UNIVERSITY OF TARTU

Institute of Computer Science
Software Engineering Curriculum

Erik Räni

Prediction Model for Tendencies in Cybersecurity

Master's Thesis (30 ECTS)

Supervisor: Justinas Janulevičius, PhD

Supervisor: Raimundas Matulevičius, PhD

Tartu 2018

Prediction Model for Tendencies in Cybersecurity

Abstract
Vulnerability disclosure time series data have been previously used to estimate some values in the future. A literature review revealed that researchers have not focused on forecasting the mean Common Vulnerability Scoring System (CVSS) severity scores of Common Weakness Enumeration (CWE) vulnerability types. This could be a problem for software risk management analysts because not knowing the vulnerability categories' upcoming severity scores could result in less accurate risk level assessments. This thesis project provides an R package that addresses the problem. It is eventually used to forecast mean monthly CVSS scores of the year 2018. MAE, RMSE, MAPE and MASE are used to evaluate the accuracy of the forecasts for the years 2016 and 2017. These measures help to choose between the models. Thirteen different types of models are considered when generating the forecasts of 2018 for a subset of 34 CWEs. According to point forecasts, ten CWEs are expected to have "High" severity in 2018.

Keywords: vulnerability, historical data, time series analysis, time series forecasting, trend, seasonality, prediction, forecasting, CVE, Common Vulnerabilities and Exposures, NVD, National Vulnerability Database, CVSS, Common Vulnerability Scoring System, CWE, Common Weakness Enumeration, ARIMA, exponential smoothing, ETS, bagged ETS, neural network, ARFIMA, TBATS, BATS, linear regression, BSM, basic structural model, naïve method, drift method, seasonal naïve method, mean method.

CERCS: T120 (systems engineering, computer technology).

Küberturvalisuse suundumuste prognoosimismudel

Lühikokkuvõte
Haavatavuste avalikustamise aegridasid on varem kasutatud mõnede väärtuste prognoosimiseks tulevikus. Kirjanduse ülevaatest selgus, et teadlased pole varem keskendunud CWE (*Common Weakness Enumeration*) haavatavustüüpide keskmise CVSS (*Common Vulnerability Scoring System*) tõsidusskoori prognoosimisele. Tarkvara riskijuhtimise analüütikute jaoks võib see olla probleem, sest haavatavuskategooriate tulevaste tõsidusskooride mitteteadmisega võivad kaasneda vähem täpsed riskitasemehinnangud. Käesoleva magistritöö raames valmib programmeerimiskeeles R loodud pakett, mis lahendab selle probleemi. Loodud rakendust kasutatakse lõpuks 2018. aasta kuukeskmiste CVSS skooride prognoosimiseks. MAE, RMSE,

MAPE ja MASE arvutatakse välja 2016. ja 2017. aasta prognooside täpsuse hindamiseks, mis aitab eri mudelite vahel valida. 2018. aasta prognooside genereerimisel 34 CWE-le kaalutakse 13 tüüpi mudeleid. Punktprognooside põhjal on 2018. aastal kümne CWE tõsidusaste "Kõrge".

Võtmesõnad: haavatavus, minevikulised andmed, aegridade analüüs, aegridade prognoosimine, trend, hooajalisus, prognoos, CVE, NVD, CVSS, CWE, ARIMA, ETS, *bagged* ETS, närvivõrk, ARFIMA, TBATS, BATS, lineaarne regressioon, BSM, naiivne meetod, triivmeetod, hoojaline naiivne meetod, keskmine meetod.

CERCS: T120 (süsteemitehnoloogia, arvutitehnoloogia).

Contents

1 Introduction

The thesis focuses on forecasting the severity of vulnerability categories based on the past data. The historical data about vulnerabilities are maintained by some entities. Somebody must determine the severity and the category of a given unique vulnerability and publish the results on a specific time. Eventually, it is possible to ask what type of vulnerability at what time in the future has what severity level. The introduction starts with the context of the study. It presents the relevant two vulnerability data maintainers and the exact places from where the vulnerability types, the publication time and the severity are obtained. As a result, the first paragraphs of the introduction contain abbreviations necessary to understand most of the succeeding text: the motivation, the aim, the scope, the research problem and the research question.

This paragraph, technical but necessary part of the introduction, presents the link between two maintainers of vulnerability data. Furthermore, it introduces the important abbreviations used in the thesis. The paragraph also names what vulnerability severity standard and which known list of vulnerabilities are put into use. Common Vulnerabilities and Exposures (CVE) Numbering Authorities (CNAs) assign identifiers (CVE IDs) for unique software vulnerabilities [1]. National Vulnerability Database (NVD) is selected to be the thesis's main data set (Chapter 3.2). NVD has the data from CVE list and some added data, which includes Common Vulnerability Scoring System (CVSS) version 2 base score and the metrics vector used to calculate the score [2]. In addition, NVD adds Common Weakness Enumeration (CWE[1]) vulnerability types to the entries if there is sufficient information and the identified categories belong to the subset of CWEs that are used by NVD [3].

Nearly each vulnerability entry in NVD has a CVSS version 2 vulnerability severity score and a vulnerability category attached to it. Each entry has also a date as a value in NVD XML data file's element `<vuln:published-datetime>`, showing when it was published. The year of this date does not always correspond to CVE ID year's part (`YYYY`) in the format `CVE-YYYY-NNNNN`. According to CVE, the `YYYY` part shows the identifier's assignment year or revelation year to the public but it does not necessarily indicate the year of the vulnerability's discovery [4]. Given the dates from `<vuln:published-datetime>`, CWE types and CVSS severity scores, it is possible to calculate mean monthly CVSS scores for specific CWE categories. One such example is given as a plot for CWE-119 (Buffer Errors) in Figure 1. It shows a regularly spaced (monthly) time series. An early example of time series is the monthly sunspot frequency time series in Arthur Schuster's work back in 1906 [5].

[1]CWE is a trademark of MITRE Corporation. Its license is reproduced in Appendix E.

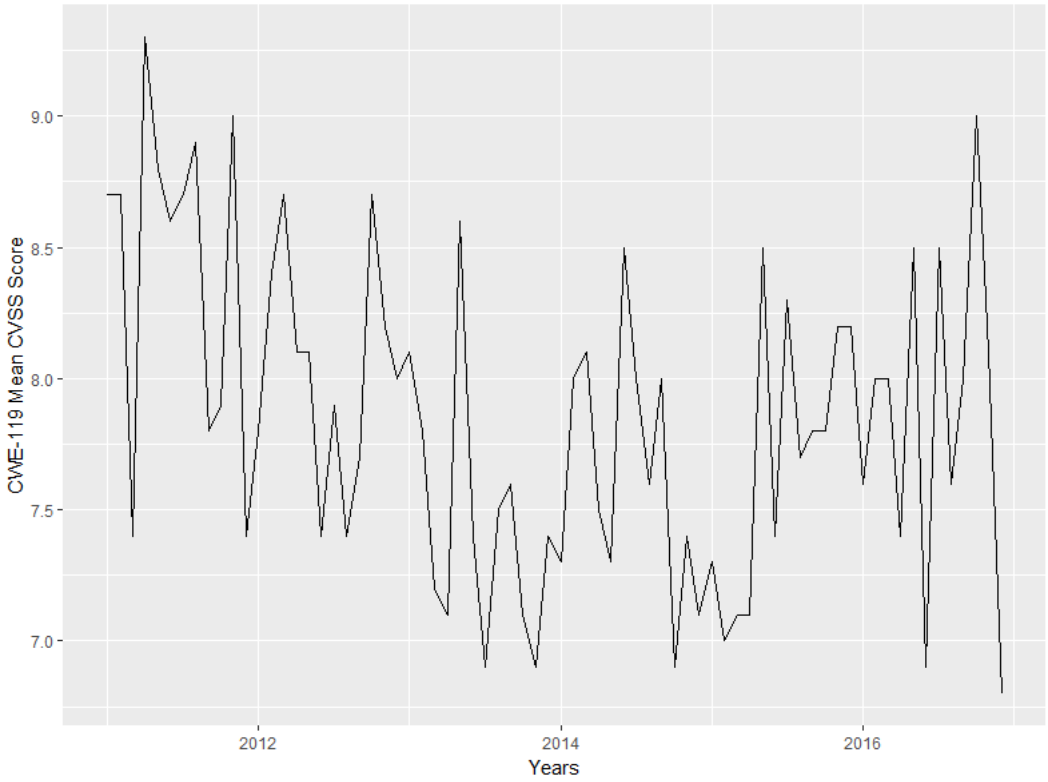

Figure 1: CWE-119 Monthly Time Series

Vulnerability bulletin providers, software application vendors, user organisations, vulnerability scanning and management organisations, security risk management firms and researchers use CVSS for various reasons [6, ch. 1.6]. Security risk management firms calculate their customers' risk levels using CVSS scores as input [6, ch. 1.6]. CVSS version 2 base score 0.0 . . . 3.9 is considered to have "Low" severity, while 4.0 . . . 6.9 means "Medium" severity and 7.0 . . . 10.0 stands for "High" severity [7]. The base metric group of the score is not affected by time or environment [6, ch. 1.1]. It consists of six metrics: Access Vector (AV), Access Complexity (AC), Authentication (Au), Confidentiality Impact (C), Integrity Impact (I) and Availability Impact (A) [6, ch. 2.1]. These can have different values (Table A.1), which are inserted into CVSS base equation to obtain the base score [6, ch. 3.2.1]. NVD analysts are the people assigning CVSS scores to vulnerability entries [7]. When they have not enough information to assign scores, they will give the highest score [7].

NVD provides the vulnerability data with CWE categories and CVSS scores. This can be turned into a monthly time series as presented in Figure 1. Meanwhile, security risk management companies use software vulnerability scores in

their business processes [6, ch. 1.6]. It is possible to estimate how the mean CVSS scores of selected CWEs continue into the future by using time series forecasting [8, ch. 1.4]. If the forecasts and the estimates of the uncertainty are good enough for the companies, then they could calculate their customer organisations' risk levels with an extra future insight. The systematic literature review in Chapter 2 discovered that NVD data have been used for forecasting but not in this type of setting, emphasising the topicality of the thesis.

The thesis project aims to find out future insights about CVSS scores of certain CWE categories. The mean monthly base score observations $y_1, \ldots, y_T$ up to time T are used to forecast the score values y_{T+h} situated in h steps' time in the future [8, ch. 1.7]. No new statistical or artificial intelligence methods are invented. Instead, many existing forecasting functions from an R package 'forecast' [9] are used. The thesis determines how accurate results are produced for 2016 and 2017 and which methods work better than the others given the data from NVD. In order to solve the research problem with a motivation to help security risk management firms, the following central research question is asked: "How to forecast monthly mean CVSS scores of a subset of CWE types?"

The project gives two outcomes: an R package 'nvdr'[2] and an analysis of the forecasting results obtained by applying the package's functions on a selected subset of data from NVD [10]. The forecasting workflow starts with downloading and processing NVD XML 2.0 files. Each vulnerability entry's CVE ID, rejection status, publishing date, CWE category and CVSS version 2.0 base score elements are extracted. Then, a subset of CWE categories of entries are selected. The time series data about each chosen CWE's mean monthly CVSS score are divided into a training set and a test set. After that, the forecasting models produce point forecasts with forecast intervals. The accuracy of the results is measured and analysed. The XML files were downloaded two times: in October 2017 covering CVE IDs from CVE-2011 to CVE-2016 and in January 2018 covering CVE IDs from CVE-2011 to CVE-2017. The forecasts are generated for 2016, for 2017 and for 2018. The forecast accuracy is calculated for 2016 forecasts and for 2017 forecasts. The forecast accuracy for 2018 can be calculated in the future using 'nvdr'.

Chapter 2 investigates the related work as a systematic literature review. It discovers different techniques used for modelling with time series data about vulnerabilities. The review also looks for accuracy measures to assess the outcomes. A gap in the existing research is found. In Chapter 3, the forecasting methods used in the thesis's contribution part are introduced. In addition, several available vulnerability discovery data sets are compared and eventually National Vulnerability Database (NVD) is selected to be the main data source of the thesis. Chapter

[2] `https://github.com/realerikrani/nvdr`

4 addresses the gap found in the related work. The data processing and model fitting is accompanied by the forecast accuracy calculations and analysis. The final chapter provides the conclusion and describes the opportunities for further research. The chapters are focused on providing answers to the questions from Table 1 by answering more specific questions defined at the start of each chapter excluding the introduction.

Table 1: One Research Question per Chapter

Chapter	Question
Chapter 2	How have researchers used time series data about vulnerabilities for forecasting?
Chapter 3	What data are used by which forecasting models in the thesis?
Chapter 4	What do the forecast results reveal?
Chapter 5	How was the central research question answered and how this impacts any further research?

2 Related Work

This subchapter is written as a systematic literature review covering the following steps: development of research questions, selection of data sources and a search strategy, study inclusion and exclusion, data extraction and, finally, analysis of the extracted data. Study quality assessment could also be part of the systematic literature review. However, it is omitted due to the high quality of the data sources. The main goal is to answer the literature review's research questions defined in Table 2 and find the gaps in the existing work. To fulfil that goal, the scientific work related to this master's thesis must be analysed.

Table 2 introduces the research questions of the systematic literature review. These questions are developed with a purpose to discover the ways other authors have used the vulnerability data to model the data or to forecast values in the future. Eventually, the corresponding answers are going to give an overview of previously published similar work and pinpoint the gaps in the state of the art. The questions presented as guidelines in the data extraction form in Table 3 are designed to assist in gathering specific details needed in the process of answering RWQ1 and RWQ2 from Table 2.

Table 2: Questions for the Related Work

Identifier	Question
RWQ1	What forecasting techniques have been used with time series data about vulnerabilities?
RWQ2	How to evaluate the outcome of vulnerabilities-related time series modelling?

2.1 Data Sources and Search Strategy

ScienceDirect, SpringerLink and IEEE Xplore are the digital data sources selected for the literature review. The search strategy involves using keywords "vulnerability", "time series analysis", "time series forecasting", "trend", "prediction", "CVE", "NVD", "CVSS" and "CWE" with logical operators and additional refinement options provided by the sources' search features.

The strategy was put into action. The search performed on IEEE Xplore digital library focused on three data fields: the abstract, the document title and author keywords (Listing 2.1). The operator *ONEAR* was used to find the research, which abstract has the phrase "time series" before the words "analysis" or "forecasting" within the range of ten words. Wildcard characters were utilised while finding the word "vulnerability" together with the words "trend", "forecasting" or "prediction" from the title in order to allow the terms to have different endings.

Keyword data field query was brought to the top level by following the principle *keywords AND (abstract OR title)*. Among the keywords, the wildcard-version of the word "vulnerability" with the other remaining terms "CVE", "NVD", "CVSS" and "CWE" was connected by *OR* operator enforcing at least one of those to be among the keywords of the search results.

Listing 2.1: Command Search on IEEE Xplore

```
(
   "Abstract":
   (
      ("time series" ONEAR/10 "analysis")
      OR
      ("time series" ONEAR/10 "forecasting")
   )
   OR
   "Document Title":
   (
      vulnerabi* AND (trend* OR forecast* OR predict*)
   )
)
AND
"Author Keywords":
   (
      vulnerabi* OR CVE OR "Common Vulnerabilities and Exposures"
      OR NVD OR "National Vulnerability Database" OR CVSS OR CWE
   )
```

ScienceDirect's expert search allowed to use the same approach as IEEE Xplore's command search (Listing 2.2). Only the data fields had different names and operator *PRE*, ScienceDirect's equivalent to IEEE Xplore's *ONEAR*, had to be used. Computer science was selected from all other available sciences to refine the outcome on ScienceDirect.

