Predicting German Election Outcomes With Machine Leaning

Dang Ngoc Huy
h.dang@mpp.hertie-school.org

Maximillian Rekuts
m.rekuts@mia.hertie-school.org

Bailey Sutton
b.sutton@exchange.hertie-school.org

Abstract

The ability to predict election outcomes has long been an aim of social science research due to the significant political, social, and economic implications of the task. However, a voter's behavior is not always rational and there are multiple confounding factors that can effect the vote share of a candidate or political party. Stoetzer, Neunhoeffer, Gschwend, Munzert, & Sternberg (2019) attempted to use classical statistical models to predict the 2017 German federal election and were able to do so within the margin of error for 6 of the 7 parties. The purpose of this paper is to use the same data and scope as Stoetzer et al., but with the aim of simplifying their model through the utilization of machine learning and deep learning techniques. The main challenge of the data is the inconsistent frequencies in the time-series resulting from different polling process by individual institutions. Therefore, this paper evaluates six different algorithms to identify the best model for predicting the changes in public support for a particular party based on previous polls. The Recurrent Neural Network (RNN) estimates outperforms the other five machine learning models in terms of accuracy for AfD, FDP, Die Linke, the Green Party, and "Other Parties" when evaluating on the same metric as by Stoetzer et al. However, the accuracy drops for prediction of voteshares for CDU and SPD. Nonetheless, the promising results of these models show the potential for machine learning and deep learning models to be utilized for vote share forecast in the context of multiparty election system.

1. Introduction [1]

The most state of the art model in election outcome prediction work with advanced statistical models to forecast outcomes. Thus, the current literature lacks usage of machine learning models and algorithms to predict election outcomes. The innate difficulty in using election data lies in the fact that the data is inherently a sequence of discrete-time data points taken over time. Thus, pre-processing the data so it can be used by machine learning algorithms is a steep task. However, the ease of interpretation of these models compared to the intricate statistical models presently used, makes machine learning an attractive avenue for analysis.

This paper will first provide an overview of the current literature in the area of election predictions. Then, the proposed methods for this paper will be outlined along with several concerns with using these models for time-series predictions. Finally, there is an overview of the experiments themselves followed by an analysis of the results and concluding remarks.

2. Related Work

The purpose of this project is to adapt the work conducted by **Stoetzer, Neunhoeffer, Gschwend, Munzert, & Sternberg in Forecasting Elections in Multiparty Systems: A Bayesian Approach Combining Polls and Fundamentals, *Political Analysis* (2019)** to build an election prediction model using machine learning methods [7]. The data from their project will be retrained with several machine learning models. A comparison will be made between their original model and the best machine learning method in order to better understand the future of election forecasting.

Stoetzer et al. (2019) constructed a forecasting model that accurately predicts the captured vote share in a multiparty election system [7]. Their project is the result of a gap of multiparty forecasting models in current literature as the majority of work is centered on the United States, which is a two-party system [7]. In this paper, the authors mainly focus on Germany's 2017 federal election, but also prove their model's broader applicability by testing it on data from the 2017 general election in New Zealand [7]. Their model has two parts. The first is a fundamentals model that uses historical information about the political parties, including pos-

[1] Project GitHub: https://github.com/huydang90/Predicting-German-Election

sible coalitions, to make predictions without campaign or polling information [7]. The second is a Bayesian measurement model that combines polling information from several polling companies to judge party support over a defined time period [7]. This combined model allows for more dynamic predictions as new information is added [7]. In order to properly weight additional polling information, the authors used a random walk backwards method that models the level of support relative to the time the poll was released [7]. This gives polling data released closer to the election data more weight in the model than earlier polling information [7]. When the authors apply this model to the 2017 German federal election, six of their seven predicted intervals successfully captured the actual vote share won by the parties [7]. They further show that using the dynamic Bayesian model led to predictive improvements over time [7]. This paper is of interest for further analysis because of its innovative approach to predicting multiparty election outcomes. However, due to the limited work in this specific area, there is also the opportunity to improve their methodology and continue testing new forecasting approaches.

When reviewing current literature to assess additional election forecasting techniques, there is limited usage of machine learning models to predict elections with data other than that from social media. Nearly all of the literature in this field uses Twitter data to construct sentiment analysis models as the primary machine learning application. While additional research is limited, there are some examples of researchers using similar data and machine learning models for predicting elections, however none specifically look at multiparty elections. Therefore, this project will seek to fill this gap by using Stoetzer et al. (2019)'s historical polling information and political structures data to craft a novel usage of machine learning techniques for election prediction modelling.

