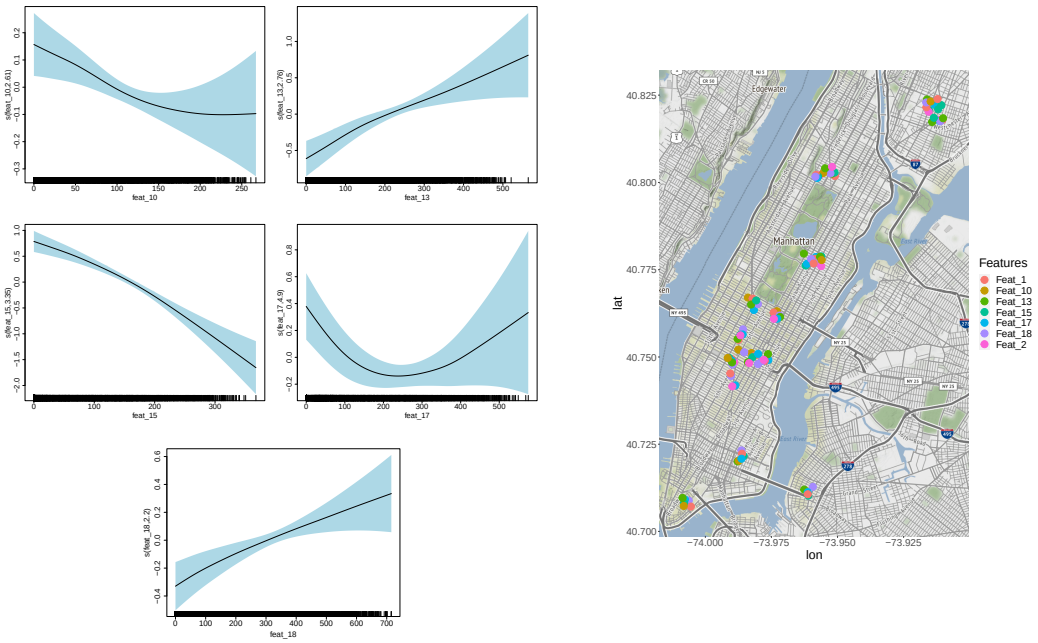


Figure 6.8: Marginal Effects for the GAM Within the 20-Feature InceptionResnetV2 CNN and Map of Largest Features

Notes: Left: Marginal effects for log-price of poisson-loss InceptionResnetV2 20-feature GAM with cubic splines trained on covariates and all non-zero features.Note that the axes are scaled differently to visualize the non-linearities. The rest of the details follow Figure 6.7. Right: Locations of selected images (with slight noise to visualize the different feature positions on the same image) with the largest feature-values from InceptionResnetV2-20 CNN.

18 are all positive, indicating that with increasing feature value, the price will increase. As the features are high in areas where offense such as theft and property damages are higher, while less traffic related offenses are recorded, this could indicate that such areas contain more residential, higher priced property.

Features 10 and 17 both have a similar trend from Figure 6.8, with a negative slope for small feature values and increasingly less negative slope with larger values. These features are mostly high for violent crimes, while being low for offense related to public order (e.g. intrusion of privacy, harassment). We see a sharp decline from zero to up to 100 for both features, which could be explained by even low amounts of violent crime having a large negative impact on price that is stronger than when we move from larger

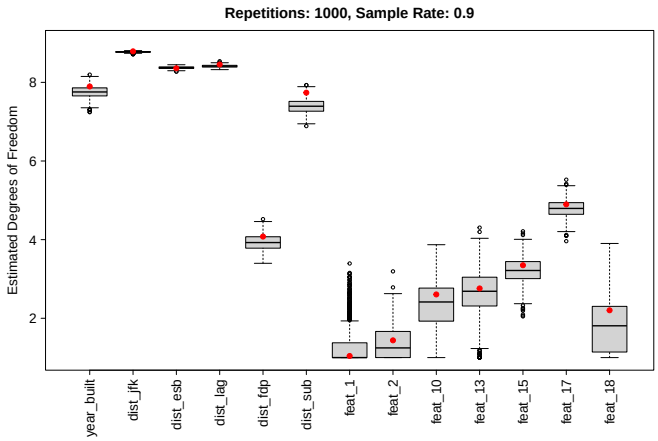


Figure 6.9: Boxplots of Estimated Effective Degrees of Freedom (EDF) for the GAM for Log-Price and Features From the 20-Feature InceptionResNetV2 CNN

Note: The red dots resemble the EDF of the full model, while the boxplots show EDFs over repeated estimations ($B = 1000$) with subsample rate of 0.9.

quantities to even higher numbers. Feature 17, however, rises again at around 250, which can be attributed to influence that Feature 17 also has on category 3 and 7 (little theft and safety), that might have a larger effect here, given the assumption that the effect of violent crime is smaller compared to the effect of offenses related to safety at a high level. Although Wood (2008) state that with the PIRLS algorithm, concurvity does not affect the optimization procedure strongly, we conduct further analysis to check the stability of the EDFs, especially for the features. Figure 6.9 shows boxplots of all EDF that are re-estimated $B = 1000$ times using a subsample of size $N_sub = \lceil 0.9n \rceil$. The results indicate that our models are rather stable, with modest larger uncertainty only for Feature 18.

To obtain some intuition of the mean effects for single images, we compute Shapley values that account for the dependence structure of the features (see Aas et al., 2021). To model the dependence structure, we either estimate a simple Gaussian structure which assumes that the covariates are multivariate normally distributed, or use a Copula approach where we assume that the dependence structure can be modeled by a Gaussian Copula.

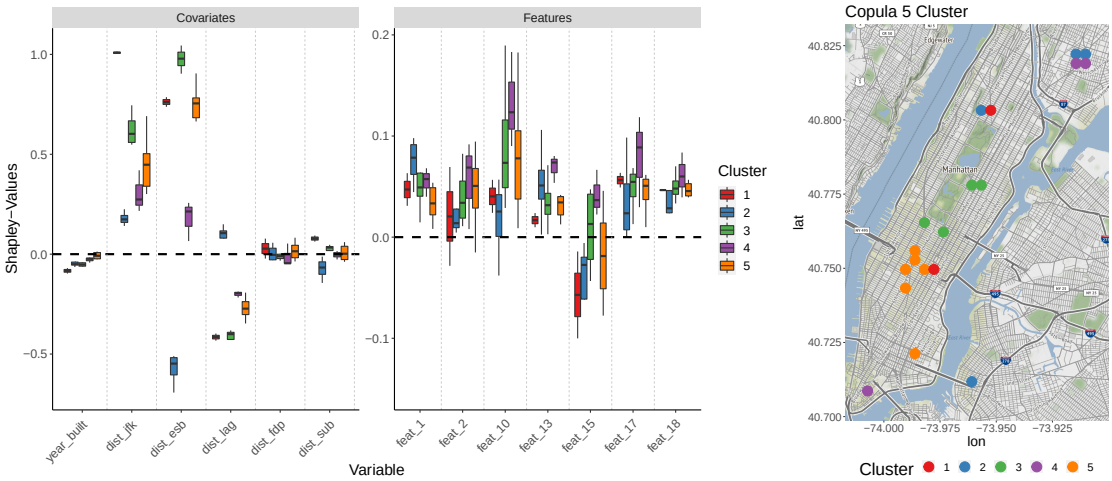


Figure 6.10: Boxplots of Shapley Values for Both the Covariates and Features Depending on the 5 Clusters and Location of Clusters on the NYC Map

Notes: Left: Boxplots of Shapley values for both the covariates and features depending on the 5 clusters. Note the different scaling of the y-axis for covariates and features, which facilitates interpretation. Right: Location of clusters on the NYC map.