Listing 2.2: Expert Search on ScienceDirect

```
(
   abs(
        ("time series" PRE/10 "analysis")
        OR
        ("time series" PRE/10 "forecasting")
      )
   OR
   ttl(
        vulnerabi* AND (trend* OR forecast* OR predict*)
      )
)
AND
key(vulnerabi* OR CVE OR "Common Vulnerabilities and Exposures"
OR NVD OR "National Vulnerability Database" OR CVSS OR CWE)
```

The search on SpringerLink, however, was performed differently as its features did not allow to focus specifically on keywords and abstracts. Therefore, previously used terms were simply connected with Boolean operators (Listing 2.3). In addition, a wildcard-version of the word "historical" was added to the query because

it seemed to give results with relevant titles during the experimentation phase of building the query. Finally, computer science was selected as the search discipline and preview-only results were omitted on SpringerLink to reduce the number of findings.

Listing 2.3: Search on SpringerLink

```
vulnera*
AND ("time series" OR histor*)
AND (trend* OR forecast* OR predict*)
AND (CVE OR NVD OR CVSS OR CWE)
```

The search was performed on 9 September 2017. ScienceDirect gave eight results, IEEE Xplore provided 32 results and the search on SpringerLink concluded with 135 results. Next, these 135 results were sorted by relevance and the bottom 68 were discarded as a systematic countermeasure for SpringerLink's poorer search option features compared to the two other data sources' capabilities. Two studies from IEEE Xplore were identified as duplicates of the studies found on SpringerLink and were eliminated.

2.2 Study Selection

The inclusion criteria are derived from the research questions of the related work subchapter. The studies should present techniques that result only in fitted models or models that can be used to forecast some future values. The data that are used by the models should be time series data or time series data with some additional attributes. The process of modelling should be accompanied by accuracy analysis. If some study contradicts any inclusion criterion or only describes the characteristics of the data without presenting models, then it will be excluded.

10 studies out of 107 were included after the study selection was executed as defined above. The titles are presented in the following list:

- Measuring, Analyzing and Predicting Security Vulnerabilities in Software Systems [11],
- Mining Trends and Patterns of Software Vulnerabilities [12],
- Time Between Vulnerability Disclosures: A Measure of Software Product Vulnerability [13],
- Time Series Modeling of Vulnerabilities [14],
- An Empirical Study on Using the National Vulnerability Database to Predict Software Vulnerabilities [15],
- Objective Risk Evaluation for Automated Security Management [16],

- Predicting Severity of Software Vulnerability Based on Grey System Theory [17],
- Using Historical Software Vulnerability Data to Forecast Future Vulnerabilities [18],
- Consensus Forecasting of Zero-Day Vulnerabilities for Network Security [19],
- Big Data for Cybersecurity: Vulnerability Disclosure Trends and Dependencies [20].

2.3 Data Extraction

Firstly, the data extraction form includes the references of each study (Table 3). Secondly, the main goal is determined. Thirdly, the information about the data set is extracted. Finally, the methods to achieve the objective along with the ways to evaluate the outcome are examined based on the guidelines written into the extraction form.

Table 3: Data Extraction Form

Extraction Field	Guideline
Reference	What is the BibTeX entry?
Goal	What is the main goal of the study?
Data Set	Where does the data come from?
Prediction/Trend Modelling	What techniques have been used to fulfil the goal of the study?
Evaluation & Outcome	What techniques have been used to evaluate the fitted models? What are the results of the study?

2.4 Information Extraction

Each of the following paragraphs represents outline information about one study based on the extracted data. The similar outlining technique was used by Mellado et al. in their systematic review of security requirements engineering [21]. As a result, a context for the later discussion is formed. The reference in the first sentence of each paragraph shows the source that is the basis of every sentence in that paragraph.

Alhazmi et al. asked whether the number of potential undetected vulnerabilities could be predicted (this paragraph is based on [11]). Their goal was to model the vulnerability discovery process. The data about the number of known monthly

vulnerabilities of seven operating systems from National Vulnerability Database (NVD) as of 2005 were used to fit a logistic and a linear model. A chi-squared test helped to assess the fit of the models. Overall, the logistic model was showing better goodness of fit results than the linear model.

Murtaza et al. examined the trends of vulnerabilities in order to predict the types of the future upcoming vulnerabilities in a software application (this paragraph is based on [12]). They used the NVD data from 2009 to 2014. That data included Common Weakness Enumeration (CWE) specifying the type of each vulnerability entry. The likelihood values of different types of vulnerabilities in particular applications and across all software applications, separately, for each year were calculated using the likelihood equations defined in the study. After that, the researchers applied the Cox Stuart trend test to find trends in the changes of likelihood values across years on both occasions. However, the test results lead to conclusions that there was insufficient evidence to suggest a significant increase or decrease of trends of software vulnerabilities both across all applications and within selected applications over the years. Next, an n-gram pattern extraction algorithm was used to discover vulnerability type occurrence patterns across applications. It was found that given an application, it is likely that identical vulnerability categories occur multiple times simultaneously. Finally, the extracted n-grams were used to build an n-gram model that enabled to predict the type of next vulnerability in an application if some of the previous vulnerability types were already known. The 2-grams gave the best overall results when compared with 3-grams, 4-grams and 5-grams based on evaluation using precision and recall.

Johnson et al. proposed a measure "time between vulnerability disclosure (TBVD)" that estimates an experienced vulnerability analyst's effort to find a new vulnerability in a software application (this paragraph is based on [13]). Then autoregressive AR(1) prediction models were fitted to the applications' TBVD time series data obtained by processing the vulnerability data (from NVD) and the data about specific analysts who had discovered the vulnerabilities (from SecurityFocus Vulnerability Database). Pearson correlation coefficient was used to assess the mean TBVD forecasts of products. Prediction accuracy was found to be better for shorter-term forecasts than longer-term forecasts.

Roumani et al. used NVD data from January 2006 to December 2013 to create models that can predict the number of future vulnerabilities for five web browsers and also a model for predicting the number of all vulnerabilities generally (this paragraph is based on [14]). Two methods were put into use: autoregressive integrated moving average (ARIMA) and exponential smoothing. The latter method meant that the researchers fitted Holt-Winters additive model and the simple seasonal model, two examples of exponential smoothing models. The autocorrelation function and the augmented Dickey-Fuller test allowed to assess the stationar-

ity of the data. The outcome of Ljung-Box test on residuals was utilised as the indicator of the model's adequacy. The plots of original values versus fitted values for the five web browsers showed good fit of the models. The models were also evaluated with stationary R^2. Finally, forecasting values were generated and the prediction accuracy was measured by symmetric mean absolute percent error (SMAPE). The prediction error was smallest with a value of 12% for the model that predicted the number of all vulnerabilities in general. The best vulnerability prediction model for a web browser (Firefox) had an SMAPE of 37%. The models for other browsers, however, had a higher than 60% SMAPE which made them inaccurate. The researchers found out that a time series' level was a significant parameter, while the seasonality parameter and the trend parameter appeared to be insignificant parameters of the prediction models.

Zhang et al. came up with a metric "time to next vulnerability" (TTNV), the time between the discoveries of consecutive vulnerability discoveries within an application (this paragraph is based on [15]). They built models that predict future TTNVs. The NVD data starting from 2005 was used as a basis from where the affected software versions and publication dates were extracted. Regression functions (linear and least median square among other functions) were used for predicting TTNV as a number of days to the next vulnerability. Several classification functions were used for predicting TTNV as a binned entity containing bins with ranges of values. The regression algorithms were evaluated by using Correlation Coefficient, Root Mean Squared Error (RMSE) and Root Relative Squared Error (RRSE) while Correctly Classified Rate provided a means for evaluating the performance of predictive classification models. The authors experimented with adding CVSS (Common Vulnerability Scoring System) metrics to the models as predictive features. Eventually, the researchers found that the data in NVD was not good enough for their approach and believed that at the time of writing the study, in the year 2011, it was unlikely that usable prediction model could be built like that.

Ahmed et al. provided a security metric framework with multiple components that could be ultimately combined into one measure that helps to evaluate risks (this paragraph is based on [16]). NVD data was used for presenting a validation for the framework. The researchers thought of a computer network system as a combination of networks and services. Historical Vulnerability Measure (HVM) and Probabilistic Vulnerability Measure (PVM) were two of the defined components of the framework. HVM uses the historical vulnerability data to express a service's past vulnerability proneness. A collection of services' PVM combines the probability of a vulnerability publication in the upcoming period and its expected severity. By splitting data into training and testing sets, HVM gave more accurate results when more historical data was available, being up to 83% accurate. The

vulnerability prediction component of the framework was up to 78% accurate.

Geng et al. proposed vulnerability severity prediction model (this paragraph is based on [17]). Based on Grey System Theory, grey prediction models are believed to outperform traditional statistical methods when there are little data available according to the researchers. The authors used vulnerability data which include severity scores from 0.0 to 10.0 sourced from Chinese[3] Common Vulnerabilities and Exposures (CVE) page. The data source had poor vulnerability data about software applications Lynx and Xpdf. The vulnerability time series data were seen as oscillatory data which might be too complex for the classical grey prediction model GM(1,1). Therefore, improvements were presented. The researchers provided mathematical transformation steps to smooth the initial data, to use GM(1,1) for prediction and to restore the data and obtain the predicted severity. Root Mean Square Error (RMSE) and Mean Relative Error (MRE) were used to measure the prediction quality. The improved usage of GM(1,1) delivered better results than the classical GM(1,1).

David Last put regression models and a classification model together into a combined model predicting the cumulative number of future vulnerabilities (this paragraph is based on [18]). Linear regression model, quadratic regression model and a regression model forecasting the average of linear and quadratic forecasts were trained with the data from NVD using a variation of time horizon trend strategies and different training period lengths. Eventually k-NN classification decided which of the regression models showed the best performance at a specific point in time given a specific subset of the data set. Mean Absolute Percent Error (MAPE), Root Mean Square Percent Error (RMSPE), Edit Distance on Real Sequences (EDR), Euclidean Distance (ED), Time Warp Edit Distance (TWED) and Dynamic Time Warping (DTW) were the distance measures used by the classification. The models based on the first two distance measures worked better over the data with consistent trends while the models based the last three measures showed better results on more jumpy data.

David Last aimed to develop models to forecast the number, location and severity of vulnerabilities in the future of 12 to 24 months (this paragraph is based on [19]). He pooled together 81 forecast models which were trained with data from NVD starting from January 2000 and went up to some point in the future so that the time between years 2012 to 2015 could accommodate the forecasting periods. The 81 models came from three model suites: Composite Regression Models, Machine Learning over Cumulative Vulnerabilities Models and Machine Learning over Monthly Vulnerabilities Models. The first two model suites used cumulative vulnerability discovery data of their training data sets while the last one used the monthly number of discovered vulnerabilities of its training data set

[3]http://cve.scap.org.cn/

instead. Inside each model suite, experimenting with different input parameters was the cause of generating eventually the total number of 81 forecasting models across all three suites. Root Mean Square Percent Error (RMSPE) was used as their individual measure of accuracy. Finally, the pool of models, from where poorly performing forecast models were excluded based on an RMSPE distance measure number, were used to calculate an optimal consensus forecast. Other distance measures were used as well for experimentation purposes while the optimal consensus parameters were searched. The consensus forecast was evaluated using the Absolute Endpoint Percent Error (AEPE). The author was satisfied with the consensus method results.

Tang et al. had a main goal of finding dependencies between general vulnerability disclosure time series data from NVD and each of the following subgroups of that data: Buffer Overflow, Memory Corruption and Gain Privilege (this paragraph is based on [20]). First, preliminary stationarity analysis was performed with Kwiatkowski-Phillips-Schmidt-Shin (KPSS) statistics at the 5% level and it showed non-stationarity in some groups. ARIMA models were fitted with different combination of parameters to capture the mean time series data behaviour. ARIMA models were evaluated by calculating the Akaike's Information Criterion (AIC), the Bayesian Information Criterion (BIC) and Root Mean Squared Error (RMSE). After that, heteroskedasticity was addressed by fitting GARCH models and evaluating the outcomes with BIC and AIC. In the end, Gaussian and Student-t copula models captured the dependency relationship stated in the main goal of the study. AIC served as the goodness of fit for the copulas. Student-t copula handled the unsymmetrical data with tail dependencies better than the Gaussian copula.

2.5 Discussion

Table 5 presents the literature's main goals, data sources, models, model assessment methods and beneficial, concluding messages in a structured way to compare the approaches. In some cases, multiple data sources, models and assessment techniques were used. For those cases, commas separate the list items inside the table cells.

The studies about building models based on vulnerability time series data rely mostly on National Vulnerability Database (NVD). Other data sources are used in two cases out of ten. In one of the two cases, the other source complements NVD. This indicates that NVD is a usable option of vulnerability time series data. However, Zhang et al. concluded their study in 2011 with the understanding that the data from NVD was with low quality, causing their predictive models to be affected negatively [15]. All the other studies didn't end with such a statement. Instead, usually some of the researchers' ideas worked quite well and the achieved

progress got mentioned. Most of the selected studies, seven to be specific, were published after the year 2011 (Figure B.1).

Each study had its main goal. Alhazmi et al. created models that encapsulated the trends of the discovered cumulative number of vulnerabilities in selected operating systems [11]. Tang et al. modelled dependencies between groups of vulnerabilities. In order to do that, the behaviour of time series data needed to be modelled in the process [20]. Models from Roumani et al. had the ability to predict the number of future vulnerabilities generally for all software and for web browsers [14]. David Last predicted the cumulative number of future vulnerabilities in application groups [18]. He also built models with an aim to forecast the number, location and severity of the upcoming vulnerabilities of application groups [19]. Severity is what Geng et al. predicted for applications as well [17]. Vulnerability's expected severity was also computed when using the risk evaluation security metric framework proposed by Ahmed et al. [16]. Murtaza et al. built models that could guess the next type of vulnerability in an application given that the types of some of the previous vulnerabilities in the application were already known [12]. Johnson et al. predicted an application's TBVD[4] values while Zhang et al. predicted an application's TTNV[5] values [13, 15]. The main goals could be categorised into five groups: type prediction, researcher-defined time-related measure prediction, model fitting (trend capture), prediction of the number of vulnerabilities and severity prediction (Table 4).

Table 4: Main Goal Categories

Category	Studies
Type Prediction	[12]
Prediction of researcher-defined time-related measure	[13, 15]
Model Fitting (Trend Capture)	[11, 20]
No. of Vulnerabilities Prediction	[14, 18, 19]
Severity Prediction	[16, 17, 19]

There were different models used to achieve the objectives of the studies. Some of the studies combined multiple models together, some experimented with building individual models to eventually pick the one that shows the best results. Regression, classification, autoregression, exponential smoothing and other techniques were put to use. Modelling and predicting also needed to be assessed. There exist many assessment approaches: coefficients, distance measures and error calculations. The predicted values created by fitted models usually are compared to the

[4] "time between vulnerability disclosure"

[5] "time to next vulnerability"

test set values. Not every model and assessment technique can be always used for all kinds of time series data. For example, some models require the data to be stationary.