Lewis-Beck & Dassonneville (2015) propose synthetic forecasting models for European elections, which combine structural models that rely on political economic theory and aggregate models that use voter preferences as their primary predictive data [6]. While structural models are the most common approach for analyzing European elections, the authors found that synthetic models are more accurate and more adept in their ability to be applied to different elections [6]. In this paper, the authors look at all three types of models to illustrate their conclusion that a synthetic model is the most accurate by using the same outcome variable, incumbent vote share [6]. They use data from Germany between 1980 and 2013, Ireland between 1977 and 2011, and the United Kingdom between 1959 and 2010 [6]. The structural model includes variables for government approval as the macro-political indicator and GDP growth rates as the macro-economic indicator [6]. The aggregate model uses polling information beginning from six months before the election and then subsequent months until one month before the election [6]. They then present two synthetic models - one with both indicators from the structural model and the polling information from the aggregate model and another with the combined predictions from the structural model and the polling information from the aggregate model [6]. The latter model is slightly more accurate than the former [6]. The conclusions found by the authors demonstrate the combined importance of structural factors and polling information in predictive modelling, which supports the data used by Stoetzer et al. (2019) [6]. While Lewis-Beck & Dassonneville (2015) used a more simplified OLS model to predict outcomes, their logic and conclusions are useful when assessing key variables to include in forecasting models.

Kennedy, Wojcik, & Lazer (2017) are some of the few researchers who have used machine learning models to predict elections using data from a source other than social media [4]. In an attempt to improve the quality of data and the number of cases when developing an election prediction model, the authors used cross-national data for 86 countries from 1945 to 2012 [4]. Their training data set contained observations between 1945 and 2006 and the test set was all elections between 2007 and 2012 [4]. They used Bayesian additive regression trees as their primary algorithm and tuned the data using cross-validation [4]. Using these techniques, the authors achieved a 78.9% accuracy rate for the training data and an 81.9% accuracy rate for the test data [4]. Following these results, the authors included more robust polling data, including public opinion polls and information on potential bias based on the political leanings of the polling institution [4]. The accuracy of their model improved by 10 percentage points after adding this additional information [4]. While the scope of their project is much greater than the one proposed here, the basis of their independent variables mirrors this planned approach. Additionally, their insight on the number of data points required for accurate predictions is an important consideration and further supports the use of historical polling data for this project. Finally, their modelling technique showed promising results and is worth testing on the data for this project.

Finally, Zolghadr, Akhavan Niaki & Niaki (2018) have also applied machine learning algorithms on a smaller scale by attempting to predict the output of US Presidential elections [8]. They included several independent variables in their model that cover several relevant social, political, and economic factors [8]. They selected the most significant variables for the model using a step-wise regression function in the pre-processing stage [8]. The authors found that taking these steps drastically improved the final model's predictive ability [8]. In order to assess which machine learning model would be the most successful for prediction, they tested three different ones: support vector regression (SVR), artificial neural networks (ANN), and a basic multivariate linear regression to be used as the baseline for comparison [8]. The SRV model was the most accurate in predicting the 2004, 2008, and 2012 US presidential elections [8]. Ultimately, the researchers recommend a methodology that combines both of the machine learning models for the most accurate predictions [8]. Similar to the work by Kennedy, Wojcik, & Lazer (2017), this paper provides a valuable basis for developing a prediction model for this project.

3. Proposed Method

The purpose of this project is to take the data from Stoetzer, Neunhoeffer, Gschwend, Munzert, & Sternberg (2019) [7] and determine whether or not there is a comparable or better machine learning model in terms of accurately predicting the results of the 2017 German federal election. The primary assessment metric for accuracy is RMSE, which mirrors what Stoetzer et al. used for their findings. In terms of predictive ability, machine learning models have the potential to perform as well as classical statistical models, such as the one used by Stoetzer et al., but with significantly greater interpretability.

There are two main concerns with using time-series polling data in relation to this project. First, there needs to be greater weight given to polling results released closer to an election than earlier results. Stoetzer et al. accomplished this in their model using a random walk backwards approach [7]. In order to ensure accuracy over time, a machine learning model must accomplish a similar task. Second, by compiling data from several official polling organizations, the resulting data set has multiple frequencies, which can also reduces a model's accuracy. The multiple frequencies are a result of the various sampling methods used by each polling organization. Some samples are taken more frequently than others, some have a greater sample size, and some have a more limited demographic scope. This creates additional noise in the model that is difficult to control or limit.