We apply this methodology on the 19 selected images from Section 6.4.1, which are visualized on the right side of Figure 6.8, and obtain Shapley values for each of the covariates and features on each image. To visualize the different, non-linear influence of each variable on price, we cluster the 19 images in different groups using k-means clustering with 3 and 5 clusters[9] on all Shapley values, which is depicted in Figure 6.10[10].

To summarize the results, we can see that there is clear heterogeneity between the influence of variables on different images, driven by the location which can be roughly identified by the Shapely values and the resulting clusters. Interestingly, it indicates that the features and covariates indeed identify areas that are similar to each other. Although it is not possible to derive clear directions of all effects in the regions due to the small sample size here, we can identify some trends and plausible clusters. On the right side

[9]The optimal number of clusters is determined for both the Gaussian and Copula approach using the gap-statistic suggested by Tibshirani et al. (2001) and the final clusters are determined using 50 restarts each. We obtain both 3 and 5 Clusters for both the methods respectively, and use both for both methods, which can be seen in Figure 6.11 and 6.13 in the Appendix.

[10]See Figure 6.12 for the same figure with three clusters in the Appendix.

of Figure 6.10, we can see the five clusters on a map of NYC. We identify two main clusters, one in Midtown Manhattan near Times Square (Cluster 5) and the second one at Central Park and Upper East Side (Cluster 3). Furthermore, two smaller clusters more on the outside of Manhattan are identified: Cluster 4 in the Bronx and in Downtown Manhattan, and Cluster 2 in Brooklyn, the Bronx, and Harlem. Finally, Cluster 1 only consists of two points and is quite close to cluster three in Figure 6.13 (bottom-left in the Appendix), where we again plot the clusters, but not a grid of NYC but based on the two largest principal components of all covariates. This shows that even though Cluster 1 is estimated separately, it is quite closely related to Cluster 3.

The left part of Figure 6.10 now shows how Shapley values vary in the different clusters over both covariates and features. First of all, we can see that the distance to the Empire State Building, JFK, and LaGuardia Airport have the biggest average influence, while the other covariates have rather low influence on these selected images. For better visibility, we visualize the features on the right boxplot on a different scale. Feature 10 has the largest positive influence overall, while Feature 15 is the only feature with mostly negative influence. It has to be noted, however, that the boxplots are relatively wide, which is likely due to the small sample size of selected images. Interestingly, Cluster 4 has relatively large positive values for both Features 10 and 17, which were linked to violent crime in Section 6.4.1. Another notable detail is the difference between Clusters 3 and 5 on Feature 15. For the more residential, high priced properties on Central Park and the Upper East Side, this feature has more of a positive mean influence, while for the more business-heavy area in Midtown Manhattan, the influence is rather negative. On the side of covariates that describe the distance from both the airports and the Empire State Building, the effects are plausible as the biggest distinctions are between clusters that are spatially further apart. For example, clusters close to the Empire State Building have the highest positive values on that covariate, while Cluster 2, which is mostly located in the Bronx and Harlem has large negative values. The same distinction can be found for the distance to LaGuardia Airport, although in the opposite direction (i.e. positive values for Cluster 2).

Table 6.2: Predictive Power Results With OOS-R^2 on Image Level Depending on CNN

CNN optimized on:	*MSE*-loss		*Poisson*-loss	
	With Z_i	Without Z_i	With Z_i	Without Z_i
Log Price				
100_layers_Features_GAM_10PC	0.582	**0.297**	0.566	0.218
100_layers_Features_GAM_raw	**0.589**	**0.306**	0.563	0.297
100_layers_Features_lm_10PC	0.467	0.284	0.423	0.204
100_layers_Features_lm_raw	0.471	0.294	0.452	0.297
20_layers_Features_GAM_10PC	0.576	0.262	0.572	0.133
20_layers_Features_GAM_raw	0.577	0.266	0.570	0.143
20_layers_Features_lm_10PC	0.441	0.197	0.414	0.104
20_layers_Features_lm_raw	0.445	0.197	0.414	0.104
750_layers_Features_GAM_10PC	0.575	0.241	**0.584**	**0.343**
750_layers_Features_GAM_raw	-	-	-	-
750_layers_Features_lm_10PC	0.468	0.249	0.460	**0.301**
750_layers_Features_lm_raw	0.337	0.125	0.401	0.270
Crimes_GAM	**0.588**	0.278	**0.588**	0.278
Crimes_lm	0.418	0.148	0.418	0.148
Raw Price				
100_layers_Features_GAM_10PC	**0.768**	**0.402**	0.748	0.301
100_layers_Features_GAM_raw	0.746	0.369	0.669	0.289
100_layers_Features_lm_10PC	0.535	0.354	0.487	0.258
100_layers_Features_lm_raw	0.528	0.353	0.507	0.351
20_layers_Features_GAM_10PC	0.755	0.368	0.743	0.111
20_layers_Features_GAM_raw	0.757	0.376	**0.752**	0.148
20_layers_Features_lm_10PC	0.513	0.275	0.457	0.096
20_layers_Features_lm_raw	0.513	0.275	0.457	0.096
750_layers_Features_GAM_10PC	**0.759**	0.381	**0.766**	**0.455**
750_layers_Features_GAM_raw	-	-	-	-
750_layers_Features_lm_10PC	0.540	0.328	0.540	0.400
750_layers_Features_lm_raw	0.333	0.184	0.310	0.197
Crimes_GAM	0.746	**0.477**	0.746	**0.477**
Crimes_lm	0.545	0.339	0.545	0.339

Notes: Out-of-sample-R^2 rounded to three digits. The dependent variable is the log-price (Top) or raw price (bottom), both per square foot. Measures are giving features of the last fully-connected layer before the output of a CNN consisting of 100, 20, or 750 neurons. For each scenario, we compute generalized additive models (GAM) and linear models (lm), each taking either the raw features or the 10 first principal components. Additional features Z_i are used for model fitting certain scenarios. *Crimes_GAM* and *Crimes_lm* use the true crimes instead of features/principal components. It was not computationally feasible to compute a GAM consisting of 750 features. The best two models in each scenario are marked in bold.

6.4.3 Predictive Power of Models

We conduct various out-of-sample prediction tasks to highlight the power of our feature-extracting approach in combination with the highly-flexible GAM. All in all, we find that our employed model has excellent forecast performance, and is always at least as good as a model using the endogenous crime variables directly for prediction. The main results are summarized in the top part of Table 6.2, while additional results for a model using raw prices per square foot, i.e. without using the logarithm, are found in the bottom part of Table 6.2 .

In both the log-price and raw-price scenario, we employ different architectures varying the number of features $(20, 100, 750)$ and either take the raw features directly or apply principal components analysis on the extracted features before the final model estimation step. There, we compare a GAM as described in Section 6.3 with a simple linear model. Furthermore, we use different loss functions for training the CNN. We either use a poisson-loss as in the main analysis, which accounts for the fact that the number of crimes is always positive, or a classic least squares loss (MSE). Results are also reported using either only the features/principal components or including additional covariates as stated in Section 6.2.

First of all, the model we estimate in Section 6.4.2 ("20_layers_Features_GAM_raw" with Poisson-loss and Z_i) performs quite well and similarly to the other models using GAM, which can be seen in Table 6.2. In general, using the log-price reduces the predictive performance by about $0.1 - 0.2$ in out-of-sample-R^2, depending on the method. This could be caused by numerical issues in forecasting, which are a result of the log-scaling during estimation, which, however, facilitates interpretation and handling of large outliers.