Table 5: Comparison Table

	Main Goal	Data Source	Models	Assessment	Message
Alhazmi et al. [11]	Model vulnerability discovery process.	NVD	Logistic, linear	Chi-squared test	Logistic model captured trends better than the linear model.
Murtaza et al. [12]	Predict future vulnerabilities' types.	NVD	N-gram	Precision, recall	2-grams outperformed higher order n-grams.
Johnson et al. [13]	Predict application's future TBVD values.	NVD, SecurityFocus Vulnerability Database	AR(1) prediction	Pearson correlation coefficient	AR(1) had higher accuracy for short-term forecasts.
Roumani et al. [14]	Predict the number of future vulnerabilities for web browsers and in general.	NVD	ARIMA, exponential smoothing	SMAPE	Prediction model of all vulnerabilities had a better SMAPE value than the models for web browsers. Time series' level was significant while seasonality and trend were insignificant parameters.
Zhang et al. [15]	Predict application's future TTNV values.	NVD	Regression, classification	Correlation Coefficient, RMSE, RRSE, Correctly Classified Rate	NVD was seen as a low quality data source affecting models badly.
Ahmed et al. [16]	Provide a security metric framework to evaluate risks.	NVD	Historical Vulnerability Measure (HVM), Probabilistic Vulnerability Measure (PVM)	Accuracy	HVM accuracy was higher with more historical data.
Geng et al. [17]	Propose vulnerability severity prediction model.	Chinese Common Vulnerabilities and Exposures (CVE)	Improved GM(1,1) grey prediction	MRE, RMSE	GM(1,1)'s predictive capability can be boosted by data smoothing.
Last [18]	Create vulnerability forecast models for specific software packages.	NVD	Linear regression, quadratic regression, linear-qyadratic-average regression, k-NN classification	MAPE, RMSPE, EDR, ED, TWED, DTW	It is possible to combine k-NN classification and linear regression to build useful prediction models.

Table 5: Comparison Table Continued

	Main Goal	Data Source	Models	Assessment	Message
Last [19]	Forecast the number, location, and severity of future vulnerabilities.	NVD	Composite regression, machine learning over monthly and cumulative vulnerabilities	RMSPE, DTW, TWED, EDR, AEPE	It is possible to combine forecast models by using consensus forecast methods to build useful consensus forecast models. Models based on smoother data set are more accurate than models based on inconsistent monthly vulnerability discovery rates.
Tang et al. [20]	Find dependencies between general vulnerability disclosure time series data and its subgroups.	NVD	ARIMA, GARCH, copula	AIC, BIC, RMSE	ARIMA models do not capture heteroskedasticity but the mean behaviour instead.

As mentioned before, Zhang et al. found that the data from NVD was of low quality and it affected the model performance as well [15]. When other researchers concluded their studies, potentially useful information was shared. 2-grams outperformed other higher order n-grams in vulnerability type prediction [12]. Prediction of "time between vulnerability disclosure" with AR(1) had higher prediction accuracy for short-term forecasts [13]. The logistic model captured the trends of cumulative number of operating systems' vulnerabilities better than the linear model [11]. ARIMA and exponential smoothing prediction models showed that seasonality and trend of vulnerability time series were insignificant parameters but the level of time series was significant for the models [14]. What is more, the prediction of the number of all vulnerabilities in general was more accurate than predicting the number of vulnerabilities in a specific web browser [14]. Historical Vulnerability Measure (HVM) was more accurate when it had access to more historical data [16]. Smoothing the data for GM(1,1) improved its predictive capability [20]. It is possible to create combined or consensus models that rely on a pool of different individual models and select the most appropriate ones to come up with the final forecast [18, 19]. Based on these results, a substantial amount of potentially smoothed historical vulnerability time series data could be used for short-term forecasts that are based on models more complex than linear models, for example, ARIMA and exponential smoothing. For further improvements, combined or consensus forecasts might be reasonable options.

The research questions of the related work's literature review (Table 2) can be answered. RWQ1 was about finding out what models had been built using time series data of vulnerabilities. There are models that capture trends (or simply fit data), predict vulnerabilities' severity, type, amount and newly defined time-

related measures about vulnerability occurrences (Table 4). These models have been built using a variety of techniques, for example, regression, classification, autoregression, exponential smoothing and combination of the methods. RWQ2 was about finding out how to evaluate the outcome of time series modelling of vulnerabilities. The outcomes of fitting a model are evaluated by comparing actual training data to the fitted values (one-step forecasts of the values of the training set with estimated parameters based on the entire training set [8, ch. 3.3]). The outcomes of forecasting are evaluated by comparing the predicted values to the test data. The specific examples of assessment approaches can be found from the penultimate column of Table 5.

The systematic literature review identified a gap in the previous research that is addressed in this study. Although researchers have created models, which predict the vulnerabilities' severity, they haven't focused on building models that especially focus on forecasting the monthly mean CVSS severity scores of CWE vulnerability categories. The next chapter introduces the forecasting models used in this thesis to address the identified gap. Furthermore, a vulnerability data set that provides the input to the models is selected.

3 Background

Specific time series forecasting models are used with selected data in order to forecast the mean CVSS severity scores of CWE vulnerability categories. Table 2 introduces the research questions of the background investigation. The questions help to understand how the forecasting techniques work and what data set should be used as input.

Table 6: Research Questions for the Background

Identifier	Question
BQ1	How are the forecasting methods or models mathematically represented?
BQ2	What vulnerability data set should be used for creating the forecasting model for tendencies in cybersecurity?

3.1 Forecasting

This chapter focuses on explaining the basics of different methods and models. It does not aim to provide detailed explanations about calculating forecast intervals. There is a difference between forecasting *methods* and forecasting *models*. Methods produce point forecasts, while models also provide the possibility to calculate forecast intervals (in addition to generating the point forecasts) [22, ch. 1.2]. A point forecast is a specific value in the future, a mean or median of the probability distribution [22, ch. 1.2]. Forecast intervals indicate the uncertainty of point forecasts: it is expected that $n\%$ of the future values will belong to $n\%$ forecast intervals surrounding the point forecasts [8, ch. 3.5].

Given an actual value of an observation y_t at time t and a fitted value $\hat{y}_{t|t-1}$ calculated by a time series model, then the difference

$$e_t = y_t - \hat{y}_{t|t-1} \tag{1}$$

is often called a residual [8, ch. 3.3]. However, there exists models such as those using Box-Cox transformation for which finding residuals require the use of different equation than (1) [23, ch. "Different types of residuals"]. A model is unbiased and it has been able to use the relevant information from the training data when the residuals are uncorrelated and have zero mean [8, ch. 3.3]. There exist calculations that allow to find multi-step forecast intervals [8, ch. 3.5]. The calculations assume that the residuals are uncorrelated and normally distributed in order to not obtain incorrect forecast intervals [8, ch. 3.5, ch. 8.8].

When training data $\{y_1, \ldots, y_T\}$ is used to fit the model and the model's forecasts are compared to the test data $\{y_{T+1}, y_{T+2}, \ldots\}$ not used in fitting the model,

then

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T} \tag{2}$$

is the equation for finding forecast errors, where $\hat{y}_{T+h|T}$ represents the forecasted value [8, ch. 3.4].

Sometimes by transforming the time series, it is possible to come up with simpler models that have increased accuracy [8, ch. 3.2]. Box-Cox transformations,

$$w_t = \begin{cases} \log(y_t) & \text{if } \lambda = 0; \\ \frac{(y_t^\lambda - 1)}{\lambda} & \text{otherwise,} \end{cases} \tag{3}$$

remove the fluctuations in the size of seasonal variation in observations $y_1, \ldots, y_T$ when given a transformation parameter λ [8, ch. 3.2]. The forecasts must be back-transformed later with

$$y_t = \begin{cases} \exp(w_t) & \text{if } \lambda = 0; \\ (\lambda w_t + 1)^{\frac{1}{\lambda}} & \text{otherwise,} \end{cases} \tag{4}$$

which can be bias-adjusted if the forecast distribution's mean is needed instead of the median [8, ch. 3.2].

The following subchapters of Chapter 3.1 provide collectively an answer to the research question BQ1 from Table 6. The thesis project uses four benchmark models: average, naïve, seasonal naïve and drift model. The average model uses the mean of the past observations' values as forecasts [8, ch. 3.1]. The naïve model uses the value of the training set's last observation as the forecast [8, ch. 3.1]. The seasonal naïve model applied on a monthly data with frequency 12 uses the last known month as the forecast of the same month in the future: when forecasting January 2016 with training set ending in December 2015, then January 2015 is the forecast for January 2016 [8, ch. 3.1]. The drift model uses the value of the training set's last observation and also considers the training set's change [8, ch. 3.1]. Linear regression models a linear relationship between the forecast variables and the predictor variables, which could be trend and seasonal variables [8, ch. 5]. ETS models generate forecast by giving weights to past observations: how much the latest observations affect the forecasts and how much should the observations of the more distant past be taken into account [8, ch. 7]. ETS models can be represented as error, trend and seasonal components [8, ch. 7.7]. BSM can also be considered as components, but BSM models work differently than ETS models and are less general [24]. Bagged ETS models create many similar pieces of time series from the initial time series [25]. Based on these, multiple ETS models produce forecasts, which are put together and used as one source of forecasts [25]. ARIMA models use the lagged observations and lagged errors as predictors [8, ch. 8]. ARIMA models might require the time series to be differenced by certain

amount of times [8, ch. 8]. ARFIMA models are similar to ARIMA models, but they allow the differencing of time series by a non-integer amount of times [9, `arfima` doc.]. Feed-forward neural network autoregression models are based on layers of neuron nodes [8, ch. 11.3]. They are iteratively trained [8, ch. 11.3]. The neurons linearly combine inputs, change the results with additional calculations and then output the results [8, ch. 11.3]. BATS models include data transformation (Box-Cox transformation), trend components, seasonal components and consider the lagged observations and lagged errors (observations and errors are combined as ARMA errors) [26]. TBATS models additionally use trigonometric terms in the seasonal component [26].

3.1.1 Benchmark

The simple forecasting methods

$$\hat{y}_{T+h|T} = \frac{y_1 + \cdots + y_T}{T}, \tag{5}$$

$$\hat{y}_{T+h|T} = y_T, \tag{6}$$

$$\hat{y}_{T+h|T} = y_{T+h-km} = y_{T+h-(\lfloor (h-1)/m \rfloor + 1)m}, \tag{7}$$

$$\hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + h\left(\frac{y_T - y_1}{T-1}\right) \tag{8}$$

for time series observations $y_1, \ldots, y_T$ with length T set the benchmark forecasts in the thesis [8, ch. 3.1]. Average method (5) calculates the future h-step estimate $\hat{y}_{T+h|T}$ by finding the mean of the values of past observations $y_1, \ldots, y_T$ [8, ch. 3.1]. Naïve method (6) assigns y_T's value, the value of the last observation, to all future estimates [8, ch. 3.1]. Seasonal naïve method (7) takes into account the seasonal period m and assigns $\hat{y}_{T+h|T}$ the value corresponding to the last observation's value of the same season picked from $y_1, \ldots, y_T$ [8, ch. 3.1]. Drift method (8) sets the forecast to equal the sum of the last observation's value y_T and the drift $h\left(\frac{y_T - y_1}{T-1}\right)$ [8, ch. 3.1].

3.1.2 Linear Regression

In simple linear regression equation

$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t, \tag{9}$$

the value of y_t is calculated by using the predictor variable x_t, intercept coefficient β_0, slope coefficient β_1 and the error term ε_t [8, ch. 5.1]. In multiple linear regression equation

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t, \tag{10}$$

there are predictor variables $x_1, \ldots, x_k$, coefficients β_0 and $\beta_1, \ldots, \beta_k$ and the error term ε_t [8, ch. 5.1], altogether more predictors and more coefficients than in simple linear regression equation. Equation (10) can be written in a matrix form [8, ch. 5.7]:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{11}$$

meaning in the case of T observations

$$\begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{2,1} & \ldots & x_{k,1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1,T} & x_{2,T} & \ldots & x_{k,T} \end{bmatrix} \times \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_T \end{bmatrix}. \tag{12}$$

The coefficient values $\beta_0, \beta_1, \ldots, \beta_k$ are estimated using the least squares estimation

$$\sum_{t=1}^{T} \varepsilon_t^2 = \sum_{t=1}^{T} (y_t - \beta_0 - \beta_1 x_{1,t} - \beta_2 x_{2,t} - \cdots - \beta_k x_{k,t})^2, \tag{13}$$

which means selecting $\beta_0, \beta_1, \ldots, \beta_k$ that give the smallest possible sum of squared errors value [8, ch. 5.2]. Forecasts of y at time t with estimated coefficients are expressed in the following way [8, ch. 5.2]:

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{1,t} + \hat{\beta}_2 x_{2,t} + \cdots + \hat{\beta}_k x_{k,t}. \tag{14}$$

Equation (14) can be used for calculating the fitted values of y corresponding to predictor variables $x_{1,t}, \ldots, x_{k,t}$, where $t = 1, \ldots, T$ [8, ch. 5.2]. In order to compute forecasts h-step into the future, lagged values (observations that are h time periods trailing the observation of y) are used as predictors in the equation [8, ch. 5.6]:

$$y_{t+h} = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \varepsilon_{t+h}. \tag{15}$$

Given $t = 1, \ldots, T$, predictor $x_{1,t} = t$ could be used to model a linear trend [8, ch. 5.3]. For example, for simple linear regression [8, ch. 5.3]:

$$y_t = \beta_0 + \beta_1 t + \varepsilon_t. \tag{16}$$

Seasonal dummy variables $d_{1,t} \ldots d_{k,t}$ of categorical predictor variables can also be used as predictors. [8, ch. 5.3]. Equation

$$y_t = \beta_0 + \beta_1 t + \beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t} + \varepsilon_t \tag{17}$$

has the coefficient β_1 associated with a linear trend predictor variable t, β_2 associated with a dummy variable $d_{2,t}$, β_3 with $d_{3,t}$ and β_4 with $d_{4,t}$. Dummy variables $d_{2,t}, d_{3,t}, d_{4,t}$ represent the second, the third and the fourth quarter [8, ch. 5.3]. A dummy variable of specific quarter obtains the value 1 when t belongs to the quarter [8, ch. 5.3]. Otherwise, the dummy variable has the value 0 [8, ch. 5.3]. The first quarter is omitted. Therefore, once the coefficients associated to the dummy variables are estimated, they show the difference from the omitted dummy variable [8, ch. 5.3]. Similarly, it is possible to use seasonal dummy variables for weekdays, "yes" and "no" answers and months [8, ch. 5.3]. It is assumed that the errors $\varepsilon_1, \ldots, \varepsilon_T$ of linear regression model (10) have mean zero, have no autocorrelation and are unassociated with $x_1, \ldots, x_k$ [8, ch. 5.1]. It is also assumed that $x_1, \ldots, x_k$ can be controlled and hence are not random [8, ch. 5.1].

3.1.3 ETS and BSM

There exist 15 exponential smoothing methods with different trend and seasonal components [8, ch. 7.4]. There could be an additive trend (A), an additive damped trend ($\mathrm{A_d}$), a multiplicative trend (M), a multiplicative damped trend ($\mathrm{M_d}$) or no trend (N) [8, ch. 7.4]. There could be an additive seasonality (A), a multiplicative seasonality (M) or no seasonality (N) [8, ch. 7.4]. Table C.1 presents the combinations of possible trend and seasonal components (T,S), where $\mathrm{T} \in \{\mathrm{N}, \mathrm{A}, \mathrm{A_d}\}$ and $\mathrm{S} \in \{\mathrm{N}, \mathrm{A}, \mathrm{M}\}$. Multiplicative trend (M) and multiplicative damped trend ($\mathrm{M_d}$) are omitted because of their inclination towards "poor forecasts" [8, ch. 7.4]. For each combination of (T,S) the table gives the corresponding equations of the h-step point forecast from time t, the slope at time t, the level at time t and the seasonal component at time t. For example, in case of the method with no trend and no seasonality (N,N), there are no slope b_t and no season s_t and the point forecast $\hat{y}_{t+h|t}$ is equal to the level at time t.