When examining data from each polling company for the various political parties, all but one party has trends of support that are more or less the same across all of polls from different institutions. Figure 1 illustrates this for the AfD. Therefore, even though the sampling methods vary across different polling organizations, the polling numbers seem to be able to reflect true public opinion at any given point in time. The final polling numbers from all of the institutions are lower but extremely close to the actual results. This is a consistent observation for all other parties, except for Die Linke, which saw a much greater variations in public support that is not well captured through the polls.

There are a few possible solutions to managing this multi-frequency time-series data. Casals, Jerez, & Sotoca

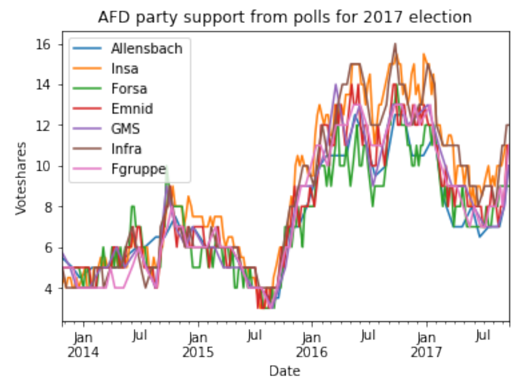

Figure 1. AfD Party Support From Polls for 2017 Election, October 2013 - September 2017

(2009) note the limits of fully aggregating data as it may result in some observations being over or under weighted [2]. Therefore, they recommend only partially aggregating the sample data in order to reduce the amount of frequencies while minimizing the negative effects of aggregation [2].

Cachucho, Meeng, Vespier, Nijssen, & Knobbe (2014) expand on this and test several possible aggregation techniques in the pre-processing phase of a machine learning model. The primary method in their analysis is termed "the accordion method" and includes several feature constructions [1]. The first is a set of aggregate functions based on the mean, median, maximum value, minimum value, standard deviation, inter-quartile range, and root mean squared error [1]. The second involves a grid search of these aggregate functions to find the optimal value for the entire set [1]. Finally, the authors tuned their accordion model to continually search for the best aggregate feature at every step of the modeling process [1]. Ultimately, their model proved to be efficient in the pre-processing stages when working with this kind of data [1].

Beyond the concerns about how to approach this kind of data in the pre-processing phase, many researchers have also struggled to find an accurate forecasting model for multi-frequency time series data. Kourentzes, Petropoulos, & Trapero used a three step process for their analysis: aggregation, forecasting, and then combining the forecasts into one results [5]. Their most novel application was during the aggregation stage where they used time series components as the basis for their aggregation, which considerably simplified the process [5].

Finally, Huber (2017) used polling data to inform predictions about German state elections and had similar roadblocks in the aggregation and pre-processing of the data [3]. The author found that as a result of the differences in poll quality and frequency across German states, some models were more accurate than others [3]. This will likely pose a similar problem for this project.

Based on the methodologies implemented in previous literature, this project seeks to solve these three main data challenges as follows:

1. **Multiple Time Series Sampled at Inconsistent Frequencies:** There are 7 polling institutions included in the data from Stoetzer et al. and each of them conducted their polls at different intervals. There is a gross imbalance in regards to the availability of data points, which makes modeling impossible. Therefore, the data needs to be upsampled by interpolating additional data with moving averages so that the number of data points are consistent across different institutions. After resampling, all data on public support for each party now has daily information from October 23, 2003 up to September 21, 2017 - three days before election day;

2. **Transforming Time Series Data into a Form Suitable for Machine Learning Algorithms:** A unique challenge of time series data is that it cannot be directly split into training and testing sets for machine learning modeling as each data point does not simply belong to a separate variable that can be selected as feature, but instead it's a part of a sequence of information. Therefore, most of the features in the models need to be engineered. For the main features, a rolling window of observations in thirty consecutive days was created. These data points in time steps of thirty days are used to forecast the vote shares three days into the future. For example, if we take our time series data thirty days at a time, we utilize these 30 values of public support for thirty consecutive as the features and the value for three days in the future as the target. Then over time, we train our machine learning models to match the thirty features to the single target. As the rolling window shifts, it takes in the latest information to improve on its accuracy. The horizon of prediction was set as three days into the future as the final value of public support that we have is for September 21, 2017, which is three days before the election day of September 24. The window size can be adjusted and act as a hyperparameter for tuning to test out different model architecture. This approach resembles the random-walk method utilized in Stoetzer et al (2019) as it takes into consideration the influence that polling numbers on different dates might have on the final prediction [7];