Secondly, using Poisson-loss vs. MSE-loss for training the CNN only meaningfully changes the prediction power for smaller models, meaning that when we only have 20 features, using the Poisson-loss instead of the MSE-loss reduces the oos-R^2. This is especially strong in one scenario when not using additional covariates, where the oos-R^2 is reduced by around 50%. This could be explained by the training process of the neural network, where there are some optimization problems for using small feature layers.

Furthermore, the number of layers does not seem to play a major role for performance for the GAM or when using PCA, while linear models seem to struggle a lot when using

too many raw features, which can be explained by the imposed linear model structure. For the GAMs, there is not much difference, while using PCA components without additional covariates seems to be slightly worse. Again, this could be attributed to information loss, while when using additional information from external covariates, this does not matter much. As expected, GAMs perform much better out-of-sample, and even in the simplest scenario, with no additional covariates and 20 features (or even when using the endogenous crime variable directly), linear models are clearly outperformed. This further encourages the usage of non-linear methods to model this relationship, even when already using non-linear techniques for creating the features and dimensionality-reduction with PCA.

As a robustness check, we repeat the forecasting with standardized covariates. The benefit of standardization here is rather limited. One the one hand, it facilitates the comparability between covariates, which would be helpful especially in Figures 6.7 and 6.8. On the other hand, the features all come out from the same model, where standardization would lead to a great loss of information and distort importance between features. In practice, standardization greatly reduces forecasting performance (see Table 6.4 in the Appendix), which is why we do not discuss it further, as the issues discussed above seem to affect the prediction performance greatly.

6.5 Conclusion

In this paper, we develop a model for house prices in NYC by incorporating information from satellite images and crime information, since employing crime directly within an interpretable model framework is infeasible due to of endogeneity concerns. We extend the framework of Deuschel et al. (2022) using neural networks to extract information from crimes. The extracted features are predictive for house prices and employing the semiparametric GAM-framework makes interpretation of the features possible outside of a simple linear model. The features are therefore different to those of Deuschel et al. (2022) and have a clear relationship with house prices, which we show to be indeed non-linear.

In more detail, using the GAM simplifies the base-framework as we do not need PCA-transformations to maintain high prediction performance, which was a crucial part in the model of Deuschel et al. (2022). We investigate the features in detail and show how they correspond to crime. We also consider the spatial structure of features in NYC and demonstrate its relation to prices, which again advocates employing the features in the house price model. We furthermore visualize the effect of the extracted features on crime and use Shapley Values in the final model. With that, we can visualize the final-model effects applying the new adapted version of Kernel-SHAP of Aas et al. (2021), crucially taking into account the dependence structure between covariates.

We also extend the framework of Deuschel et al. (2022) by employing a new architecture with a different CNN as a basis and using a more suitable loss function for interpretation. Furthermore, we find that the GAM serves a suitable alternative to machine-learning procedures such as boosting or random forests, while keeping an interpretable structure. For future research, it would be interesting to apply our methodology with GAMs in different related issues where there is either missing, insufficient data, or endogeneity.

6.6 Appendix

Table 6.3: Overview of the 9 Crime Classes

leading internal code number	Type of offenses	5 most occuring crimes (no. of occurences)
1	Offenses involving physical injury, sexual conduct, restraint, intimidation	Assault 3 & related offenses (521470) / Felony assault (190216) / Miscellaneous penal law (84016) / Sex crimes (39049) / Rape (13014)
2	Offenses involving damage to and intrusion upon property	Criminal mischief & related offenses (493471) / Burglary (171449) / Criminal trespass (59178) / Arson (12083) / Miscellaneous penal law (4003)
3	Offenses involving theft	Petit larceny (824298) / Robbery (182646) / Petit larceny of motor vehicle (696)
4	Offenses involving theft (grand)	Grand larceny (421768) / Grand larceny of motor vehicle (84956) / Possession of stolen property (26953) / Unauthorized use of a vehicle (14634) / Other offenses related to theft (11744)
5	Offenses against public health and morals	Dangerous drugs (316692) / Miscellaneous penal law (5593) / Gambling (2109) / Sex crimes (1263) / Prostitution & related offenses (824)
6	Offenses Against Public Order, Public Sensibilities and the Right to Privacy	Harassment 2 (601329) / Offenses against public order sensibility and privacy (259058) / Sex crimes (14277) / Miscellaneous penal law (5612) / Offenses related to children (1269)
7	Offenses against public administration and public safety / provisions relating to firearms, fireworks, pornography equipment and vehicles used in the transportation of gambling records	Dangerous weapons (119193) / Offenses against public administration (101718) / Theft-fraud (49720) / Forgery (47961) / Frauds (32036)
8	Other	Administrative code (10983) / Other state laws (non penal law) (4586) / Nys laws-unclassified felony (4359) / Alcoholic beverage control law (841) / Agriculture & markets law-unclassified (346)
9	Vehicle and Traffic Regulations	Intoxicated & impaired driving (69048) / Vehicle and traffic laws (60628)

Note: Code numbers as in the internal classification of the NYPD. Types of offenses are matched to the penal law extracted from the New York Senate penal laws (`https://www.nysenate.gov/legislation/laws/PEN/P3`).

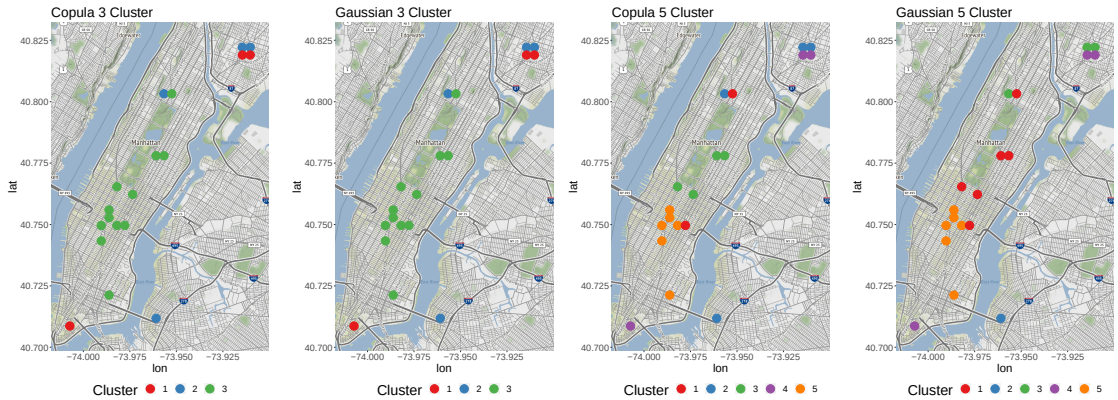


Figure 6.11: Maps of the 19 Selected Images Grouped by Shapley Value Similarty
Notes: Clusters are obtained using the similarity of the Shapley values for all covariates.

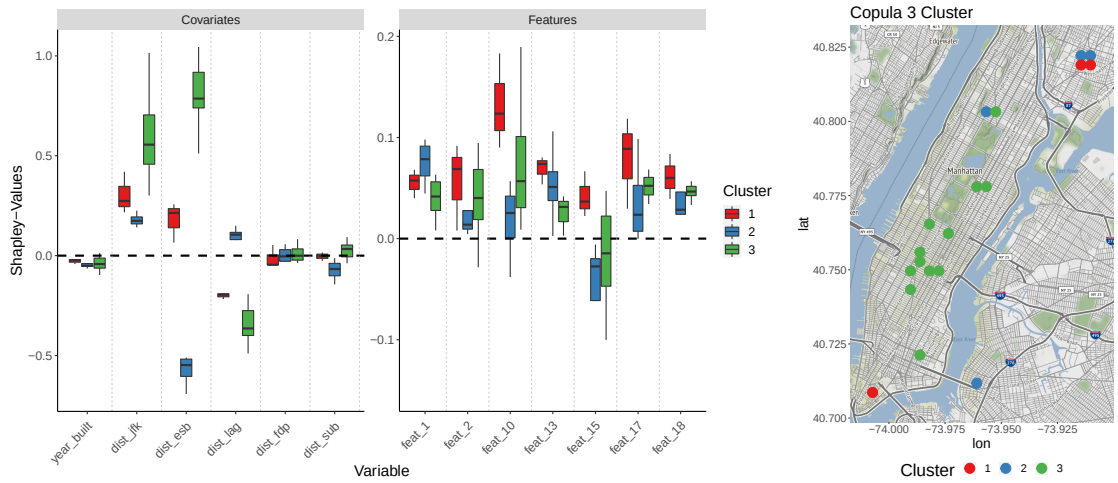


Figure 6.12: Boxplots of Shapley Values for Both the Covariates and Features Depending on the 3 Clusters and Location of Clusters on the NYC Map
Notes: Left: Boxplots of Shapley values for both the covariates and features depending on the 3 clusters. Note the different scaling of the y-axis for covariates and features, which facilitates interpretation. Right: Location of clusters on the NYC map.