The equations in Table C.1 use smoothing parameters $0 \leq \alpha \leq 1$ [8, ch. 7.1], $0 \leq \beta^* \leq 1$ [8, ch. 7.2], $0 \leq \gamma \leq 1 - \alpha$ [8, ch. 7.3] and $0 < \phi < 1$ (usually $0.8 < \phi < 0.98$) [8, ch. 7.2] estimated by using techniques such as minimising the sum of squared errors (SSE) [8, ch. 7.1] or by maximising the likelihood (MLE) [8, ch. 7.6]. The methods with seasonality components feature m (the seasons count of a year) and a final year's seasonal component variable $h_m^+ = \lfloor (h-1)/m \rfloor + 1$ [8, ch. 7.4]. It is also necessary to estimate ℓ_0, b_0 and $s_0, s_{-1}, \ldots, s_{-m+1}$ [8, ch. 7.6]. The forecast methods from Table C.1 produce point forecasts as do statistical state space models [8, ch. 7.5]. The state space models, however, produce forecast intervals in addition to point forecasts [8, ch. 7.5].

The methods from Table C.1 have corresponding state space models forming the ETS statistical framework (Table C.2), which take into account the "normally and independently distributed" error term $\varepsilon_t \sim \mathrm{NID}(0, \sigma^2)$ [8, ch. 7.5]. In Table

C.2, the observation, the slope, the level and the seasonal component at time t are presented as equations. Some of these use the parameter $\beta = \alpha\beta^*$ [8, ch. 7.5]. The models are discovered by combining error, trend and seasonal components (E,T,S), where T $\in \{\mathrm{N}, \mathrm{A}, \mathrm{A_d}\}$, S $\in \{\mathrm{N}, \mathrm{A}, \mathrm{M}\}$ and E $\in \{\mathrm{A}, \mathrm{M}\}$ [8, ch. 7.5]. As a consequence, these models are called ETS ("ExponenTial Smoothing") models [8, ch. 7.5]. The models with multiplicative errors are numerically unstable when they are given a time series containing non-positive values [8, ch. 7.6]. Additionally, the models (A,N,M), (A,A,M), and (A,$\mathrm{A_d}$,M) might cause numerical difficulties [8, ch. 7.6] and are coloured in Table C.2.

Let L represent a model's likelihood and k the model's number of estimated parameters, then Akaike's Information Criterion (AIC) for ETS

$$\mathrm{AIC} = -2\log(L) + 2k, \tag{18}$$

Bayesian Information Criterion (BIC) for ETS

$$\mathrm{BIC} = \mathrm{AIC} + k[\log(T) - 2] \tag{19}$$

and AIC with a correction "for small sample bias" for ETS

$$\mathrm{AIC_c} = \mathrm{AIC} + \frac{k(k+1)}{T-k-1} \tag{20}$$

are available equations for selecting the best model within ETS framework for a particular case [8, ch. 7.6]. Forecasting intervals can be produced by simulation for all of the ETS models and by algebraic formulae for some of the models [8, ch. 7.7]. The intervals represent forecast distribution, which have medians considered as the point forecasts [8, ch. 7.7]. The equations for h-step forecasts involving time $T > t$ can be derived from the model equations from Table C.2 by iterating over $t = T+1, \ldots, T+h$ with $\varepsilon_t = 0$ [8, ch. 7.7].

Basic Structural Model (BSM) can be represented as components similarly to ETS. There is an equation for observation at time t

$$y_t = \ell_t + s_{1,t} + \varepsilon_t, \tag{21}$$

which is a sum of an error term $\varepsilon_t \sim \mathrm{NID}(0, \sigma_\varepsilon^2)$, a level at time t

$$\ell_t = \ell_{t-1} + b_{t-1} + \xi_t, \tag{22}$$

where $\xi_t \sim \mathrm{NID}(0, \sigma_\xi^2)$, and a seasonal component at time t

$$s_{1,t} = -\sum_{j=1}^{m-1} s_{j,t-1} + \eta_t, \tag{23}$$

where the error term $\eta_t \sim \text{NID}(0, \sigma_\eta^2)$ and $s_{j,t} = s_{j-1,t-1}$ and $j = 2, \ldots, m-1$, where m is the seasonal period [24, slide 10]. The level Equation (22) also uses a slope at time t

$$b_t = b_{t-1} + \zeta_t, \tag{24}$$

where the error term $\zeta_t \sim \text{NID}(0, \sigma_\zeta^2)$ [24, slide 10]. ETS models use the error term ε_t in their model equations (Table C.2), while a BSM model have different error processes ε_t, ξ_t, η_t and ζ_t in its equations.

3.1.4 Bagged ETS

Bagging stands for using a bootstrap aggregation [25]. It has been shown that a bagged ETS could give more accurate results than ETS [25].

As a first step, the time series is transformed by using a Box-Cox transformation with $0 \leq \lambda \leq 1$ [25]. After that, a non-seasonal time series is decomposed with loess method resulting in a trend part and a remainder part, while seasonal time series is decomposed with STL method resulting in a trend part, a remainder part and a seasonal part [25].

A moving block bootstrap (MBB) method processes the remainder component from the decomposition step [25]. The MBB outputs a bootstrapped series that is then added together with the decomposed parts other than the remainder [25]. Next, the summed series is back-transformed through inversion of the Box-Cox transformation [25]. This results in a group of time series [25].

For each time series from the group, a best ETS model according to AIC_c is found [25]. A median of the point forecasts from all of these ETS models form the point forecast of the bagged ETS model [25].

3.1.5 ARIMA and ARFIMA

There exist ARIMA(p, d, q) and ARIMA$(p, d, q)(P, D, Q)_m$: non-seasonal and seasonal ARIMA (AutoRegressive Integrated Moving Average) models [8, ch. 8.9]. ARIMA model is the result of combining autoregressive model AR(p), differencing operation and moving average model MA(q) [8, ch. 8.5].

The p in the autoregressive model with an intercept c

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t \tag{25}$$

shows how many lagged values of y_t are used to forecast the value of y_t [8, ch. 8.3]. The q in the moving average model with an intercept c

$$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} \tag{26}$$

shows how many past forecast errors are used to forecast the value of y_t [8, ch. 8.4]. In both AR(p) and MA(q) models, e_t stands for errors [8, ch. 8.4], which have no autocorrelation [8, ch. 2.9]. The two models together form

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t, \tag{27}$$

which is the equation of ARMA (AutoRegressive Moving Average) model for stationary time series, which is observed on time t and which has properties that are independent from t [8, ch 8.1]. For non-stationary time series, however, ARIMA models are used, which include d times of first differencing [8, ch 8.5] resulting in an equation with differenced series y'_t [8, ch. 8.5]:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t. \tag{28}$$

First differencing is done d times in order to make the time series stationary [8, ch. 8.1]. First difference is calculated by subtracting an observation's value at time $t-1$ from the observation's value at time t [8, ch. 8.1]. The first-order first difference equation is

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1-B)y_t, \tag{29}$$

where B is the backshift operator [8, ch. 8.1, ch. 8.2]. The second-order first difference equation is

$$y''_t = y'_t - y'_{t-1} = y_t - 2y_{t-1} + y_{t-2} = (1-B)^2 y_t, \tag{30}$$

which shows how the higher-order differences can be found when the time series stays non-stationary after the initial first differencing [8, ch. 8.1]. The non-seasonal ARIMA model uses p, d, q and $c = \mu(1 - \phi_1 - \cdots - \phi_p)$ with μ as the mean of y'_t in Equation (28) [8, ch. 8.5]. This equation can also be expressed as

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(1-B)^d y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q)e_t, \tag{31}$$

which is the same as

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(y'_t - \mu) = (1 + \theta_1 B + \cdots + \theta_q B^q)e_t \tag{32}$$

by using $y'_t = (1-B)^d y_t$ [8, ch. 8.5].

Seasonal models must also take into account $(P, D, Q)_m$ with $\Phi_1, \ldots, \Phi_P$ and $\Theta_1, \ldots, \Theta_Q$ in addition to (p, d, q) with $\phi_1, \ldots, \phi_p$ and $\theta_1, \ldots, \theta_q$ [8, ch. 8.9]. Parameter m marks the number of periods of a year [8, ch. 8.9]. Uppercase P and Q correspond to the orders of seasonal AR(P) and MA(Q) components [8, ch. 8.9] while D shows how many times the seasonal difference

$$y'_t = y_t - y_{t-m} \tag{33}$$

has been taken [8, ch. 8.1]. In seasonal ARIMA models' equations, corresponding non-seasonal and seasonal terms are multiplied together [8, ch. 8.9].

Hyndman and Khandakar (2008) algorithm can be used for ARIMA modelling [8, ch. 8.7]. In the algorithm, KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test can be used to determine the necessity and the degree of first differencing that makes the time series stationary [8, ch. 8.1]. OCSB (Osborn-Chui-Smith-Birchenhall) test can be used to determine the necessity of seasonal differencing that makes the time series stationary [9, `auto.arima` doc.]. The orders of autoregressive and moving average parts can be found through the minimisation of AICc using the stationary data [8, ch. 8.7]. The AICc, AIC and BIC for ARIMA models, given the data's likelihood L, the series' length T and parameters' count k, are defined as follows [8, ch. 8.6]:

$$\text{AIC} = -2\log(L) + 2(p+q+k+1), \tag{34}$$

$$\text{AICc} = \text{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2}, \tag{35}$$

$$\text{BIC} = \text{AIC} + [\log(T) - 2](p+q+k+1). \tag{36}$$

The missing parameters can be estimated by using the maximum likelihood estimation (MLE) [8, ch. 8.7].

Given an ARIMA model equation, the point forecasts equations for $h = 1, 2, 3, \ldots$ can be obtained by iterating over h after the following steps: bring y_t to the one side of the equality symbol and everything else on the other side, write $T + h$ instead of t and perform replacements described in Table 7 [8, ch. 8.8]. In order to avoid obtaining forecast intervals that might not be correct, the residuals must be "uncorrelated and normally distributed" [8, ch. 8.8].

Table 7: Replacements after t has been replaced with $T + h$ [8, ch. 8.8]

Equation Component	Replacement
Future observation	Future observation's forecast
Future error	Zero
Past error	Past error's residual

ARFIMA (AutoRegressive Fractionally Integrated Moving Average) models also have parameters p, d, q like ARIMA models [9, `arfima` doc.]. Hyndman and Khandakar (2008) algorithm can also be used for determining the autoregressive order p and the moving average order q [9, `arfima` doc.]. The first differencing degree d that can be a non-integer for ARFIMA models, could be estimated with

Haslett-Raftery (1989) algorithm [9, `arfima` doc.].

3.1.6 Feed-Forward Neural Network

A feed-forward neural network always contains an input layer and an output layer [8, ch. 11.3]. It might also contain hidden layers between the input and the output layer [8, ch. 11.3]. One layer's output is subsequent layer's input [8, ch. 11.3]. Figure 2 is an example of a network with six input nodes, four hidden layer nodes and two output nodes.

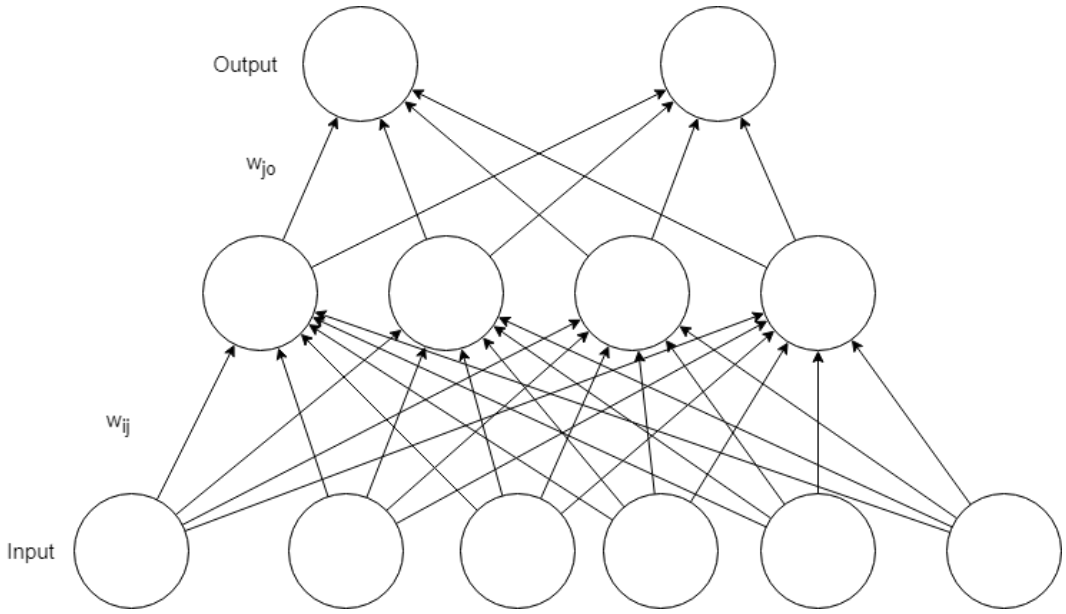

Figure 2: Three Layers of Neurons

For non-seasonal time series data, a feed-forward network that uses p lagged values $y_{t-1}, \ldots, y_{t-p}$ as inputs to forecast y_t can be expressed as NNAR(p, k), where k is the count of nodes in the hidden layer [8, ch. 11.3]. Similarly, a neural network for seasonal time series data can be expressed as NNAR$(p, P, k)_m$, where m is the number of periods and P regulates how many seasonal input values are used – there are input values $y_{t-m}, \ldots, y_{t-Pm}$ in addition to $y_{t-1}, \ldots, y_{t-p}$ [8, ch. 11.3]. The non-seasonal inputs and seasonal inputs may sometimes overlap. Once it happens, then the same input will not be used twice. When considering a non-seasonal feed-forward neural network NNAR(p, k) as a function f, then

$$y_t = f((y_{t-1}, \ldots, y_{t-p})') + \varepsilon_t, \tag{37}$$

is the representation of the model with homoscedastic errors $\{\varepsilon_t\}$ [8, ch. 11.3].

Every jth non-input node finds a weighted linear combination of its n inputs $x_i \dots x_n$, adds a bias constant b_j [27, ch. 8.10] and then changes the results with a nonlinear function to produce the output value of the node [8, ch. 11.3]. The weights' $w_{i,j}$ values between nodes i and j are random values improved by processing the input data during the training period and bounded by a decay parameter [8, ch. 11.3]. Also the bias constants are modified during model training [8, ch. 11.3]. Each neural network training iteration starts with random weight values [9, `nnetar` doc.]. It is a common practice to average the results of multiple training iterations [8, ch. 11.3]. The forecast intervals of the models are found through simulation which uses bootstrapped residuals or random selection of errors from normal distribution [8, ch. 11.3].

3.1.7 BATS and TBATS

BATS($\omega, \{p, q\}, \phi, m_1, \dots, m_T$) and TBATS($\omega, \{p, q\}, \phi, \{m_1, k_1\}, \dots \{m_T, k_T\}$) frameworks can be used for forecasting values of time series with complex seasonality such as series with non-integer period or series with multiple seasonal patterns [26]. Both use α and β as smoothing parameters and $m_1, \dots, m_T$ as seasonal periods [26]. TBATS's k_i paired with m_i shows how many pairs of Fourier-like terms (harmonics) were selected for ith type of seasonality [26]. BATS uses smoothing parameter γ_i (where i marks the seasonal pattern from 1 up to T), while TBATS needs smoothing parameters $\gamma_1^{(i)}$ and $\gamma_2^{(i)}$ [26].