3. **Modeling Multiple Time Series with Machine Learning Algorithms:** The final obstacle to overcome is how to input these time series data into each model. There are two possible approaches: creating individual models for data from each party or aggregating the information of all institutions and parties into one combined model. While more data usually means better predictions, as exploratory analysis has shown, the poll numbers from all institutions show surprisingly similar trends and patterns in public support for nearly every party. Therefore, a simple model with data taken as the average of polling numbers from all institutions for a particular party might be able to predict an outcome more accurately as it contains the main signal while reducing random variation noise. Conversely, combining data of all parties into a single model would not be sensible as the innate variation in public support for each individual party will induce high bias in the model and interfere with learning and prediction.

This project employs two approaches to determine which model has the best predictive ability:

1. **Traditional machine learning algorithms for regression**: This method utilizes a simple model set-up with training and test set data to predict party vote shares on election day, with 5 different algorithms - Linear Regression, Decision Tree, Random Forest, Gradient Boosting and XGBoost;

2. **Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) units**: As time series data are temporal, the most appropriate methodology may be a sequence model, such as RNN and LSTM units. The state vector and the cell state in these models allow us to maintain context across a series. With time series data, the closer the data point is to the prediction date, the bigger impact it will have on the final outcome. For this data set, the closer a poll is to an election day, the more powerful its predictive ability is. Stoetzer et al. (2019) account for by this using the backward random-walk technique, therefore, it is necessary to similarly weight the polling data in the machine learning model. By using RNNs and LSTMs, this project similarly carries context from close to far away by using cell state that retains this information through a long learning process.

4. Experiments

Data: Data from the 2017 German federal election was used to create a time series data set that can be processed by the machine learning models. The main data set contains information about the percentage of party support from 7 different polling organizations in Germany. Relevant information includes the date and year the polls were conducted, name of the organization, sample size of polling surveys, party support, and the number of days until the election from the date the survey data was released.

The data was transformed and divided into 7 separate data sets containing aggregated polling information from different institutions for the different political parties represented in the German Parliament.

Evaluation method: The primary assumption is the polling numbers closely reflect the underlying trends in party support from the public at any given point in time, as demonstrated in the methodology. The main goal of the project is to build machine learning and deep learning models that can predict the final vote shares of each of these political parties based on the polling number for their public support. For this purpose, the project will utilize root mean squared error (RMSE) as the main evaluation metric, which was also utilized by Stoetzer et al. in the evaluation of their model. This will allow for a final comparison of the results.

Experimental details: After pre-processing the time-series data following the procedures outlined in methodology, we created the features and labels for modeling. Following this, the training and test sets were created. Especially for time-series data, it is important to split the data at this stage to avoid data leakage of future information that would contaminate the trained model. For our particular problem, we preserved the last 3 months of polls before the election day as the test set and the rest was used as the training set.

However, the final goal of these models is to accurately forecast the election results from 2017. Therefore, after training and validating the model, the ultimate test is forecasting the vote shares of the political parties 3 days into the future based on the last 30 days of polling results.

After obtaining the necessary training and test set for our supervised task, we experimented with five separate Machine Learning models:

- **Linear Regression Model:** This model was chosen as the baseline upon which to improve predictive performance. It was able to deliver a surprisingly small RMSE result despite being a relatively simple and straightforward model;
- **Decision Tree Model:** The model performs with good accuracy out-of-the-box without much tuning on the training and test set. The prediction on the election day result, however, is not much of an improvement on linear regression model as seen in Table 1;
- **Random Forest Model:** A dictionary of hyperparameters was created to search for the optimized options during the testing phase. The model was then looped with a parameter grid for different settings of the hyperparameters and then the scores were saved. The grid search was able to determine the optimal tuning option, which yielded good result;
- **Gradient Boosting Model:** The accuracy and final forecasts of this model are relatively similar to those of Random Forest as the only difference between these two models is that for Gradient Boosting, the decision trees are built successively so that each subsequent tree fits the residual errors from the previous tree;
- **XGBoost Model:** Using GRIDSEARCHCV from the SCIKIT-LEARN library, the optimal hyperparameters were identified during the testing phase. However, the model performs well out-of-the-box with the newly transformed data and achieved high accuracy with the training and test set. However, its forecast for election day results did not improve much in comparison to other models.