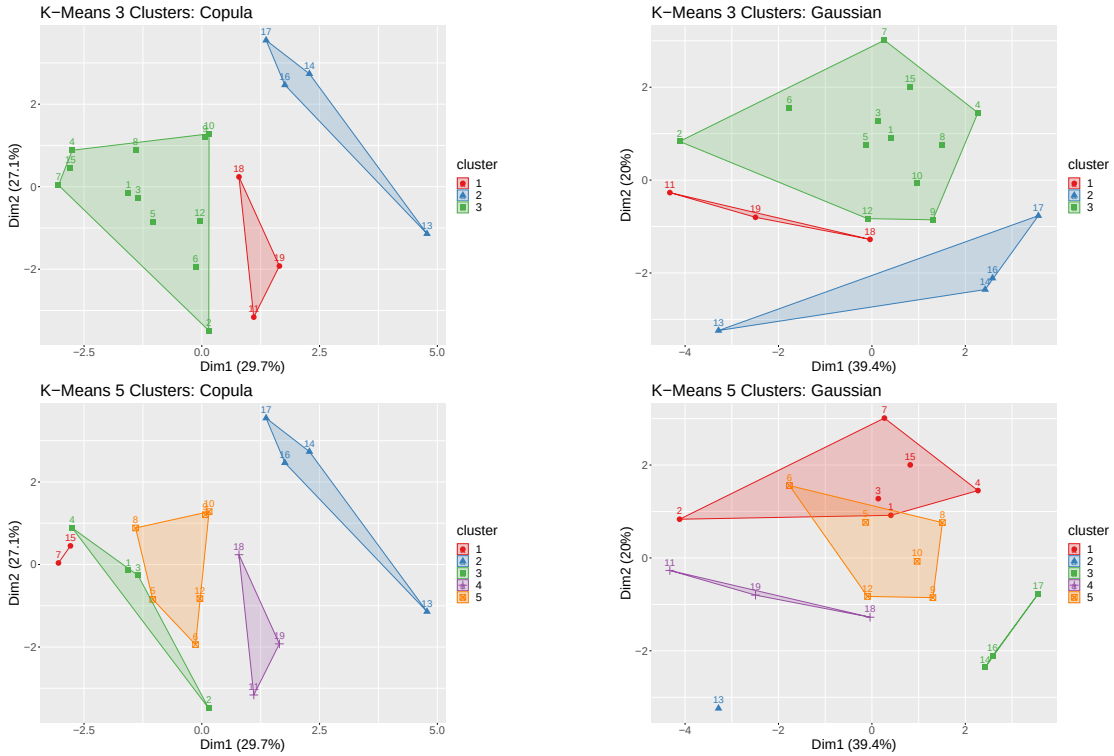


Figure 6.13: Position of Clusters for the 19 Selected Images Plotted by the First Two Principal Components

Notes: Similarity is determined by Shapley value for all covariates.

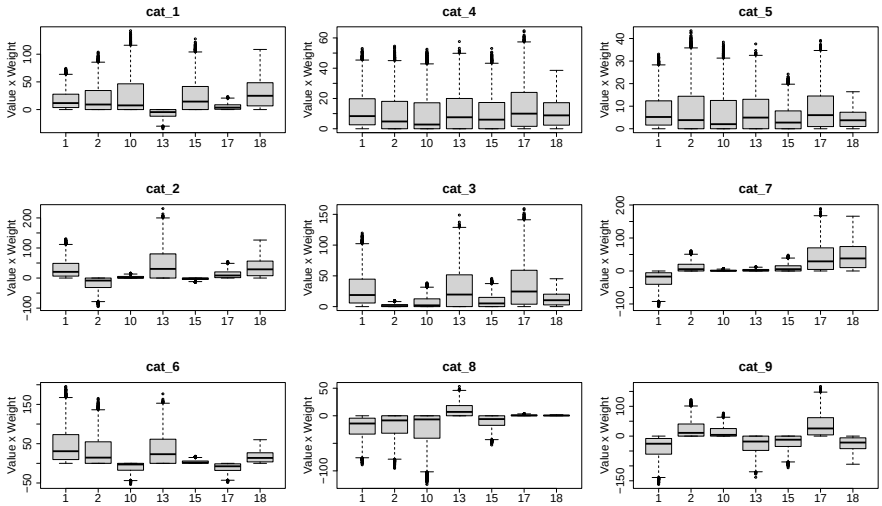


Figure 6.14: Boxplots of Features Time Weights for the InceptionResnetV2 With 20 Features (Only Non-zero)

Note: Each boxplot represents the weighted predictions of each feature for each crime category.

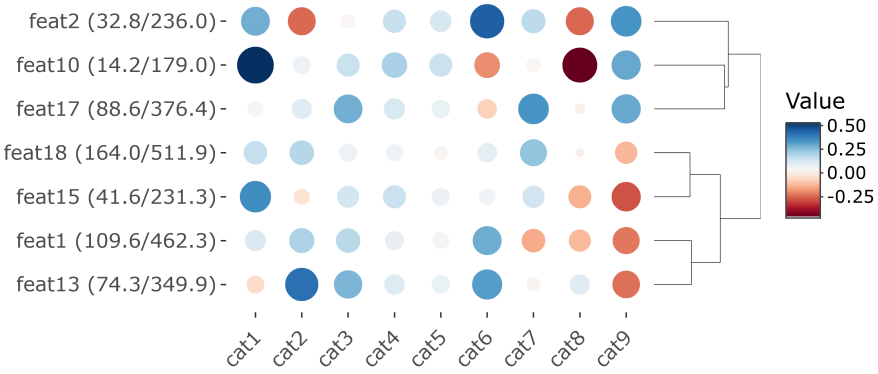


Figure 6.15: Weights of All Non-zero Features From the InceptionResnetV2-20 CNN for the Final Nine Crime Categories

Note: The columns include the 50% and 95% quantiles of feature values in parentheses.