The Box-Cox parameter ω is used in

$$y_t^{(\omega)} = \begin{cases} \log y_t & \text{if } \omega = 0; \\ \frac{y_t^{\omega} - 1}{\omega} & \text{otherwise} \end{cases}, \tag{38}$$

to perform a Box-Cox transformation of observation y_t that results in $y_t^{(\omega)}$ [26]. The transformed observation can be calculated in equation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^{T} s_{t-m_i}^{(i)} + d_t, \tag{39}$$

with the damping parameter ϕ for achieving damped trend [26]. The equation uses a local level ℓ_t

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t \tag{40}$$

and a local trend b_t

$$b_t = (1 - \phi) b + \phi b_{t-1} + \beta d_t \tag{41}$$

that is calculated by using a global trend b and an ARMA(p, q) process d_t

$$d_t = \sum_{i=1}^{p} \phi_i d_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t, \tag{42}$$

which incorporates the error term $\varepsilon_t \sim \text{NID}(0, \sigma^2)$ [26].

BATS and TBATS are different in terms of the ith seasonal component. BATS's one at time t is defined as

$$s_t^{(i)} = s_{t-m_i}^{(i)} + \gamma_i d_i, \tag{43}$$

while TBATS's one comes in multiple parts starting with

$$s_t^{(i)} = \sum_{j=1}^{k_j} s_{j,t}^{(i)} \tag{44}$$

that's sub-component $s_{j,t}^{(i)}$ is

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \tag{45}$$

and which also needs $s_{j,t}^{*(i)}$ equation

$$s_{j,t}^{*(i)} = -s_{j,t-1} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t, \tag{46}$$

where $\lambda_j^{(i)} = \frac{2\pi j}{m_i}$, completing the trigonometric approach that gives the letter "T" into the name TBATS [26].

3.2 Data Set Selection

The forecasting model for tendencies in cybersecurity needs to be based on some data set containing the discovery time and severity of software vulnerabilities. This subchapter discusses some of the available options. Finally, an explained choice is made.

In 2016, Janulevičius found six existing vulnerability data sets: National Vulnerability Database (NVD), the Exploit Database (EDB), exploit kits database (EKITS), Symantec Security Response threat write-ups, 0day.today and Rapid7 Vulnerability and Exploit Database [28]. The last two data sets, 0day.today and Rapid7 Vulnerability & Exploit Database, were discarded and not analysed by the researcher [28]. He found 0day.today to have a lack of validity while Rapid7 database was discovered to be replicating entries from NVD [28]. The size of EDB, EKITS and Symantec's write-ups were significantly smaller than NVD: 35220, 216 and 1125 entries compared to 74100 entries [28]. Furthermore, the analysis was concluded with an understanding that NVD was a superset of all the information available in the other data sets. Therefore, NVD was chosen to be the main source of vulnerability scores for that doctoral dissertation [28].

There exists Vulnerability Database Catalog based on the work done by Vulnerability Reporting and Data eXchange Special Interest Group (VRDX-SIG) members [29]. The SIGs have tried to select freely accessible, public and multi-vendor-coverage vulnerability databases (VDBs). For each VDB, the following information is provided [29]:

- overview (VDB name, maintainer, URL, description),
- ID scheme (number of ID schemes, ID format),
- CVE (use of CVE),
- CWE (use of CWE IDs, use of all CWE IDs or a subset),
- CVSS (use of CVSS base, temporal and environmental metrics, use of version 3 or version 2 of CVSS),
- CPE (use of CPE),
- XML Data Feed (usage of CVRF language, use of RSS/Atom),
- VDB contents (title, description, products affected, impact, severity, solution, vendor information, references, credit/finder, available languages and search feature).

Table 8 is a comparison table, containing a subset of the available information: VDB name, maintainer, whether the data is available in English and whether CVE, CWE and CVSS are used. Common names, types and severity scores of vulnerabilities in the data would give some standards to rely on during the creation of the forecasting model and would help later to put the forecasting results into an understandable context. There are four VDBs out of 22 that are available in English and use CVE, CWE, CVSS. These four are JVN iPedia, National Vulnerability Database (NVD), ICS-CERT Advisory and CERT/CC Vulnerability Notes Database. However, ICS-CERT Advisory uses different CVSS versions for new and old vulnerabilities which makes the data inconsistent for long-term CVSS analysis. CERT/CC Vulnerability Notes Database's website refers to NVD as a "more comprehensive coverage of public vulnerability reports" [30].

That leaves JVN iPedia and National Vulnerability Database (NVD) as the potential candidates for the main data set of the thesis. Both provide vulnerabilities' data, which are categorised by a subset of CWE IDs and assessed by the version 2 of CVSS severity base metrics (Table 8). NVD uses the same ID scheme as the Common Vulnerabilities and Exposures (CVE) list by MITRE [29], the source from where NVD fetches data for further analysis. For each CVE entry, there exists one NVD entry with the same ID. On the other hand, JVN iPedia

has its own ID scheme and for each of its entry, there might exist multiple corresponding entries in Common Vulnerabilities and Exposures (CVE) list by MITRE (e.g. CVE-2017-10838 and CVE-2017-10839 for JVNDB-2017-000207 [6]), which could needlessly complicate the vulnerability discovery analysis and modelling. As a result of looking into Janulevičius's work and Vulnerability Database Catalog's data, National Vulnerability Database (NVD) is selected to be the main data set of the thesis and the basis for the forecasting model for tendencies in cybersecurity. Consequently, the research question BQ2 from Table 6 is answered.

Table 8: VDB Comparison Using Data from [29]

VDB Name	Maintainer	CVE	CWE	CVSS	English Available
JVN iPedia	IPA	Yes	Yes (subset)	Yes (v2 of base)	Yes
National Vulnerability Database (NVD)	NIST	Yes	Yes (subset)	Yes (v2 of base)	Yes
ICS-CERT Advisory	ICS-CERT	Yes	Yes (all)	Yes (v3 for new and v2 for old vulnerabilities, base, some temporal)	Yes
CERT/CC Vulnerability Notes Database	CERT/CC	Yes	Yes (all)	Yes (v2 of base, temporal, environmental)	Yes
CERT-EU Security Advisories	CERT-EU	Yes	No	Yes (v2 of base)	Yes
Japan Vulnerability Notes (JVN)	JPCERT/CC	Yes	No	Yes (v2 of base)	Yes
TippingPoint Zero Day Initiative	TippingPoint	Yes	No	Yes (v2 of base)	Yes
China National Vulnerability Database (CNVD)	CNCERT/CC	Yes	No	Yes (v2 of base)	No
scip VulDB	scip AG	Yes	No	Yes (v2 of base, temporal)	Yes
AusCERT Security Bulletins	AusCERT	Yes	No	No	Yes
Common Vulnerabilities and Exposures (CVE)	MITRE	Yes	No	No	Yes
Exploit Database	Offensive Security	Yes	No	No	Yes
JC3 Bulletin Archive	Department of Energy	Yes	No	No	Yes

[6] http://jvndb.jvn.jp/en/contents/2017/JVNDB-2017-000207.html

Table 8: VDB Comparison Continued with Data from [29]

VDB Name	Maintainer	CVE	CWE	CVSS	English Available
NCSC-FI Vulnerability Database	Finnish Communications Regulatory Authority (FICORA) - National Cyber Security Centre Finland (NCSC-FI)	Yes	No	No	Yes
Packet Storm	Packet Storm	Yes	No	No	Yes
SecuriTeam	Beyond Security	Yes	No	No	Yes
Vulnerabilities	Security Focus	Yes	No	No	Yes
SecurityTracker	SecurityGlobal.net LLC	Yes	No	No	Yes
Verisign Vulnerability Reports	Verisign	Yes	No	No	Yes
Vulnerability & Exploit Database	Rapid7	No	No	No	Yes
China National Vulnerability Database of Information Security (CNNVD)	China Information Security Evaluation Center	No	No	No	No
WooYun.org	WooYun.org	No	No	No	No

National Vulnerability Database (NVD), maintained by National Institute of Standards and Technology (NIST) of U.S. Department of Commerce and sponsored by Department of Homeland Security's National Cybersecurity and Communications Integration Center's United States Computer Emergency Readiness Team (DHS/NCCIC/US-CERT), is synchronised with Common Vulnerabilities and Exposures (CVE) list of software vulnerability identifiers [2]. NVD adds additional information, including CVSS severity scores and CWE vulnerability types, to the data of CVE [2]. Title 17 of the United States Code makes NVD data feeds available for public use [10]. The data can be downloaded either in JSON or XML format. Since JSON vulnerability feeds are a BETA release with a probably changing format as of October 2017 [10], XML vulnerability feeds are used instead.

4 Contribution and Evaluation

Table 9 introduces the research questions of the evaluation. EQ1 investigates what needs to be done in order to prepare the data for modelling. After the models have generated the forecasts, EQ2 focuses on finding the most accurate ones. EQ3 goes into checking the forecast intervals as they might be incorrect when certain assumptions are not met. When the most accurate model types have been discovered, then they are used to forecast the unseen future as part of answering EQ4.

Table 9: Research Questions for the Contribution

Identifier	Question
EQ1	How to process the data from NVD?
EQ2	Which models' point forecasts are the most accurate?
EQ3	Which forecast intervals can be taken seriously?
EQ4	What are the forecasts for the unseen future?

4.1 Data

NVD XML format has an `<nvd>` tag containing vulnerabilities as `<entry>` tags. An example entry is shown as a tree in Figure 3. Among other data, the `<entry>` contains CVE-ID assigned by CVE, time when the vulnerability was published, CWE type and CVSS version 2 base score with its vectors that were used to calculate the score.

Data preprocessing steps were written as R code inside R package 'nvdr'[7]. It was necessary to extract the subset of the data relevant to the forecasting model. From a tree of a vulnerability's `<entry>` element, the value of the `id` attribute of `<entry>`, the value of `<vuln:published-datetime>`, the first seven values of the children of `<cvss:base_metrics>` and the values of `id` attributes of all `<cwe>` elements were extracted. In addition, the value of `<vuln:summary>` was checked for substring "** REJECT **" to determine whether the vulnerability entry had been rejected (Table 10 [8]).

Many of the vulnerability entries have only one corresponding CWE but there also exist entries with multiple CWE categories (Figure 3) and entries without the `<cwe>` elements. The potential absence is logical as NVD only uses a subset of CWE categories. A difference between NVD's JSON and XML data sets were found. While JSON files contained a notice "NVD-CWE-Other" in the place of the vulnerability category when the category was out of the NVD's CWE subset,

[7] `https://github.com/realerikrani/nvdr`

[8] `https://github.com/realerikrani/nvdr/blob/master/man/nvd.Rd`

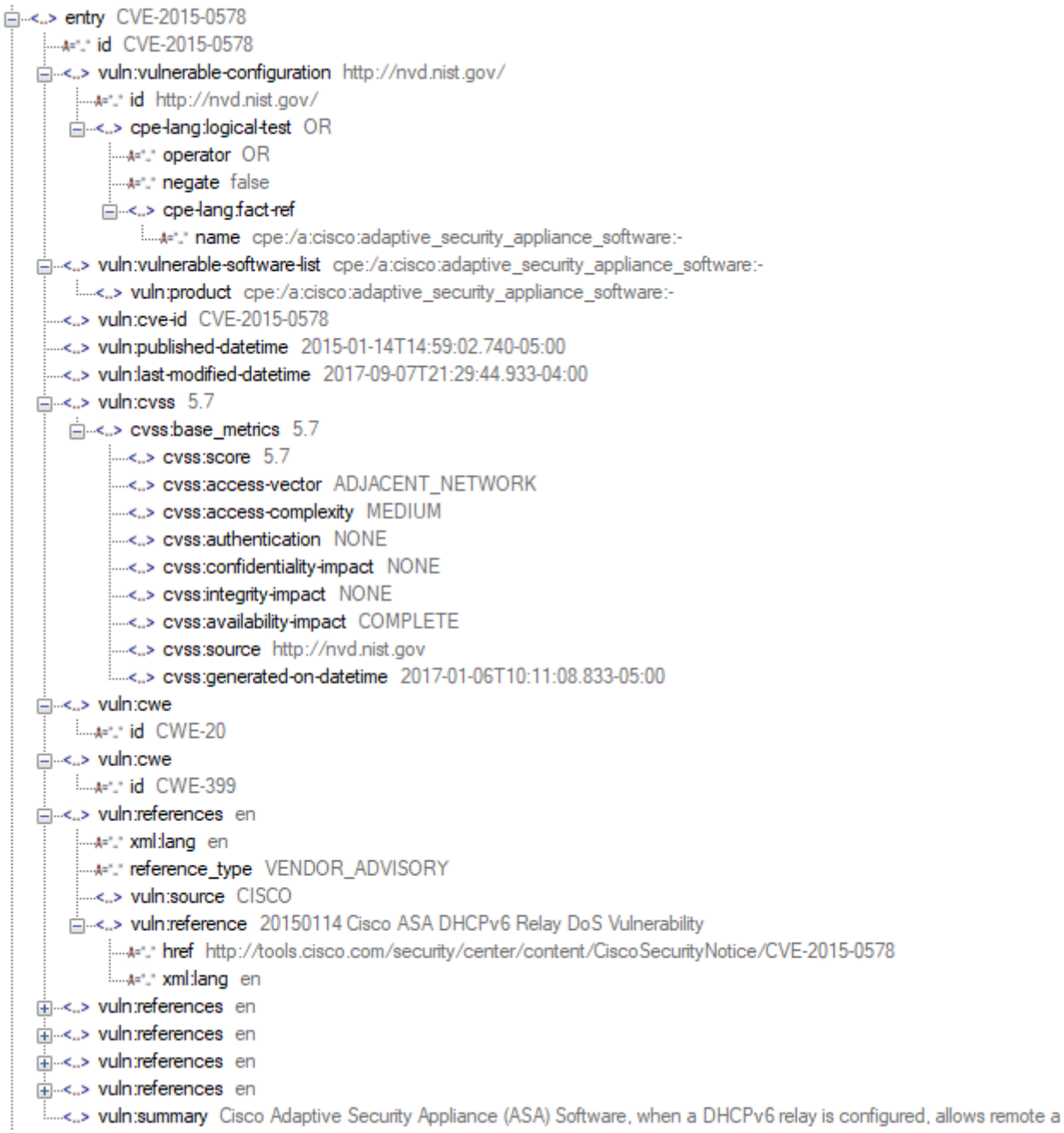

Figure 3: A tree of an `<entry>` tag about CVE-2015-0578

XML files simply did not include the `<cwe>` element as a child of `<entry>` element. Missing data was marked as `NA` in the processed data set.

Entries from CVE-2011 to CVE-2016 as of 7 October 2017 were processed and saved in a binary format in the package 'nvdr', making it accessible to all users of the package. The data has 11 variables and 41190 rows. The data's overview is presented in Table 10[9]. For those, who want to perform the preprocessing on certain NVD's XML Version 2 file on their own, 'nvdr' provides functionality that allows to do that. This functionality was used in January 2018 to obtain up-to-date data covering entries from CVE-2011 to CVE-2017.