After experimenting with Machine Learning models, we implemented a Deep Learning model using Recurrent Neural Network and Long Short-Term Memory units. Through testing, the final model architecture was determined as followed:

- **LSTM layer:** 100 neurons in the first visible layer;
- **Dropout rate:** 20% to prevent over fitting;
- **Output layer:** 1 neuron to forecast public support on the next time step;
- **Loss function:** Mean-squared-error;
- **Optimizer:** Adam;
- **Time step:** 1 day ahead and 1 day before.

In order to input the data into the RNN model architecture, the daily public support data was transformed into 1,429 different time steps representing the 1,429 days from October 23, 2003 to September 21, 2017. The training set consisted of the first two years of data and the test set was the last 699 days. The final prediction was iterated one time step at a time based on the latest information on public support recorded on September 21, 2017.

Results: The results of average RMSE on the election day prediction across different political parties from all five Machine Learning models, and the Recurrent Neural Network models are illustrated in Table 1. Details on each model's performance for individual parties are collected in the appendix of this research.

An interesting conclusion after fitting the different models is the significance of different features to each model. By analyzing feature importance, it is possible to identify

Method	Average RMSE
Linear Regression	1.36
Decision Tree	1.44
Random Forest	1.44
Gradient Boosting	1.46
XGBoost	1.47
RNN	1.32

Table 1. Comparison of Model RMSEs

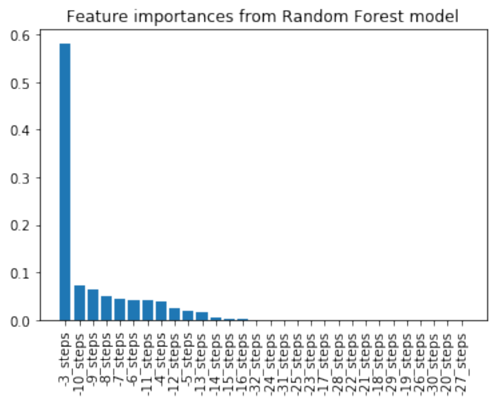

Figure 2. Feature Importance from Random Forest Model for Green Party

the most relevant features that drive the model's final prediction, which helps obtain a better understanding of the model's logic and areas for improvement. A commonality amongst these different models is the importance of knowing public support for a party on the latest day for the prediction 3 days into the future, which is logical since we expect that the latest information would carry the most predictive power. Figure 2 and Figure 3 present the most important features in decreasing importance for the Gradient Boosting and XGBoost models, respectively. The figures compare how each variable affects the model's predictive ability.

5. Analysis

Overall, the machine learning and deep learning models performed as well if not better than those used by Stoetzer et al. Using the AfD results as an example, the total vote share estimates across all the models in this project were slightly higher than Stoetzer et al.'s and closer to the true vote share amount. Additionally, in all but the linear regression and XGBoost models, the RMSE was lower with significant RMSE improvements in the RNN model (see Table 9). For the RNN model overall, the final RMSE was 1.32, which is an improvement of 0.56 over Stoetzer et al. (which retained a 1.88 RMSE for the final forecast). However, it is important to note that this low RMSE was in part due to