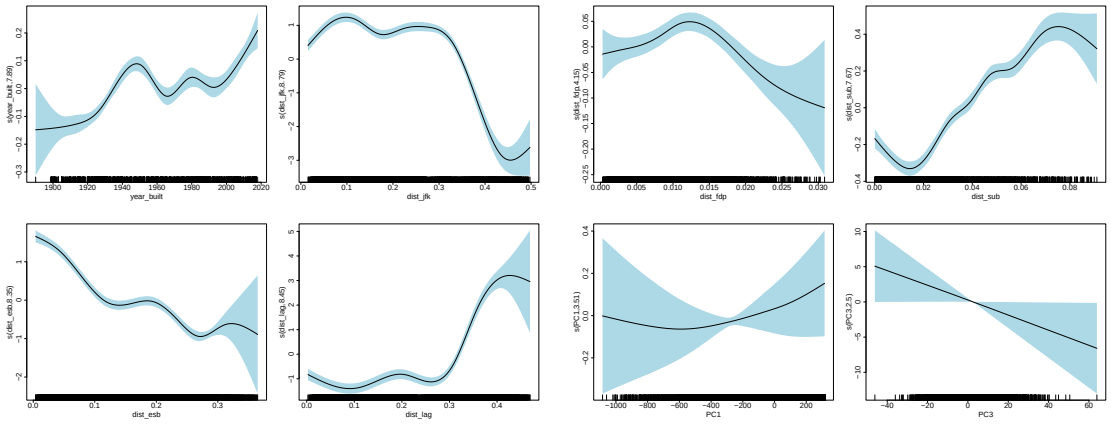


Figure 6.16: Marginal Effects for the GAM Within the 20-Feature InceptionResNetV2 and PCs

Notes: GAM is using cubic splines trained on covariates and first principal components (only signficant PCs are plotted). Black lines show GAM-estimates and shaded areas depict two standard errors above and below using Bayesian variance estimates. Black lines at x-axis are rug plots showing the distribution of data points. Note that the axes are scaled differently to visualize the non-linearities.

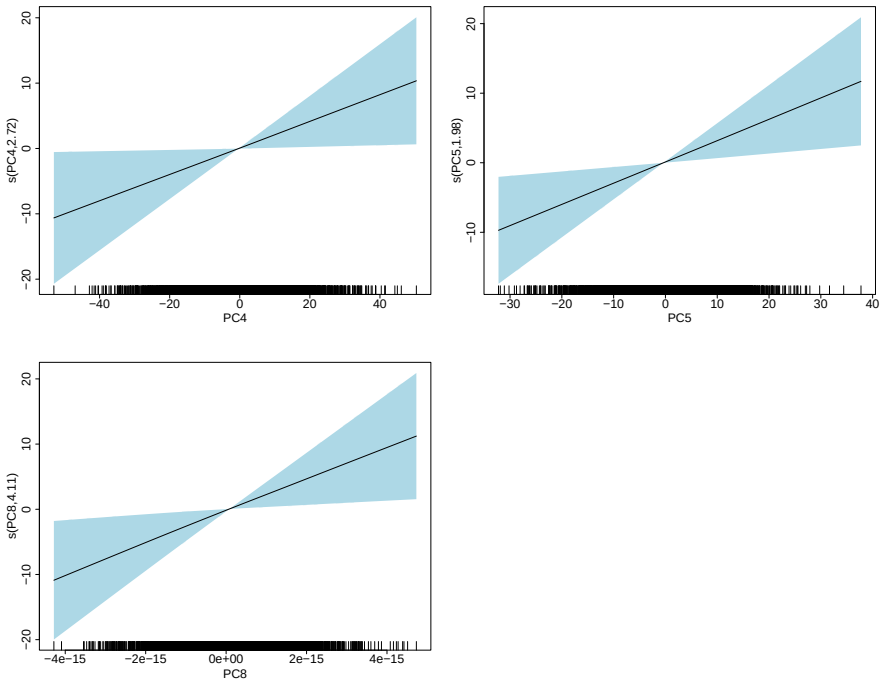


Figure 6.17: Marginal Effects of InceptionResnetV2 (Poisson Loss) 20-Feature GAM With Cubic Splines Trained With Covariates and Significant First Principal Components on Log-Price

Notes: Scaling of the y-axis is different to visualize the non-linearities. The rest of the details follow Figure 6.16.

Table 6.4: Predictive Power Results With OOS-R^2 and Standardized Covariates

	MSE-loss		*Poisson*-loss	
	With Z_i	Without Z_i	With Z_i	Without Z_i
Log Price				
100_layers_Features_GAM_10PC	0.291	0.276	0.303	0.216
100_layers_Features_GAM_raw	0.315	**0.305**	0.255	0.292
100_layers_Features_lm_10PC	0.461	**0.288**	0.433	0.205
100_layers_Features_lm_raw	**0.462**	0.259	**0.457**	0.290
20_layers_Features_GAM_10PC	0.297	0.246	0.216	0.065
20_layers_Features_GAM_raw	0.282	0.269	0.263	0.141
20_layers_Features_lm_10PC	0.441	0.199	0.420	0.106
20_layers_Features_lm_raw	0.449	0.202	0.420	0.105
750_layers_Features_GAM_10PC	0.263	0.155	0.315	**0.342**
750_layers_Features_GAM_raw	-	-	-	-
750_layers_Features_lm_10PC	**0.468**	0.256	**0.468**	**0.304**
750_layers_Features_lm_raw	0.047	-0.347	0.362	0.209
Crimes_GAM	0.341	0.084	0.341	0.084
Crimes_lm	0.411	0.107	0.411	0.107
Raw Price				
100_layers_Features_GAM_10PC	**0.707**	**0.397**	0.700	0.298
100_layers_Features_GAM_raw	0.684	0.358	0.567	0.266
100_layers_Features_lm_10PC	0.540	0.375	0.497	0.260
100_layers_Features_lm_raw	0.517	0.315	0.506	0.333
20_layers_Features_GAM_10PC	0.695	0.363	-0.448	-1.054
20_layers_Features_GAM_raw	0.701	**0.377**	0.695	0.145
20_layers_Features_lm_10PC	0.520	0.289	0.465	0.099
20_layers_Features_lm_raw	0.523	0.291	0.466	0.100
750_layers_Features_GAM_10PC	0.688	0.339	**0.711**	**0.456**
750_layers_Features_GAM_raw	-	-	-	-
750_layers_Features_lm_10PC	0.556	0.358	0.550	**0.409**
750_layers_Features_lm_raw	-0.452	-0.773	0.174	0.038
Crimes_GAM	**0.728**	0.375	**0.728**	0.375
Crimes_lm	0.522	0.267	0.522	0.267

Notes: Out-of-sample-R^2 rounded to three digits. The dependent variable is the price per square foot with (top) and without (bottom) logarithm. Features and additional covariates are standardized. Measures are giving features of the last fully-connected layer before the output of a CNN consisting of 100, 20, or 750 neurons. For each scenario, we compute generalized additive models (GAM) and linear models (lm), each taking either the raw features or the 10 first principal components. Additional features Z_i are used for model fitting certain scenarios. *Crimes_GAM* and *Crimes_lm* use the true crimes instead of features/principal components. It was not computationally feasible to compute a GAM consisting of 750 features. The best two models in each scenario are marked in bold.

Bibliography

Aas, K., M. Jullum, and A. Løland (2021): "Explaining individual predictions when features are dependent: More accurate approximations to Shapley values," *Artificial Intelligence*, 298, 103502.

Abadie, A., S. Athey, G. W. Imbens, and J. M. Wooldridge (2020): "Sampling-based vs. Design-based Uncertainty in Regression Analysis," *Econometrica*, 88, 265–296.

Achanta, R., A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk (2012): "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE transactions on pattern analysis and machine intelligence*, 34, 2274–2281.

Altmann, A., L. Tološi, O. Sander, and T. Lengauer (2010): "Permutation importance: A corrected feature importance measure," *Bioinformatics*, 26, 1340–1347.

Angrist, J. and Brigham Frandsen (2019): "Machine Labor," *NBER Working Paper Series*, 1689–1699.

Anselin, L. and N. Lozano-Gracia (2007): "Errors in variables and spatial effects in hedonic house price models of ambient air quality," *Empirical Economics*, 34, 5–34.

Ardia, D., K. Boudt, and L. Catania (2019): "Generalized autoregressive score models in R: The GAS package," *Journal of Statistical Software*, 88.