Table 10: Overview of the Processed Data

Data Field	Description
cve_id	Common Vulnerabilities and Exposures ID
cve_rejected	Vulnerability's rejection indicator
published	Vulnerability's publication date from `<vuln:published-datetime>`
cwe	Common Weakness Enumeration category
cvss_score	Base Score by Common Vulnerability Scoring System version 2.0
cvss_av	Base Score's Access Vector (AV)
cvss_ac	Base Score's Access Complexity (AC)
cvss_au	Base Score's Authentication (Au)
cvss_c	Base Score's Confidentiality Impact (C)
cvss_i	Base Score's Integrity Impact (I)
cvss_a	Base Score's Availability Impact (A)

Executing `na.omit(unique(nvdr::nvd[, "cwe"]))` in R console, gives 224 unique CWE categories corresponding to the CVE entries in the (CVE-2011 to CVE-2016) data set. However, some of them are groups of categories, for example `CWE-264|CWE-287`. The groups are separated during the further processing. This means that a sample entry `CWE-AAA|CWE-BBB` published on `20XX-XX-XX` becomes two entries: `CWE-AAA` published on `20XX-XX-XX` and `CWE-BBB` published on `20XX-XX-XX`. All entries containing missing values in any of the fields `published`, `cwe` and `cvss_score` are omitted. The data field `published` is used as a publication time of the vulnerability. Its year and the year's part of a CVE ID (`20XX`) don't always match. Research question EQ1 from Table 9 is answered.

[9] `https://github.com/realerikrani/nvdr/blob/master/man/nvd.Rd`

4.2 Measures

The forecasting accuracy is measured by calculating mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE) and mean absolute scaled error (MASE) [8, ch. 3.4][31, topic 3]:

$$\mathrm{MAE} = \frac{1}{T}\sum_{t=1}^{T}\left|y_t - \hat{y}_{t|t-1}\right|, \tag{47}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1})^2}, \tag{48}$$

$$\mathrm{MAPE} = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{100(y_t - \hat{y}_{t|t-1})}{y_t}\right|, \tag{49}$$

$$\mathrm{MASE} = \frac{1}{T}\sum_{t=1}^{T}\frac{\left|y_t - \hat{y}_{t|t-1}\right|}{Q}, \tag{50}$$

where the following Q calculation uses only the training data ("training MAE"):

$$Q = \begin{cases} \frac{1}{T-1}\sum_{t=2}^{T}|y_t - y_{t-1}| & \text{if } \{y_t\} \text{ is non-seasonal;} \\ \frac{1}{T-m}\sum_{t=m+1}^{T}|y_t - y_{t-m}| & \text{if } \{y_t\} \text{ is seasonal.} \end{cases} \tag{51}$$

In the thesis, Breusch-Godfrey test for time series linear models and Ljung-Box test for multiple models with a significance level of $\alpha = 0.05$ are used to check whether it is possible to reject the null hypothesis that the residuals are independently distributed – it is rejected when the p-value ≤ 0.05 [8, ch. 8.7]. For ARFIMA, bagged ETS, BSM and neural network autoregression models, autocorrelation function (ACF) plots for residuals [8, ch. 2.8] with $\pm 2/\sqrt{\text{"time series length"}}$ significance bounds are used to check the hypothesis that the residuals are independently distributed (they have no autocorrelation) – lag spikes on ACF plot must stay within significance bounds in order to not reject the hypothesis [8, ch. 2.9]. The ACF plots are generated with 'forecast' package's [9] function `checkresiduals`. The Breusch-Godfrey and Ljung-Box tests are carried out in 'nvdr' using the same approach as `checkresiduals` uses[10].

[10] The exact way of lag calculations from file `https://github.com/robjhyndman/forecast/blob/c87f33/R/checkresiduals.R` are used in nvdr.

Shapiro-Wilk test is used with a significance level of $\alpha = 0.05$ to check whether is it possible to reject the null hypothesis that the residuals are normally distributed – it is rejected when the p-value ≤ 0.05. Bootstrapped forecast intervals are generated in the cases, where ARIMA models, benchmark models or ETS models have uncorrelated but not normally distributed residuals. This means using samplings from past residuals for future simulations [8, ch. 3.5]. In the cases of feed-forward neural network autoregression models, the need for bootstrapped intervals must be determined by manually executing the required functions. When some model, which uses bootstrapped residuals for generating the forecast intervals, has chosen by 'nvdr' for some specific CWE and then the selected type of the chosen model is used to forecast untestable future of the specific CWE, then the new forecast intervals are also based on bootstrapped residuals.

The residuals' zero mean is checked by manually executing the necessary functions by the thesis's author.

4.3 Procedure

At first a monthly time series from January 2011 to December 2015 is used as a training set and a monthly time series from January 2016 to December 2016 is used as a test set. Later, the data processing steps described in this paragraph are performed again starting with a monthly time series from January 2011 to December 2016 as a training set and monthly time series from January 2017 to December 2017 as a test set. With package 'nvdr', it is possible to generate forecasts for all available CWEs or focus on a set of CWEs that meet certain criteria. In the thesis, the training and the test set (merged together) are used to find CWEs which have occurred at least 100 times in any of the years during the time period. What is more, each CWE's yearly mean CVSS scores are summed and divided by the number of years of the period, resulting in a mean yearly mean CVSS scores, which enables to identify critical CWEs with the minimum mean yearly mean CVSS score above 4.0. Thirdly, for all CWEs that have been published at least 200 times during the time period, a mean absolute percent change $\frac{1}{\text{training\_set\_years\_count}} \sum_{i=\text{training\_set\_start\_year}}^{\text{training\_set\_end\_year}} \left| \frac{\text{test\_year\_count} - \text{year}_i\text{\_count}}{\text{year}_i\text{\_count}} \right|$ of vulnerability publishing counts is calculated and the CWEs above the median of the obtained numbers for all CWEs are considered as the most changing CWEs. If year_i's count is 0, then the absolute percent change value in the mean absolute percent change equation is set to 0. By combining these three approaches, a subset of all CWEs are chosen for forecasting testable 2016 and initially untestable 2017. Afterwards, this technique is used again for finding a subset of CWEs, which monthly mean CVSS scores for untestable 2018 are forecasted.

For each selected CWE entry, nearly all forecasting models from Chapter 3.1

are used to generate point forecasts alongside 80% and 95% forecast intervals. Only bagged ETS models produce point forecasts with intervals which minimum and maximum values are simply the minimum and maximum values of the used ensembles' point forecasts [9, `forecast.baggedModel` doc.]. In each case of benchmark model selection, forecasts are generated with and without Box-Cox transformation and the more accurate forecasts of the two, according to which ones gets the minimum numbers for the most of MAE, RMSE, MAPE and MASE scores, are selected. If Box-Cox transformation is used with benchmark models, then the λ for the transformation is determined by the 'forecast' package's [9] function `BoxCox.lambda` with lower limit -1 and upper limit 2.

The data are also transformed with Box-Cox transformation using the output from `BoxCox.lambda` as λ before the estimation of ARFIMA, ARIMA, ETS, time series linear regression and feed-forward neural network models. In order to speed up the process, the package 'nvdr' always uses the transformation for these models and does not compare the results to the same models using data that is not transformed. When a time series does not need a Box-Cox transformation, then the function `BoxCox.lambda` should return 1 in ideal case, which means the time series's shape would remain the same [8, ch. 3.2]. For TBATS models, the 'forecast' package's [9] function `tbats` fits the TBATS models with and without Box-Cox transformation and makes a selection between those options by comparing the AIC value.

The inverse Box-Cox transformation is used on the outcomes obtained with the transformed data, resulting in backtransformed forecast intervals and point forecasts representing the forecast densities' median values. Benchmark, ARFIMA, ARIMA, ETS, TBATS and time series linear regression models are set to bias-adjust the values and output the mean forecast instead of the median forecast.

The MAE, RMSE, MAPE and MASE of the mean, the naïve, the seasonal naïve and the drift forecasts are compared for each CWE. The one that gets the minimum number for the most of those accuracy measures for a given CWE is chosen as the benchmark method for that CWE. Similarly, the MAE, RMSE, MAPE and MASE of each benchmark model and all the other models' accuracy measures are compared for each CWE. The method that gets the minimum number for the most of those accuracy measures is chosen as the best model for a specific CWE.

The previous training set and test set are merged into a new training set to forecast the mean CVSS scores of the next 12 months of 2017 by using the types of models for each CWE that previously showed the best accuracy measures for 2016. At the start of 2018, the latest NVD data for CVE-2011 to CVE-2017 is downloaded and the accuracy of the 2017 forecasts is measured with the latest data's subset from January 2017 to December 2017. The entire new data from January

2011 to December 2017 is finally used to generate forecasts for 2018. When a discovered best model is one of the benchmark models and Box-Cox transformation was applied when finding the best model type, then the transformation is also applied when using the model type later (with the new training set) to forecast the untestable future.

Table 11 summarises the main steps. It also provides further details about the most important input and output of the activities. The table does not explain the steps how the CWEs are selected or how the best model types are chosen.

Table 11: Steps (initially X = 2016 and Y = 2017)

Input	Activity	Output
`https://nvd.nist.gov/vuln/data-feeds#CVE_FEED`	1. Obtain data	CVE-2011…CVE-X XML 2.0 files
CVE-2011…CVE-X XML 2.0 files	2. Process data	Time series between the start of 2011 and the end of X
Time series between the start of 2011 and the end of X	3. Forecast testable X	Training set, test set, residuals, point forecasts for X and forecast intervals for X
Test set, point forecasts for X	4. Measure forecast accuracy for X	MAE, RMSE, MAPE, MASE
MAE, RMSE, MAPE, MASE	5. Select most accurate model types	Most accurate model types for each selected CWE
Residuals from forecasting models, model types, forecast intervals for X	6. Analyse residuals (if $X \neq 2017$)	Knowledge whether forecast intervals for X can be taken seriously
Most accurate model types for each selected CWE	7. Forecast untestable Y	Training set, residuals and point forecasts for Y and forecast intervals for Y
Residuals from forecasting models, model types, forecast intervals for Y	8. Analyse residuals (if $Y = 2018$)	Knowledge whether forecast intervals for Y can be taken seriously
`https://nvd.nist.gov/vuln/data-feeds#CVE_FEED`	9. Obtain new data when it becomes available	CVE-2011…CVE-Y XML 2.0 files
CVE-2011…CVE-Y XML 2.0 files	10. Process new data	Time series between the start of 2011 and the end of Y
Time series between the start of Y and the end of Y as test set, point forecasts for Y	11. Measure forecast accuracy for Y	MAE, RMSE, MAPE, MASE
Time series between the start of 2011 and the end of Y	12. Repeat steps $3-8$ with $X = X+1$ and $Y = Y+1$	Analysed point forecasts and forecast intervals for the future

4.4 R Package 'nvdr'

The R package 'nvdr' is available to download via its GitHub repository[11]. The installation instructions are in "README.md" file and in a user guide. The documentation, the user guide and the performance measures are available via the package's GitHub Pages website[12]. When a commit is made to the master branch of the repository, then Travis CI[13] tool is set to automatically run `R CMD check`, which checks the package for multiple problems[14]. During the development, the 'lintr'[15] package was used to perform static code analysis. One notification from that analysis was ignored: non-lowercase method names were used in the code. It was ignored because an object-oriented approach was used and camel case convention seemed appropriate for naming classes and methods.

The package's name, 'nvdr', comes from the fact that it functions with the data from NVD and it is written in R language[16]. As it was stated in the introduction, the thesis's forecasting results are obtained by applying the package's functions on a selected subset of data from NVD. It is common knowledge that R language is suitable for data science and statistical computing. Furthermore, there exists the package 'forecast' [9], which provides multiple functions for time series forecasting. Therefore, it was reasonable decision to write 'nvdr' in R language and take use of its packages provided by its community.

Object oriented programming (OOP) in R is possible by using S3, S4, or R6 systems [32, ch. 11]. S3 and S4 implement functional OOP, but "R6 implements encapsulated OOP" [32, ch. 11]. Therefore, it was decided to use R6 as it is more similar to OOP in Java and Python [32, ch. 15]. Class `CWE` is defined for extracting and processing the data from XML files. An object of class `CVSSForecaster` can afterwards use the object of class `CWE` from which it extracts the necessary time series data. The object of class `CVSSForecaster` can then build and store objects of class `NVDModel`. Class `NVDModel` has two subclasses: `FcastModel` and `FitModel`. These two subclasses have subclasses of their own. `FcastModel` represents models which are created and used for forecasting in one step. `FitModel` represents models which are fitted (one separate step) and then used for forecasting (step two). The class hierarchy helps to avoid code duplication.

The classes at the bottom of the hierarchy provide specialised methods that call the forecasting functions mostly from 'forecast' package [9]. The performance of 'nvdr' is affected by the performance of these methods. The package's GitHub

[11] `https://github.com/realerikrani/nvdr`

[12] `https://realerikrani.github.io/nvdr/`

[13] `https://travis-ci.org/`

[14] Hadley Wickham has made an overview of the `R CMD check` in `http://r-pkgs.had.co.nz/check.html`.

[15] `https://cran.r-project.org/web/packages/lintr/index.html`

[16] `https://www.r-project.org/`

Pages website has a page about performance measures[17]. The data that comes with 'nvdr' was used and forecast models were created for 25 CWEs. Execution of `setBaggedETS` took 24.5 minutes, while `setTSLinear` finished in 0.9 seconds[18]. Data extraction from seven XML files with `setBaseData` lasted 2.3 minutes.

4.5 Results and Discussion

Forecasts for 2016, 2017 and 2018 are presented and analysed in Chapters 4.5.1, 4.5.2 and 4.5.3. The forecasting activities performed for 2016 and 2017 provide answers to research question EQ2 from Table 9: discovering the most accurate models for selected CWEs. Whenever the forecast intervals are analysed for a given case, then the research question EQ3 (Table 9) is being answered. The answer to question EQ4 (Table 9) is given in Chapter 4.5.3: providing the vulnerability category forecasts for 2018, a time period that has not ended yet at the time of writing the thesis.

4.5.1 Forecasting 2016

Given the training set from January 2011 to December 2015 and the test set from January 2016 to December 2016 as of October 2017, the forecasts were generated for 25 CWEs. The best models based on the forecast accuracy measures are presented in Listing 4.1 and all the corresponding forecasts are given as plots together in Figures D.4, D.5 and D.6 and some separately in this chapter. These 25 CWEs were discovered by using the three CWE selection techniques from Chapter 4.3. Table D.1 helps to understand the meanings of the CWEs mentioned in this chapter.

[17] `https://realerikrani.github.io/nvdr/articles/performance.html`

[18] The testing computer had 15.9 GB RAM and "Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40GHz" as mentioned on the website.