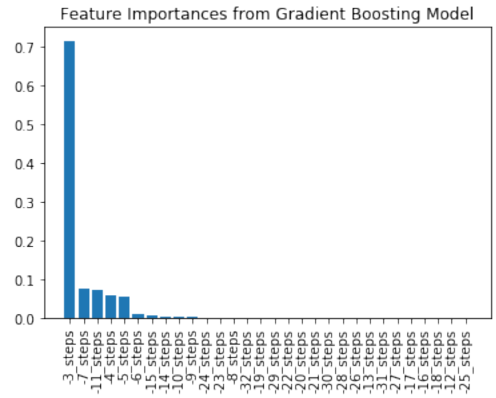

Figure 3. Feature Importance from XGBoost Model for Green Party

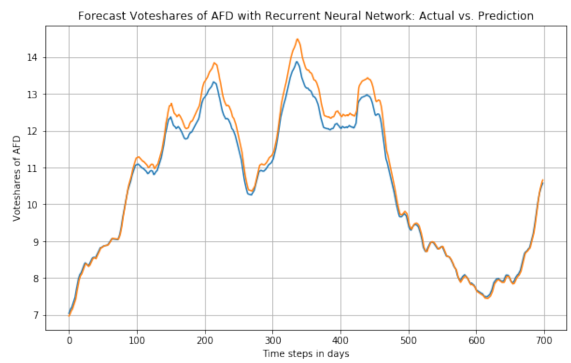

Figure 4. Recurrent Neural Network Model Prediction on AFD public support

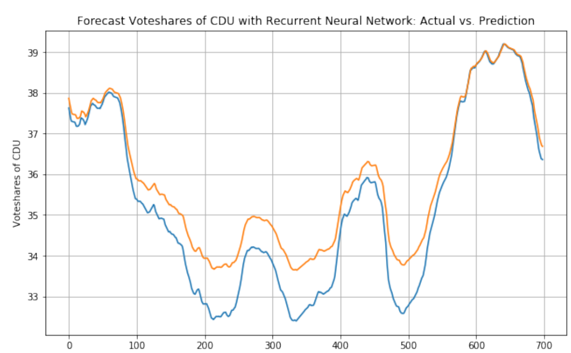

Figure 5. Recurrent Neural Network Model Prediction for CDU public support

the very low RMSEs for the smaller parties in the analysis. RNN estimates for the CDU and SPD had higher RMSEs than Stoetzer et al.'s findings.

Beyond the quantitative improvements, there are significant qualitative improvements in terms of interpretability. Except for the Recurrent Neural Network models, all other Machine Learning models are straightforward and can be interpreted and explained through the examination of methods such as permutation importance, partial plots and SHAP values to understand model insights and what drives its predictive capability. In addition, the models presented in this paper do not require the same level of advanced statistical understanding and assumptions that Stoetzer et al.'s Dynamic Bayesian model does. This makes replication more feasible and allows for the continued usage of this model for future elections, not limited to multiparty system.

6. Conclusions

Our research has overcome the issue of prediction with time-series data by transforming it into a supervised learning problem through the use of resampling, rolling window feature selection, forecasting horizon, and time-step preference. We have successfully implemented five separate Machine Learning models and one Deep Learning model all of which were able to obtain a fairly accurate prediction of the final vote shares of individual political parties based on polling data on the multiparty election system of Germany. The Recurrent Neural Network model with Long Short-Term Memory units outperforms all other models, in terms of accuracy in RMSE metrics.

However, it would be overoptimistic to directly apply these frameworks into any election forecast without analyzing the specific context of the problem. While the models are relatively simple to implement, they also rely on the primary assumption that the data collected truly represent the public support of political parties, which in turn would reflect the final vote shares these parties would receive on election day.

Nevertheless, the success of these methods in predicting the party vote shares of the German election in 2017 presents an exciting venue of exploration in the application of Machine Learning and Deep Learning for election forecasting and can be explored much further to gauge the full extent of their utility.

7. Acknowledgements

We would like to express our appreciation to Dr. Lukas Stoetzer for providing his time, data and insight on guiding our research project in its initial phase.

8. Contributions

Bailey Sutton was in charge of experimenting the machine learning regression algorithm. She also contributed to the literature review and in writing the final report. Maximilian Rekuts was in charge of test-running the baseline machine learning algorithm and collaborating with Dang Ngoc Huy on the technical writing; he also contributed to the literature review. Huy Ngoc Dang was responsible for creating the machine learning algorithm, implementing the RNN and on reporting the results and analysis.

References

[1] R. Cachucho, M. Meeng, U. Vespier, S. Nijssen, and A. Knobbe. Mining multivariate time series with mixed sampling rates. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 413–423, New York, NY, USA, 2014. ACM.

[2] J. Casals, M. Jerez, and S. Sotoca. Modelling and forecasting time series sampled at different frequencies. *Journal of Forecasting*, 28:316 – 342, 07 2009.

[3] S. Huber. *The role of polls for election forecasting in German state elections*. PhD thesis, Universität Passau, 2018.

[4] R. Kennedy, S. Wojcik, and D. Lazer. Improving election prediction internationally. *Science*, 355(6324):515–520, 2017.

[5] N. Kourentzes, F. Petropoulos, and J. Trapero. Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, 30:291–302, 04 2014.

[6] M. S. Lewis-Beck and R. Dassonneville. Forecasting elections in europe: Synthetic models. *Research & Politics*, 2(1):2053168014565128, 2015.

[7] L. F. Stoetzer, M. Neunhoeffer, T. Gschwend, S. Munzert, and S. Sternberg. Forecasting elections in multiparty systems: A bayesian approach combining polls and fundamentals. *Political Analysis*, 27(2):255–262, 2019.

[8] M. Zolghadr, S. A. A. Niaki, and S. Niaki. Modeling and forecasting us presidential election using learning algorithms. *Journal of Industrial Engineering International*, 14(3):491–500, 2018.