Ashenfelter, O. and D. Card (1985): "Using the Longitudinal Structure of Earnings to Estimate the Effect of Training Programs," *The Review of Economics and Statistics*, 67, 648.

Athey, S. (2017): "Beyond prediction: Using big data for policy problems," *Science*, 355, 483–485.

Athey, S. and G. Imbens (2016): "Recursive partitioning for heterogeneous causal effects," *Proceedings of the National Academy of Sciences of the United States of America*, 113, 7353–7360.

Athey, S. and G. W. Imbens (2019): "Machine Learning Methods That Economists Should Know about," *Annual Review of Economics*, 11, 685–725.

——— (2022): "Design-based analysis in Difference-In-Differences settings with staggered adoption," *Journal of Econometrics*, 226, 62–79.

Athey, S., J. Tibshirani, and S. Wager (2019): "Generalized random forests," *Annals of Statistics*, 47, 1179–1203.

Baier, T. and M. Helbig (2011): "War all die Aufregung umsonst? Über die Auswirkung der Einführung von Studiengebühren auf die Studienbereitschaft in Deutschland." *Discussion Paper*, 2011-001.

Barber, R. F. and E. J. Candès (2015): "Controlling the false discovery rate via knockoffs," *Annals of Statistics*, 43, 2055–2085.

——— (2019): "A knockoff filter for high-dimensional selective inference," *Annals of Statistics*, 47, 2504–2537.

Bareinboim, E. and J. Pearl (2016): "Causal inference and the data-fusion problem," *Proceedings of the National Academy of Sciences*, 113, 7345–7352.

Baur, D. G., K. H. Hong, and A. D. Lee (2018): "Bitcoin: Medium of exchange or speculative assets?" *Journal of International Financial Markets, Institutions and Money*, 54, 177–189.

Belloni, A., V. Chernozhukov, and C. Hansen (2014a): "High-dimensional methods and inference on structural and treatment effects," *Journal of Economic Perspectives*, 28, 29–50.

——— (2014b): "Inference on treatment effects after selection among high-dimensional controls," *Review of Economic Studies*, 81, 608–650.

Belloni, A., V. Chernozhukov, C. Hansen, and D. Kozbur (2016): "Inference in High-Dimensional Panel Models With an Application to Gun Control," *Journal of Business and Economic Statistics*, 34, 590–605.

Bellotti, A., D. Brigo, P. Gambetti, and F. Vrins (2021): "Forecasting recovery rates on non-performing loans with machine learning," *International Journal of Forecasting*, 37, 428–444.

Bellotti, T. and J. Crook (2012): "Loss given default models incorporating macroeconomic variables for credit cards," *International Journal of Forecasting*, 28, 171–182.

Bernardi, M. and L. Catania (2018): "The Model Confidence Set package for R," *International Journal of Computational Economics and Econometrics*, 8, 144–158.

Blei, D. M. and P. Smyth (2017): "Science and data science," *Proceedings of the National Academy of Sciences*, 114, 8689–8692.

Bollerslev, T. (1986): "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, 31, 307–327.

Breiman, L. (1996): "Some properties of splitting criteria," *Machine Learning*, 24, 41–47.

——— (2001): "Random Forests," *Machine Learning*, 45, 5–32.

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984): *Classification and regression trees*, CRC press.

Bruckmeier, K. and B. U. Wigger (2014): "The effects of tuition fees on transition from high school to university in Germany," *Economics of Education Review*, 41, 14–23.

Brunauer, W. A., S. Lang, P. Wechselberger, and S. Bienert (2010): "Additive Hedonic Regression Models with Spatial Scaling Factors: An Application for Rents in Vienna," *Journal of Real Estate Finance and Economics*, 41, 390–411.

Buhaug, H. and H. Urdal (2013): "An urbanization bomb? Population growth and social disorder in cities," *Global Environmental Change*, 23, 1–10.

Bui, T. D., J. Yan, and R. E. Turner (2017): "A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation," *Journal of Machine Learning Research*, 18, 1–72.

Candès, E., Y. Fan, L. Janson, and J. Lv (2018): "Panning for gold: 'model-X' knockoffs for high dimensional controlled variable selection," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 80, 551–577.

Card, D. and A. B. Krueger (1994): "Minimum wages and employment: a case study of the fast-food industry in New Jersey and Pennsylvania," *American Economic Review*, 84, 772–793.

Chang, N. C. (2020): "Double/debiased machine learning for difference-in-differences models," *Econometrics Journal*, 23, 177–191.

Cheah, E. T. and J. Fry (2015): "Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin," *Economics Letters*, 130, 32–36.

Chen, L., M. Pelger, and J. Zhu (2019): "Deep Learning in Asset Pricing," *SSRN Electronic Journal*.

Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins (2018): "Double/debiased machine learning for treatment and structural parameters," *Econometrics Journal*, 21, C1–C68.

Chinco, A., A. D. Clark-Joseph, and M. Ye (2019): "Sparse Signals in the Cross-Section of Returns," *The Journal of Finance*, 74, 449–492.

Christoffersen, P. F. (1998): "Evaluating Interval Forecasts," *International Economic Review*, 39, 841.

Chu, J., S. Chan, S. Nadarajah, and J. Osterrieder (2017): "GARCH Modelling of Cryptocurrencies," *Journal of Risk and Financial Management*, 10, 17.

Dai, R. and R. F. Barber (2016): "The knockoff filter for FDR control in group-sparse and multitask regression," *33rd International Conference on Machine Learning, ICML 2016*, 4, 2752–2760.

Das, S. R. and P. Hanouna (2009): "Implied recovery," *Journal of Economic Dynamics and Control*, 33, 1837–1857.

de La Paz, P. T., J. Berry, D. McIlhatton, D. Chapman, and K. Bergonzoli (2022): "The impact of crimes on house prices in LA County," *Journal of European Real Estate Research*, 15, 88–111.

Denny, K. (2014): "The effect of abolishing university tuition costs: Evidence from Ireland," *Labour Economics*, 26, 26–33.

Deuschel, J., K. Görgen, and M. Schienle (2022): "Predicting Property Prices Using Augmented Crime Data: Extracting Information from Satellite Images with Convolutional Neural Networks," *KIT Working Paper*.

Dhar, V. (2013): "Data science and prediction," *Communications of the ACM*, 56, 64–73.

Dubin, R. A. (1998): "Predicting House Prices Using Multiple Listings Data," *Journal of Real Estate Finance and Economics*, 17, 35–59.

Dwenger, N., J. Storck, and K. Wrohlich (2012): "Do tuition fees affect the mobility of university applicants? Evidence from a natural experiment," *Economics of Education Review*, 31, 155–167.

Dynarski, S. M. (2003): "Does aid matter? Measuring the effect of student aid on college attendance and completion," *American Economic Review*, 93, 279–288.

Einav, L. and J. Levin (2014): "Economics in the age of big data," *Science*, 346, 1243089.

Elendner, H., S. Trimborn, B. Ong, and T. M. Lee (2017): "The Cross-Section of Crypto-Currencies as Financial Assets: Investing in Crypto-Currencies Beyond Bitcoin,"

in *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 1: Cryptocurrency, FinTech, InsurTech, and Regulation*, Elsevier, 145–173.

Engle, R. F. and S. Manganelli (2004): "CAViaR: Conditional autoregressive value at risk by regression quantiles," *Journal of Business and Economic Statistics*, 22, 367–381.

Federal Statistical Office (2014a): "Monetäre hochschulstatistische Kennzahlen," Tech. rep., Federal Statistical Office.

——— (2014b): "Nichtmonetäre hochschulstatistische Kennzahlen," Tech. rep., Federal Statistical Office.

——— (2014c): "Personal an Hochschulen," Tech. rep., Federal Statistical Office.