Listing 4.1: Best Forecast Accuracy Results for 2016

	cwe	method	MAE	RMSE	MAPE	MASE
1:	CWE–119	ETS(A,N,N)	0.4952756	0.6236950	6.453029	0.6951237
2:	CWE–79	NNAR(12,1,6)[12]	0.1449825	0.1762762	3.538222	1.0873687
3:	CWE–264	Random walk with drift	0.3819209	0.4929276	5.270236	0.6256725
4:	CWE–20	Seasonal naive method	0.5628902	0.7139288	9.533621	0.9066688
5:	CWE–200	Naive method	0.1567467	0.1831451	3.741093	0.3271235
6:	CWE–310	NNAR(13,1,7)[12]	0.7243963	0.9515430	14.088904	0.6535907
7:	CWE–399	TBATS(1, {0,0}, -, {<12,2>})	0.7528506	1.0626528	15.981409	0.8835411
8:	CWE–89	Seasonal naive method	0.5000000	0.7325754	6.750909	1.9834711
9:	CWE–352	Seasonal naive method	0.2666667	0.4490731	4.274751	1.2190476
10:	CWE–22	Mean	1.0451772	1.8584948	Inf	1.3030781
11:	CWE–189	ARIMA(0,1,3)	2.7641111	3.6154603	Inf	3.0153939
12:	CWE–94	BATS(1, {0,0}, -, -)	3.1922883	4.4188081	Inf	2.3646580
13:	CWE–284	Linear regression model	0.7925139	0.9481393	14.179140	0.2815741
14:	CWE–287	Naive method	0.6436357	0.8778816	10.281005	0.6865448
15:	CWE–255	baggedModel	1.3180593	1.5340520	24.850372	0.5741093
16:	CWE–254	NNAR(2,1,2)[12]	0.4383507	0.5038926	8.340568	0.3222180
17:	CWE–17	Mean	2.3294444	2.5692173	Inf	1.2971384
18:	CWE–416	Basic structural model	3.8615978	3.9296579	Inf	0.9385149
19:	CWE–78	Basic structural model	1.9506621	3.4913696	Inf	0.5066655
20:	CWE–134	Basic structural model	1.8200003	2.9281476	Inf	0.4711975
21:	CWE–190	Linear regression model	4.2062687	4.6897652	Inf	2.5018698
22:	CWE–77	Seasonal naive method	2.2821732	3.5904477	Inf	0.8538138
23:	CWE–362	NNAR(1,1,2)[12]	1.4068894	1.5870417	26.570987	0.6537337
24:	CWE–59	ETS(A,N,N)	3.2467272	3.3642079	Inf	1.3229449
25:	CWE–19	Naive method	0.9166667	1.1365151	15.287480	0.4767064
	cwe	method	MAE	RMSE	MAPE	MASE

There were 18 CWEs that occurred at least 100 times in any of the years during the time period. The total sums of the 18 weaknesses types over the time period are shown in Figure D.1. CWE-119 was counted 4988 times which was the biggest total count followed by CWE-79 with 3874 occurrences and CWE-264 with 3348 occurrences. Eight CWEs out of 18 had a total count above 1000. Two CWEs, CWE-17 and CWE-416, had a total count less than 200. The first one, CWE-17, was published 127 times in 2015. The second one, CWE-416, was published on 110 occasions in 2016.

There were 23 CWEs (Figure D.2) with mean yearly mean CVSS scores above 4.0. Weakness type CWE-78 had the value 8.6 as the highest score of the results. Out of 23 CWEs, 14 CWEs had mean yearly mean CVSS score above 6.0.

The mean absolute percent change of vulnerability publishing counts were calculated and the scores of eight CWEs were found to be above the median of the entire set of the scores (Figure D.3). The value 52.77 for CWE-284 is significantly higher than the rest of the values, which stay below 5.0. The notably higher number was obtained by calculating

$$\frac{1}{5}\left(0+\left|\frac{420-4}{4}\right|+\left|\frac{420-3}{3}\right|+\left|\frac{420-21}{21}\right|+\left|\frac{420-147}{147}\right|\right)\approx 52.77, \qquad (52)$$

where 420 was the count of 2016, 147 the count of 2015, 21 the count of 2014,

3 the count of 2013, 4 the count of 2012 and 0 the count of 2011. There was a significant difference between 2016 and each of the previous years.

The Venn diagram in Figure 4 shows that the intersection of all the three sets results in seven CWEs. The thesis at this stage, however, uses the union of the sets, which means selecting 25 CWEs' for forecasting their monthly mean CVSS scores in 2016.

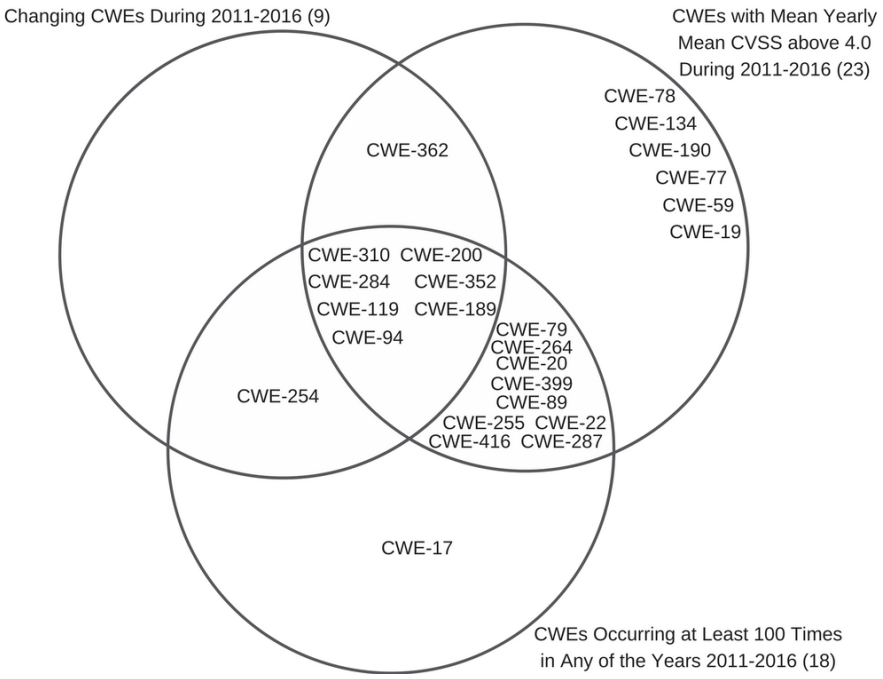

Figure 4: Venn Diagram Showing Which CWEs are Shared by the Groups

In addition to forecast accuracy results in Listing 4.1, the Figures D.4, D.5 and D.6 show the test set values as the red line that can be compared with the blue line representing the point forecasts. On 10 occasions, MAPE is `Inf` (Listing 4.1) because of the test sets that contain observations which values are 0. The rest of MAPE values for other CWEs' CVSS score forecasts are between 3.54% and 26.57%. MAE values are between 0.14 and 4.21, RMSE between 0.18 and 4.69 and MASE between 0.28 and 3.02.

ETS(A,N,N) model, simple exponential smoothing with additive errors, was chosen by 'nvdr' with parameters obtained by 'forecast' package [9] using maximum likelihood estimation for **CWE-119**. It was selected by 'forecast' package's [9] `ets` function from possible ETS models by minimising the AIC_c. It has an additive error component, no trend component and no seasonal component. Its equations

from Table C.2 [8, ch. 7.5, Table 7.8] are

$$y_t = \ell_{t-1} + \varepsilon_t \tag{53}$$

for the measurement and

$$\ell_t = \ell_{t-1} + \alpha\varepsilon_t \tag{54}$$

for the state. The estimate of the initial level component is $\ell_0 = 1.074$. The smoothing parameter $0 \leq \alpha = 0.2138 \leq 1$ is closer to zero than to one, showing that not much weight is put on the most recent time series observations to perform an exponentially weighted forecasting. It causes the exponentially decreasing weights to decay more slowly on a time-scale directed to past than with an α that is closer to one. The low value of α causes smoother changes in the level than an α closer to one would cause. The used Box-Cox transformation parameter estimate obtained with function `BoxCox.lambda` [9] is $\lambda = -0.739$. This should stabilise the variance of the series similarly to an inverse transformation because λ is close to -1.

When 'nvdr' calls the 'forecast' package's [9] `ets` function with a Box-Cox transformation parameter, then only additive models are considered. Multiplicative errors and seasonal components allow ETS models to represent non-stable variance without Box-Cox transformation. The CWE-119 training data was used by the author to fit an alternative ETS model with `ets` that uses no transformation and permits multiplicative trend in the potential model search space. This additional experiment gave the result ETS(M,Ad,N), a model with multiplicative errors and an additive damped trend, which was worse than ETS(A,N,N) as it had higher MAE, RMSE, MAPE and MASE. The point forecasts for ETS(M,Ad,N) were the forecast distributions' medians because of multiplicative errors [8, ch. 7.7].

CWE-119's ETS(A,N,N) model forecasts have accuracy measures MAE $\approx$ 0.495, RMSE $\approx$ 0.624, MAPE $\approx$ 6.45% and MASE $\approx$ 0.695. MASE lower than one means that the ETS(A,N,N) model's forecasts are more accurate on the test set than the average naïve forecasts on CWE-119's training set.

The forecast intervals for CWE-119 in Figure D.4 can be taken seriously as the residuals of the fitted model pass the Ljung-Box test with p-value > 0.05 and the Shapiro-Wilk test with p-value > 0.05. The CWE-119 plot in Figure 5 shows how the actual monthly mean CVSS scores have mostly stayed within the 85% forecast interval. Because of the additive error model, the width of the forecasting errors has a slower growth than it would have been in the case of multiplicative error model. The point forecasts that form a slightly rising straight line, however, they do not zigzag as the test set. The point forecasts around 7.9 indicated "High" severity for CWE-119 in 2016. The 80% forecast intervals also confirmed it. The 95% intervals lower limit gave an indication that CWE-119 could obtain "Medium" severity throughout 2016. This actually happened in June 2016 and in December

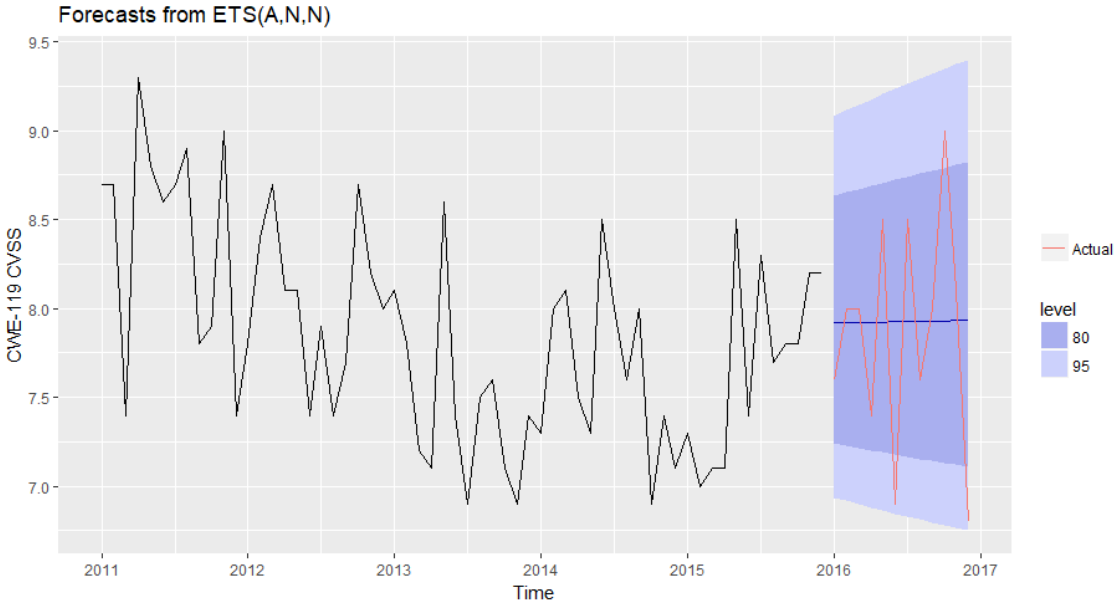

Figure 5: The 2016 CWE-119 CVSS Forecasts

2016 where the mean CVSS scores were 6.9 and 6.8 respectively. CWE-119 stands for "Buffer Errors"[19] [3].

ETS(A,N,N) model was chosen as well for **CWE-59**. Its smoothing parameter is estimated as $\alpha = 0.0001$ and initial state $\ell_0 = 2.0911$. The α for CWE-59, which is very close to zero causes smooth changes to the level. The parameter for the Box-Cox transformation is $\lambda = 0.7463$. The λ is close to one. When λ would be one, then the shape of the data would not change [8, ch. 3.2]. The CWE-59 training data was additionally used by the author for fitting an alternative ETS model for CWE-59 with `ets` that uses no transformation and permits multiplicative trend in the potential model search space. This extra experiment gave also the result ETS(A,N,N) with intial state $\ell_0 = 3.7814$. This model was worse than the chosen ETS(A,N,N) as it had higher MAE, RMSE and MASE: MAE ≈ 3.299, RMSE ≈ 3.425 and MASE ≈ 1.344 against MAE ≈ 3.247, RMSE ≈ 3.364 and MASE ≈ 1.323. Nevertheless, MASE bigger than one for the chosen model means that its forecasts on the test set are less accurate than average naïve forecasts on CWE-59 training set.

The selected ETS(A,N,N) model for CWE-59 passed the Ljung-Box test with p-value > 0.05. However, the null hypothesis of the Shapiro-Wilk test, which states that the residuals are normally distributed, was rejected as p-value ≤ 0.05. Therefore, bootstrapped forecast intervals were generated (Figure 6) and these in-

[19] `http://cwe.mitre.org/data/definitions/119.html`

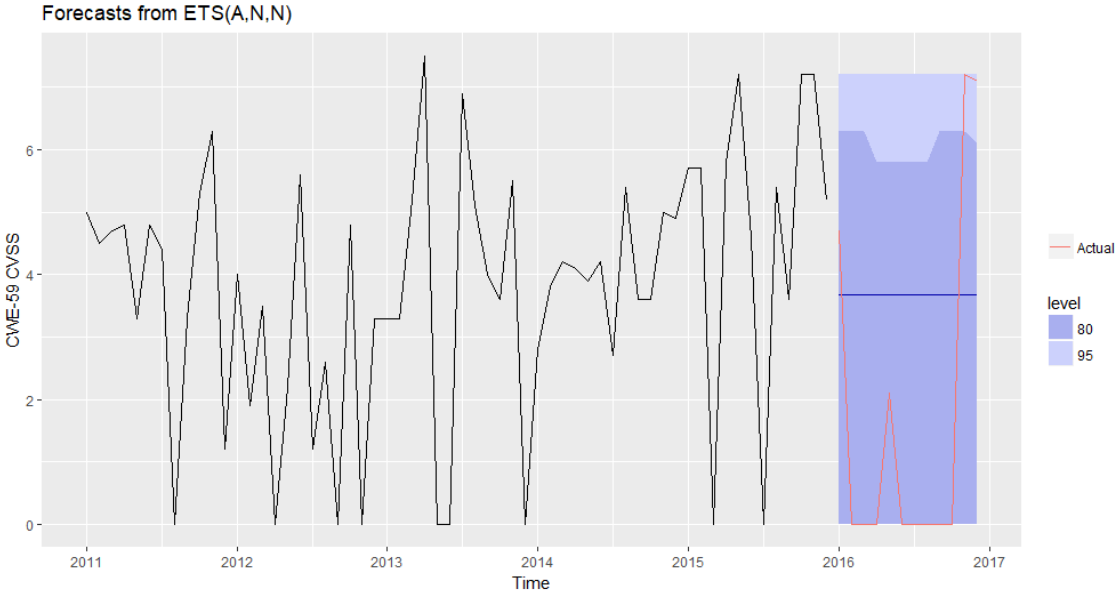

Figure 6: The 2016 CWE-59 CVSS Forecasts

tervals can be taken seriously. According to CWE-59 slightly increasing monthly mean CVSS point forecasts around 3.67, the weakness type should have a "Low" severity in 2016. The lower limit of 80% and 95% intervals suggest that "Low" severity is a possibility. However, the upper limit of 80% forecast intervals suggest that the scores might reach "Medium" severity in the beginning and at the end of 2016, while the upper limit of 95% intervals indicate that it is likely that some future values have a "High" severity throughout 2016. The test set reveals "Medium" severity in January 2016, "High" severity in November and December of 2016 and "Low" severity in May 2016 and in other months of 2016. CWE-59 stands for "Link Following"[20] [3].

The feed-forward neural network autoregression models were created using the 'forecast' package's function `nnetar` which fitted 20 models for every neural network model. Each fitting started with random weight values. Each model's forecasts are averages of the corresponding 20 models. Networks with only one layer and one seasonal lag were generated. Forecast intervals are based on 1000 simulations with errors from normal distribution.