[2] [1] [5] [3] [7] [8] [6] [4]

9. Appendix

Method	Estimate	RMSE
Linear Regression	10.68	1.92
Decision Tree	10.79	1.81
Random Forest	10.79	1.81
Gradient Boosting	10.75	1.85
XGBoost	10.72	1.88
RNN	11.23	1.36

Table 2. Estimates & RMSE for AfD Results by Model

Method	Estimate	RMSE
Linear Regression	36.3	3.4
Decision Tree	36.2	3.31
Random Forest	36.17	3.27
Gradient Boosting	36.25	3.35
XGBoost	36.26	3.36
RNN	37.15	4.25

Table 3. Estimates & RMSE for CDU Results by Model

Method	Estimate	RMSE
Linear Regression	21.74	1.24
Decision Tree	21.59	1.09
Random Forest	21.6	1.1
Gradient Boosting	21.65	1.15
XGBoost	21.66	1.16
RNN	22.97	2.47

Table 4. Estimates & RMSE for SPD Results by Model

Method	Estimate	RMSE
Linear Regression	9.55	1.15
Decision Tree	8.79	1.91
Random Forest	8.76	1.94
Gradient Boosting	8.81	1.89
XGBoost	8.78	1.92
RNN	10.08	0.61

Table 5. Estimates & RMSE for FDP Results by Model

Method	Estimate	RMSE
Linear Regression	9.47	0.27
Decision Tree	9.46	0.26
Random Forest	9.48	0.28
Gradient Boosting	9.47	0.27
XGBoost	9.47	0.27
RNN	9.35	0.15

Table 6. Estimates & RMSE for Die Linke Results by Model

Method	Estimate	RMSE
Linear Regression	7.77	1.13
Decision Tree	7.84	1.06
Random Forest	7.76	1.14
Gradient Boosting	7.75	1.15
XGBoost	7.76	1.14
RNN	8.66	0.23

Table 7. Estimates & RMSE for Green Party Results by Model

Method	Estimate	RMSE
Linear Regression	4.54	0.46
Decision Tree	4.36	0.64
Random Forest	4.43	0.57
Gradient Boosting	4.45	0.55
XGBoost	4.45	0.55
RNN	5.12	0.13