——— (2014d): "Studierende an Hochschulen," Tech. rep., Federal Statistical Office.

Feng, G., S. Giglio, and D. Xiu (2020): "Taming the Factor Zoo: A Test of New Factors," *The Journal of Finance*, 75, 1327–1370.

Freyberger, J., A. Neuhierl, and M. Weber (2020): "Dissecting Characteristics Nonparametrically," *Review of Financial Studies*, 33, 2326–2377.

Friedman, J., T. Hastie, H. Höfling, and R. Tibshirani (2007): "Pathwise coordinate optimization," *The Annals of Applied Statistics*, 1, 302–332.

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

Ghysels, E. and G. Nguyen (2019): "Price Discovery of a Speculative Asset: Evidence from a Bitcoin Exchange," *Journal of Risk and Financial Management*, 12, 164.

Giacomini, R. and I. Komunjer (2005): "Evaluation and combination of conditional quantile forecasts," *Journal of Business and Economic Statistics*, 23, 416–431.

Giacomini, R. and H. White (2006): "Tests of conditional predictive ability," *Econometrica*, 74, 1545–1578.

Gibbons, S. (2004): "The costs of urban property crime," *Economic Journal*, 114, F441–F463.

Gkillas, K. and P. Katsiampa (2018): "An application of extreme value theory to cryptocurrencies," *Economics Letters*, 164, 109–111.

Glaser, F., K. Zimmermann, M. Haferkorn, M. C. Weber, and M. Siering (2014): "Bitcoin - Asset or currency? Revealing users' hidden intentions," *ECIS 2014 Proceedings - 22nd European Conference on Information Systems*, 1–14.

Glosten, L. R., R. Jagannathan, and D. E. Runkle (1993): "On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks," *The Journal of Finance*, 48, 1779–1801.

Gu, S., B. Kelly, and D. Xiu (2020): "Empirical Asset Pricing via Machine Learning," *The Review of Financial Studies*, 33, 2223–2273.

Hafner, C. M. (2020): "Testing for Bubbles in Cryptocurrencies with Time-Varying Volatility," *Journal of Financial Econometrics*, 18, 233–249.

Hájek, P. (2011): "Municipal credit rating modelling by neural networks," *Decision Support Systems*, 51, 108–118.

Hansen, P. R., A. Lunde, and J. M. Nason (2011): "The model confidence set," *Econometrica*, 79, 453–497.

Hartford, J., G. Lewis, K. Leyton-Brown, and M. Taddy (2017): "Deep IV: A Flexible Approach for Counterfactual Prediction," in *International Conference on Machine Learning*, PMLR, 1414–1423.

Hastie, T., R. Tibshirani, and J. Friedman (2009): *The Elements of Statistical Learning*, Springer, second ed.

Haughwout, A., J. Orr, and D. Bedoll (2008): "The Price of Land in the New York Metropolitan Area," *Current Issues in Economics and Finance*, 14, 1–7.

He, K., X. Zhang, S. Ren, and J. Sun (2016): "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hencic, A. and C. Gouriéroux (2015): "Noncausal autoregressive model in application to bitcoin/USD exchange rates," in *Studies in Computational Intelligence*, Springer, vol. 583, 17–40.

Hübner, M. (2012): "Do tuition fees affect enrollment behavior? Evidence from a 'natural experiment' in Germany," *Economics of Education Review*, 31, 949–960.

Huijsman, R., T. Kloek, D. A. Kodde, and J. M. M. Ritzen (1986): "An Empirical Analysis of College Enrollment in the Netherlands," *De Economist*, 134, 181–190.

Ihlanfeldt, K. and T. Mayock (2010): "Panel data estimates of the effects of different types of crime on housing prices," *Regional Science and Urban Economics*, 40, 161–172.

Imbens, G. (2004): "Nonparametric estimation of average treatment effects under exogeneity: A review," *Review of Economics and Statistics*, 86, 4–29.

Jankowitsch, R., F. Nagler, and M. G. Subrahmanyam (2014): "The determinants of recovery rates in the US corporate bond market," *Journal of Financial Economics*, 114, 155–177.

Jansen, J., S. R. Das, and F. J. Fabozzi (2018): "Local volatility and the recovery rate of credit default swaps," *Journal of Economic Dynamics and Control*, 92, 1–29.

Jean, N., M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon (2016): "Combining satellite imagery and machine learning to predict poverty," *Science*, 353, 790–794.

Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009): "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 248–255.

Jim, C. Y. and W. Y. Chen (2009): "Value of scenic views: Hedonic assessment of private housing in Hong Kong," *Landscape and Urban Planning*, 91, 226–234.

Kane, T. J. (1994): "College Entry by Blacks since 1970 : The Role of College Costs, Family Background, and the Returns to Education," *Journal of Political Economy*, 102, 878–911.

Kaposty, F., J. Kriebel, and M. Löderbusch (2020): "Predicting loss given default in leasing: A closer look at models and variable selection," *International Journal of Forecasting*, 36, 248–266.

Keijsers, B., B. Diris, and E. Kole (2018): "Cyclicality in losses on bank loans," *Journal of Applied Econometrics*, 33, 533–552.

Kellner, R., M. Nagl, and D. Rösch (2022): "Opening the black box – Quantile neural networks for loss given default prediction," *Journal of Banking and Finance*, 134, 106334.

Kelly, B. T., S. Pruitt, and Y. Su (2019): "Characteristics are covariances: A unified model of risk and return," *Journal of Financial Economics*, 134, 501–524.

Koenker, R. and G. Bassett (1978): "Regression Quantiles," *Econometrica*, 46, 33.

Koenker, R. and K. F. Hallock (2001): "Quantile regression," *Journal of economic perspectives*, 15, 143–156.

Kohler, M. (2010): "Exchange Rates During Financial Crises," *BIS Quarterly Review*, March, 39–50.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012): "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, vol. 25, 1097–1105.

Kupiec, P. H. (1995): "Techniques for Verifying the Accuracy of Risk Measurement Models," *The Journal of Derivatives*, 3, 73–84.

Kwiatkowski, D., P. C. Phillips, P. Schmidt, and Y. Shin (1992): "Testing the null hypothesis of stationarity against the alternative of a unit root. How sure are we that economic time series have a unit root?" *Journal of Econometrics*, 54, 159–178.

Lee, J. (2018): "A Neural Network Method for Nonlinear Time Series Analysis," *Journal of Time Series Econometrics*, 11, 20160011.

Leow, M. and C. Mues (2012): "Predicting loss given default (LGD) for residential mortgage loans: A two-stage model and empirical evidence for UK bank data," *International Journal of Forecasting*, 28, 183–195.

Li, Z. and S. N. Wood (2020): "Faster model matrix crossproducts for large generalized linear models with discretized covariates," *Statistics and Computing*, 30, 19–25.

Liu, W., A. Semeyutin, C. K. M. Lau, and G. Gozgor (2020): "Forecasting Value-at-Risk of Cryptocurrencies with RiskMetrics type models," *Research in International Business and Finance*, 54, 101259.

Liu, Y. and A. Tsyvinski (2020): "Risks and Returns of Cryptocurrency," *The Review of Financial Studies*, 34, 2689–2727.

Lundberg, S. M. and S.-I. Lee (2017): "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., vol. 30, 4765–4774.

Maciel, L. (2020): "Cryptocurrencies value-at-risk and expected shortfall: Do regime-switching volatility models improve forecasting?" *International Journal of Finance & Economics*.

MacKinnon, J. G. and H. White (1985): "Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties," *Journal of Econometrics*, 29, 305–325.

Marra, G. and S. N. Wood (2012): "Coverage Properties of Confidence Intervals for Generalized Additive Model Components," *Scandinavian Journal of Statistics*, 39, 53–74.

McCauley, R. N. and P. McGuire (2009): "Dollar appreciation in 2008: safe haven, carry trades, dollar shortage and overhedging," *BIS Quarterly Review*, December, 85–93.

McPherson, M. S. and M. O. Schapiro (1991): "Does Student Aid Affect College Enrollment? New Evidence on a Persistent Controversy," *Economics of Education Review*, 81, 309–318.

Meinshausen, N. (2006): "Quantile Regression Forests," *Journal of Machine Learning Research*, 7, 983–999.

Meinshausen, N. and P. Bühlmann (2010): "Stability selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72, 417–473.

Mitze, T., C. Burgard, and B. Alecke (2015): "The tuition fee 'shock': Analysing the response of first-year students to a spatially discontinuous policy change in Germany," *Papers in Regional Science*, 94, 385–419.

Mullainathan, S. and J. Spiess (2017): "Machine learning: An applied econometric approach," *Journal of Economic Perspectives*, 31, 87–106.

Najjar, A., S. Kaneko, and Y. Miyanaga (2017): "Crime Mapping from Satellite Imagery via Deep Learning," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, CA, 752–760.

Nazemi, A., F. Baumann, and F. J. Fabozzi (2022): "Intertemporal defaulted bond recoveries prediction via machine learning," *European Journal of Operational Research*, 297, 1162–1177.

Nazemi, A. and F. J. Fabozzi (2018): "Macroeconomic variable selection for creditor recovery rates," *Journal of Banking and Finance*, 89, 14–25.

Nazemi, A., K. Heidenreich, and F. J. Fabozzi (2018): "Improving corporate bond recovery rate prediction using multi-factor support vector regressions," *European Journal of Operational Research*, 271, 664–675.

Neill, C. (2009): "Tuition fees and the demand for university places," *Economics of Education Review*, 28, 561–570.

Noorbakhsh, A. and D. Culp (2002): "The demand for higher education: Pennsylvania's nonresident tuition experience," *Economics of Education Review*, 21, 277–286.

Pafka, S. and I. Kondor (2001): "Evaluating the RiskMetrics methodology in measuring volatility and Value-at-Risk in financial markets," *Physica A: Statistical Mechanics and its Applications*, 299, 305–310.

Perez, L. and J. Wang (2017): "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," *arXiv preprint arXiv:1712.04621*.

Petukhina, A., S. Trimborn, W. K. Härdle, and H. Elendner (2021): "Investing with cryptocurrencies – evaluating their potential for portfolio allocation strategies," *Quantitative Finance*, 0, 1–29.

Platanakis, E. and A. Urquhart (2019): "Portfolio management with cryptocurrencies: The role of estimation risk," *Economics Letters*, 177, 76–80.

Qi, M. and X. Zhao (2011): "Comparison of modeling methods for Loss Given Default," *Journal of Banking and Finance*, 35, 2842–2855.

Ren, Z., Y. Wei, and E. Candès (2021): "Derandomizing Knockoffs," *Journal of the American Statistical Association*, 1–29.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016): ""Why should i trust you?" Explaining the predictions of any classifier," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Augu, 1135–1144.

Rischard, M., Z. Branson, L. Miratrix, and L. Bornn (2020): "Do School Districts Affect NYC House Prices? Identifying Border Differences Using a Bayesian Nonparametric Approach to Geographic Regression Discontinuity Designs," *Journal of the American Statistical Association*, 116, 619–631.

Romano, Y., M. Sesia, and E. Candès (2020): "Deep Knockoffs," *Journal of the American Statistical Association*, 115, 1861–1872.

Rosenbaum, P. R. and D. B. Rubin (1983): "The central role of the propensity score in observational studies for causal effects," *Biometrika*, 70, 41–55.

Rubin, D. B. (1974): "Estimating causal effects of treatment in randomized and nonrandomized studies," *Journal of Educational Psychology*, 66, 688–701.

——— (1977): "Assignment to Treatment Group on the Basis of a Covariate," *Journal of Educational Statistics*, 2, 1–26.

Schwartz, A. E., I. Voicu, and K. M. Horn (2014): "Do choice schools break the link between public schools and property values? Evidence from house prices in New York City," *Regional Science and Urban Economics*, 49, 1–10.

Selmi, R., A. Tiwari, and S. Hammoudeh (2018): "Efficiency or speculation? A dynamic analysis of the Bitcoin market," *Economics Bulletin*, 38, 2037–2046.

Shapley, L. S. (1953): "A value for n-person games," *Contributions to the Theory of Games*, 2, 307–317.

Shrikumar, A., P. Greenside, and A. Kundaje (2017): "Learning important features through propagating activation differences," *34th International Conference on Machine Learning, ICML 2017*, 7, 4844–4866.

Sieg, H. and C. Yoon (2020): "Waiting for Affordable Housing in New York City," *Quantitative Economics*, 11, 277–313.

Simonyan, K. and A. Zisserman (2015): "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015*, 1–14.

Suykens, J. A. and J. Vandewalle (1999): "Least squares support vector machine classifiers," *Neural Processing Letters*, 9, 293–300.

Szegedy, C., S. Ioffe, V. Vanhoucke, and A. A. Alemi (2017): "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*.

Takeda, A. and M. Sugiyama (2008): "N-Support Vector Machine As Conditional Value-At-Risk Minimization," *Proceedings of the 25th International Conference on Machine Learning*, 1056–1063.

Tibshirani, R. (1996): "Regression Selection and Shrinkage via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, 58, 267–288.

Tibshirani, R., G. Walther, and T. Hastie (2001): "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63, 411–423.

Trimborn, S., M. Li, and W. K. Härdle (2020): "Investing with Cryptocurrencies - A Liquidity Constrained Investment Approach," *Journal of Financial Econometrics*, 18, 280–306.

Trucios, C., A. K. Tiwari, and F. Alqahtani (2020): "Value-at-risk and expected shortfall in cryptocurrencies' portfolio: A vine copula–based approach," *Applied Economics*, 52, 2580–2593.

Vigliotti, M. G. and H. Jones (2020): "The Rise and Rise of Cryptocurrencies," in *The Executive Guide to Blockchain*, Springer, 71–91.

Vomfell, L., W. K. Härdle, and S. Lessmann (2018): "Improving Crime Count Forecasts Using Twitter and Taxi Data," *Decision Support Systems*, 113, 73–85.

Wagner, G., J. Frick, and J. Schupp (2007): "The German Socio-Economic Panel Study (SOEP): Scope, Evolution and Enhancements," *Schmollers Jahrbuch - Journal of Applied Social Science Studies*, 127, 139–170.

Wood, S. N. (2008): "Fast stable direct fitting and smoothness selection for generalized additive models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70, 495–518.

——— (2013): "On p-values for smooth components of an extended generalized additive model," *Biometrika*, 100, 221–228.

Wood, S. N., Y. Goude, and S. Shaw (2015): "Generalized additive models for large data sets," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 64, 139–155.

Wood, S. N., Z. Li, G. Shaddick, and N. H. Augustin (2017): "Generalized Additive Models for Gigadata: Modeling the U.K. Black Smoke Network Daily Data," *Journal of the American Statistical Association*, 112, 1199–1210.

Wooldridge, J. M. (2002): *Econometric analysis of cross section and panel data*, Cambridge and London: MIT Press.

Yao, X., J. Crook, and G. Andreeva (2015): "Support vector regression for loss given default modelling," *European Journal of Operational Research*, 240, 528–538.

Yoo, S., J. Im, and J. E. Wagner (2012): "Variable selection for hedonic model using machine learning approaches: A case study in Onondaga County, NY," *Landscape and Urban Planning*, 107, 293–306.