Table 8. Estimates & RMSE for Other Parties Results by Model

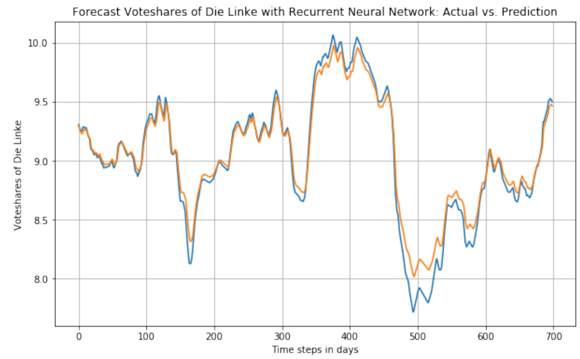

Figure 6. Recurrent Neural Network Model Prediction for Die Linke public support

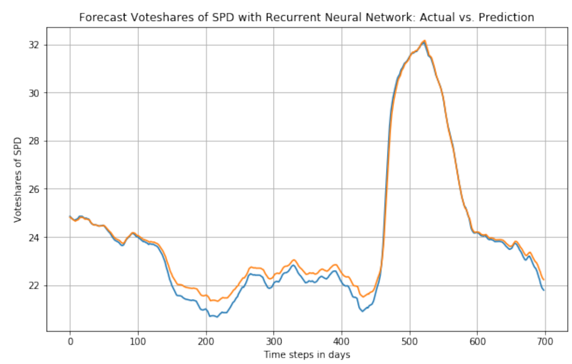

Figure 7. Recurrent Neural Network Model Prediction for SPD public support

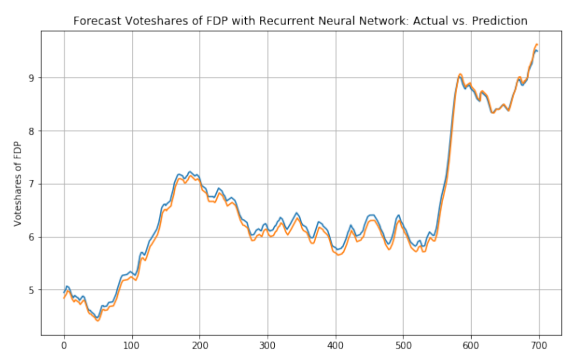

Figure 8. Recurrent Neural Network Model Prediction for FDP public support

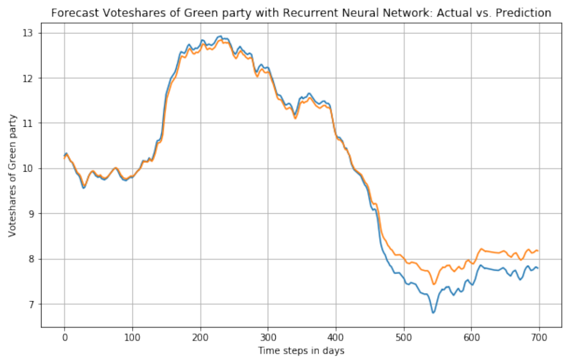

Figure 9. Recurrent Neural Network Model Prediction for Green Party public support

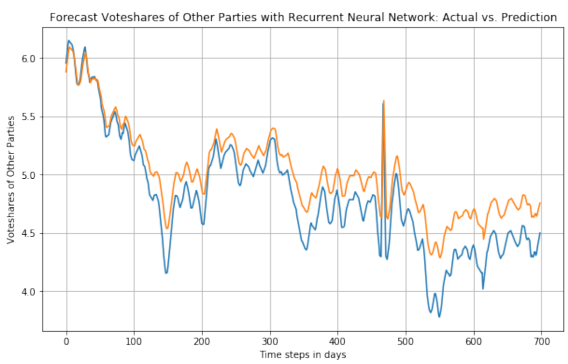

Figure 10. Recurrent Neural Network Model Prediction for Other Parties public support

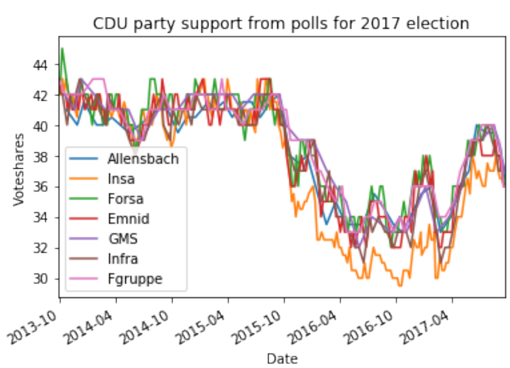

Figure 11. CDU Party Support From Polls for 2017 Election, October 2013 - September 2017

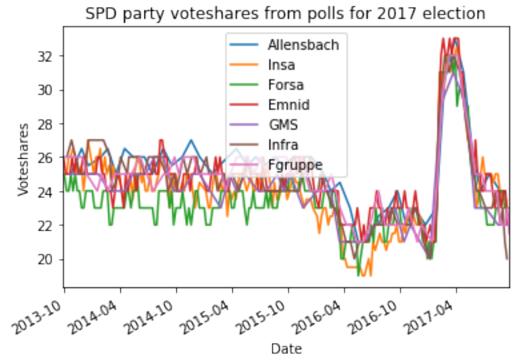

Figure 12. SPD Party Support From Polls for 2017 Election, October 2013 - September 2017

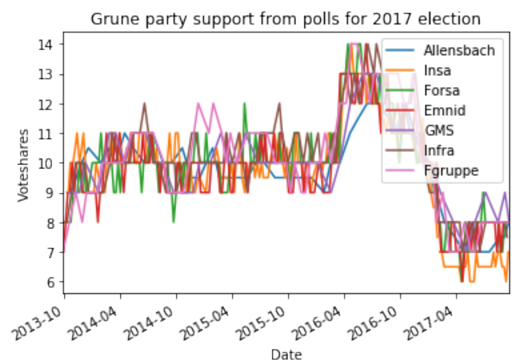

Figure 13. Green Party Support From Polls for 2017 Election, October 2013 - September 2017

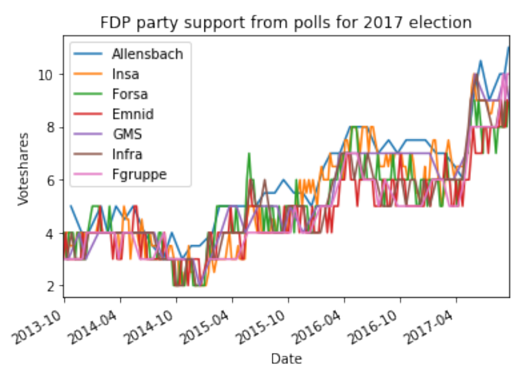

Figure 14. FDP Party Support From Polls for 2017 Election, October 2013 - September 2017