NNAR(12,1,6)$_{12}$, a feed-forward neural network autoregression model, was selected by 'nvdr' for **CWE-79**. It uses 12 lagged values $y_{t-1}, \ldots, y_{t-12}$, which includes an overlapping seasonal input value $y_{t-m} = y_{t-12}$ as input. The hidden layer consists of 6 nodes. The model uses 85 weights: 12×6 weights on links

[20] `http://cwe.mitre.org/data/definitions/59.html`

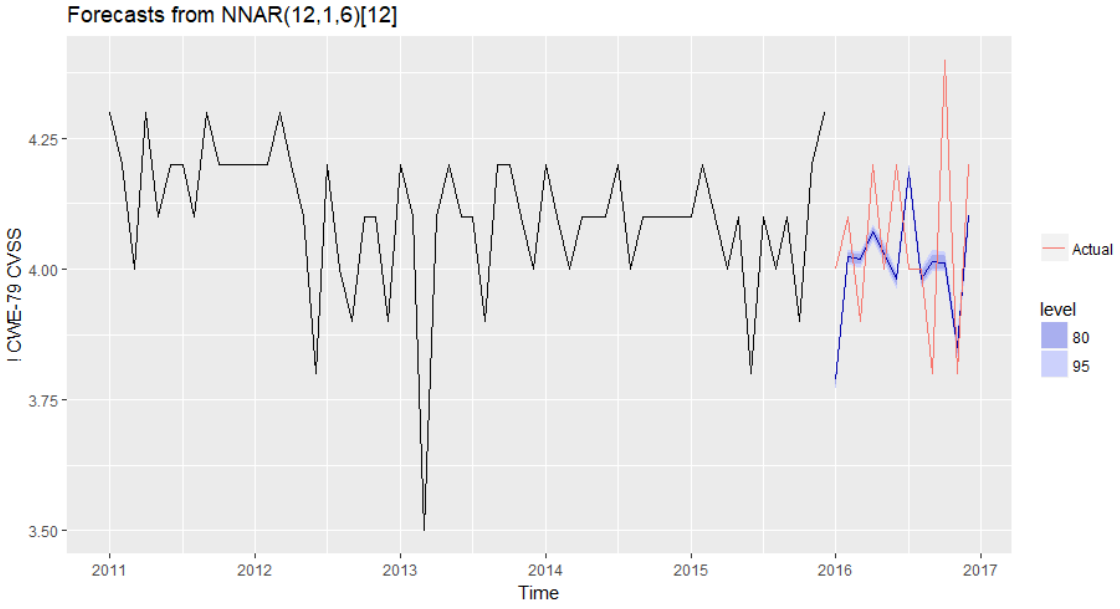

Figure 7: The 2016 CWE-79 CVSS Forecasts

between the input layer and the hidden layer, six weights on links between the hidden layer and the output layer, six bias constants for hidden nodes and one bias constant for the output node. The Box-Cox transformation parameter $\lambda = 1.999924$ is close to two, which means that the observations' values are basically squared in order to stabilise the variance.

Figure 7 shows that the point forecasts for CWE-79 mean CVSS scores are surrounded by narrow forecast intervals. The Shapiro-Wilk test's null hypothesis, which states that the residuals are normally distributed, was rejected. The ACF plot for residuals in Figure D.7 reveals a spike at lag eight going outside the bounds and indicating that there exist autocorrelation in the residuals. As a result, the forecasting intervals cannot be taken seriously. The point forecasts gave an indication that during 2016, CWE-79 would have either "Low" severity or "Medium" severity: "Low" in January, June, August and November. Indeed, CWE-79 had either "Low" or "Medium" severity in 2016 according to the test set. Actual data show that CWE-79 had "Low" severity in March, September and November. The model's forecast accuracy was MAE ≈ 0.145, RMSE ≈ 0.176, MAPE $\approx 3.54\%$ and MASE ≈ 1.087. MASE that close to one means that the NNAR$(12,1,6)_{12}$ model's forecasts on the test set are essentially as accurate as average seasonal naïve forecasts's accuracy on CWE-79's training set. CWE-79 stands for "Cross-Site Scripting (XSS)"[21] [3].

[21] `http://cwe.mitre.org/data/definitions/79.html`

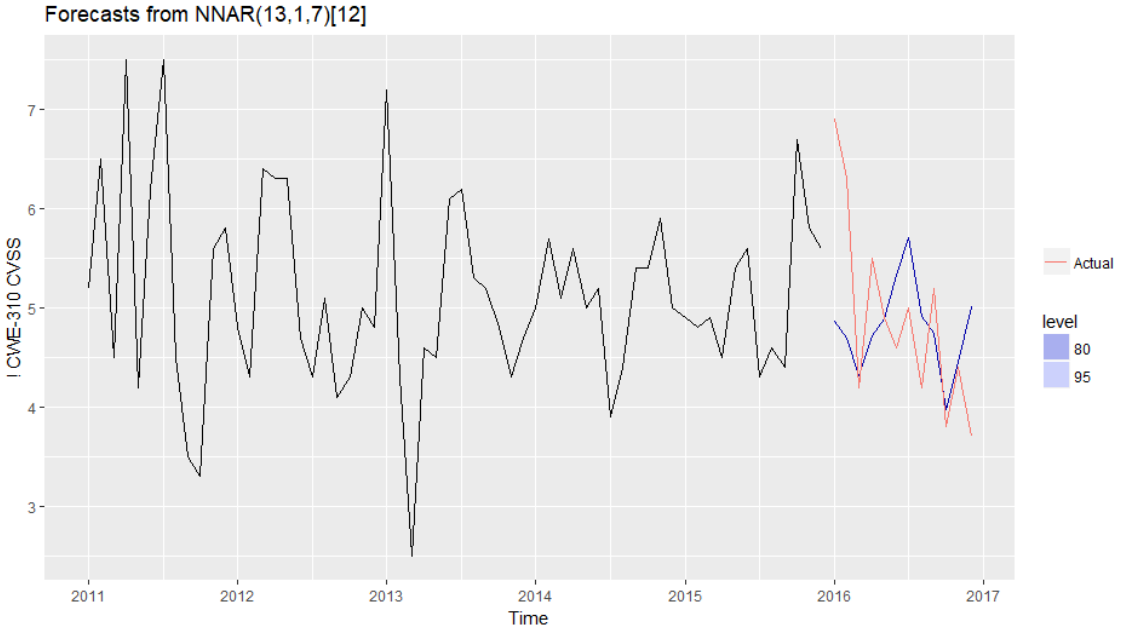

Figure 8: The 2016 CWE-310 CVSS Forecasts

NNAR(13,1,7)$_{12}$ was selected by 'nvdr' for **CWE-310**. It uses 13 lagged values $y_{t-1}, \ldots, y_{t-13}$, which includes an overlapping seasonal input value $y_{t-m} = y_{t-12}$ as input. The hidden layer consists of 7 nodes. The model uses 106 weights: 13×7 weights on links between the input layer and the hidden layer, seven weights on links between the hidden layer and the output layer, seven bias constants for hidden nodes and one bias constant for the output node. The transformation parameter $\lambda = -0.9999242$ is close to -1, which means a reciprocal transformation.

Figure 8 shows that the point forecasts for CWE-310 mean CVSS scores are surrounded by narrow forecast intervals similarly to CWE-79's case. The Shapiro-Wilk test's null hypothesis, which states that the residuals are normally distributed, was rejected similarly to CWE-79's case. The narrow forecasting intervals from Figure 8 cannot be taken seriously. The ACF plot in Figure D.8 shows no spikes outside the bounds indicating that the residuals are uncorrelated. Since the residuals seemed to be uncorrelated but not normally distributed, the author also experimented generating forecasting intervals from bootstrapped residuals. However, the forecasting intervals remained very narrow. This is likely the cause of too complex and overfitted model with small residuals [9, `forecast.nnetar` doc.]. The minimum value of the residuals of the fitted model is -0.0003917728 and the maximum value of the residuals is 0.0007597956. One overfitting sign is the fact that the model's accuracy on training set is expressed by MAE ≈ 0.000225, RMSE ≈ 0.000281, MAPE $\approx 0.00447\%$ and MASE ≈ 0.000203, while the model's

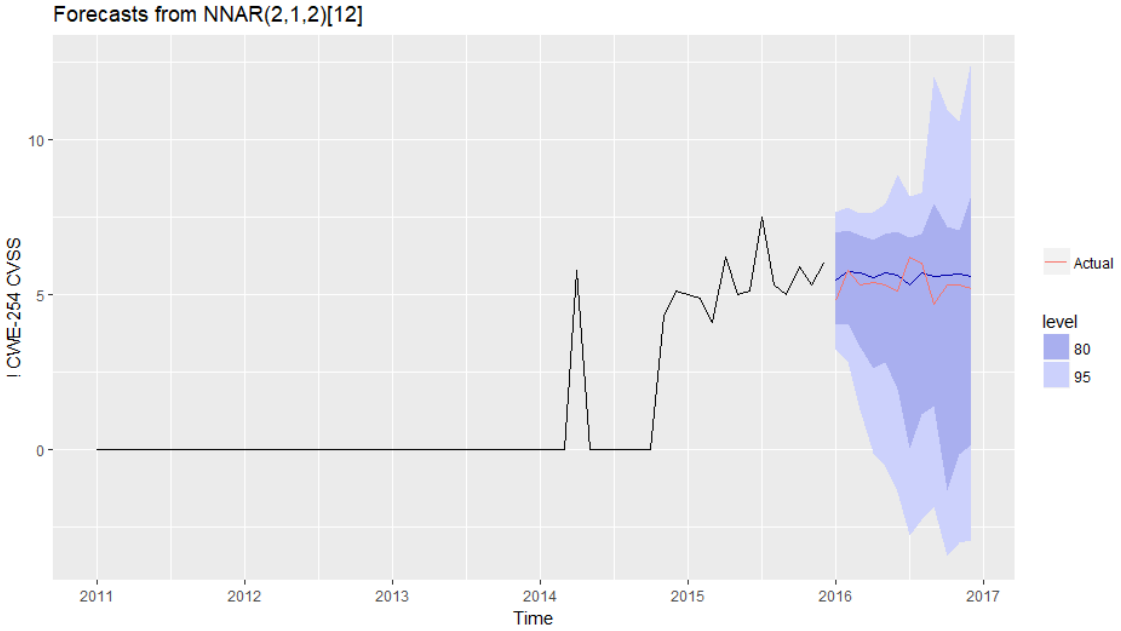

Figure 9: The 2016 CWE-254 CVSS Forecasts

forecast accuracy on test set is much worse: MAE $\approx$ 0.724, RMSE $\approx$ 0.952, MAPE $\approx$ 14.08% and MASE $\approx$ 0.654. The NNAR(13,1,7)$_{12}$ model's forecast accuracy on the test set is better than the average naïve seasonal forecast on the training set of CWE-310 because MASE < 1.

The point forecasts for CWE-310 can still be used. These suggest that the severity of CWE-310 would be "Medium" in 2016 in every month except October with "Low" severity. The test set shows (Figure 8) that CWE-310 actually had "Medium" severity in all months except October and December, which severity levels were "Low". CWE-310 stands for "Cryptographic Issues" [22] [3].

NNAR(2,1,2)$_{12}$ was selected by 'nvdr' for **CWE-254**. It uses two lagged values, y_{t-1} and y_{t-2}, and one seasonal input value y_{t-12} as input. The hidden layer consists of 2 nodes. The model uses 11 weights: 3×2 weights on links between the input layer and the hidden layer, two weights on links between the hidden layer and the output layer, two bias constants for hidden nodes and one bias constant for the output node. The transformation parameter $\lambda = 0.9999998$ is close to one, which indicates that it was not necessary to change the shape of the data.

Figure 9 shows wide forecast intervals surrounding the point forecasts. The training set for CWE-254 in years 2011–2013 did not contain entries and caused the

[22] `http://cwe.mitre.org/data/definitions/310.html`

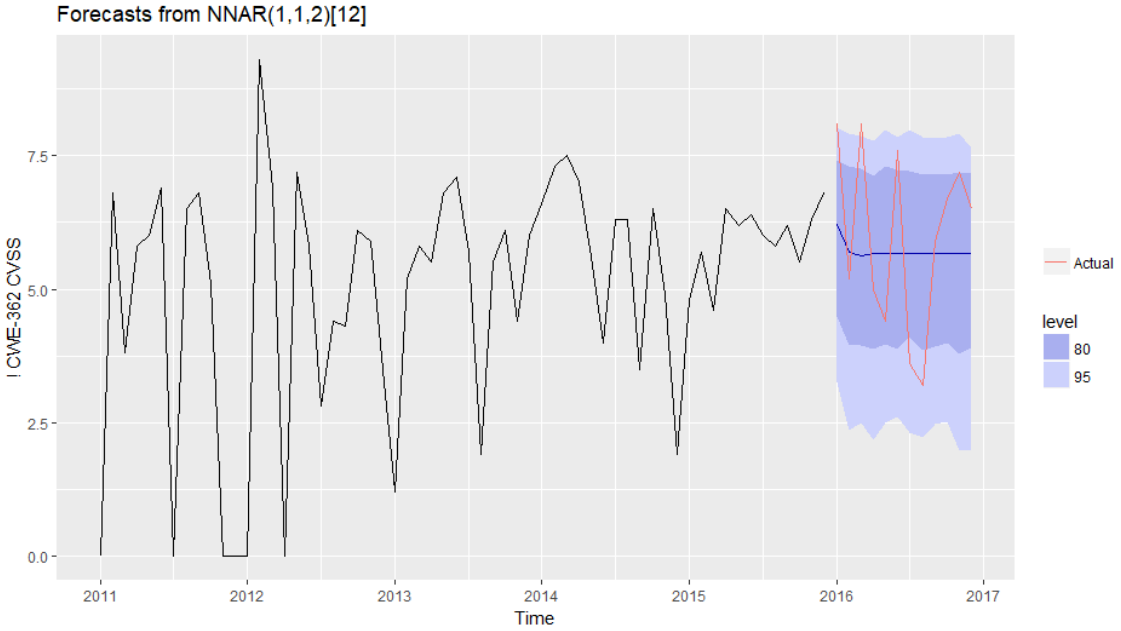

Figure 10: The 2016 CWE-362 CVSS Forecasts

mean CVSS to stay zero during the period. The Shapiro-Wilk test's null hypothesis was rejected: the residuals are not normally distributed. The ACF plot in Figure D.9 reveals a spike outside the bounds at lag seven, which indicates that there exist autocorrelation in the residuals. As a result, the forecasting intervals cannot be taken seriously, in contrast to point forecasts. According to the point forecast, it is likely that CWE-254 has "Medium" severity in 2016. This is confirmed by the test data. The forecast accuracy measures were the following: MAE ≈ 0.438, RMSE ≈ 0.504, MAPE $\approx 8.34\%$ and MASE ≈ 0.322. The NNAR$(2,1,2)_{12}$ model's forecast accuracy on the test set is better than the average naïve seasonal forecast on the training set of CWE-254 because MASE < 1. CWE-254 stands for "Security Features"[23] [3].

NNAR(1,1,2)$_{12}$ was selected by 'nvdr' for **CWE-362**. It uses one lagged value y_{t-1} and one seasonal input value y_{t-12} as input. The hidden layer consists of 2 nodes. The model uses 9 weights: 2×2 weights on links between the input layer and the hidden layer, two weights on links between the hidden layer and the output layer, two bias constants for hidden nodes and one bias constant for the output node. The transformation parameter $\lambda = 1.999959$ is close to two, which means that the observations' values are basically squared in order to stabilise the variance similarly to CWE-79's time series.

[23] `http://cwe.mitre.org/data/definitions/254.html`

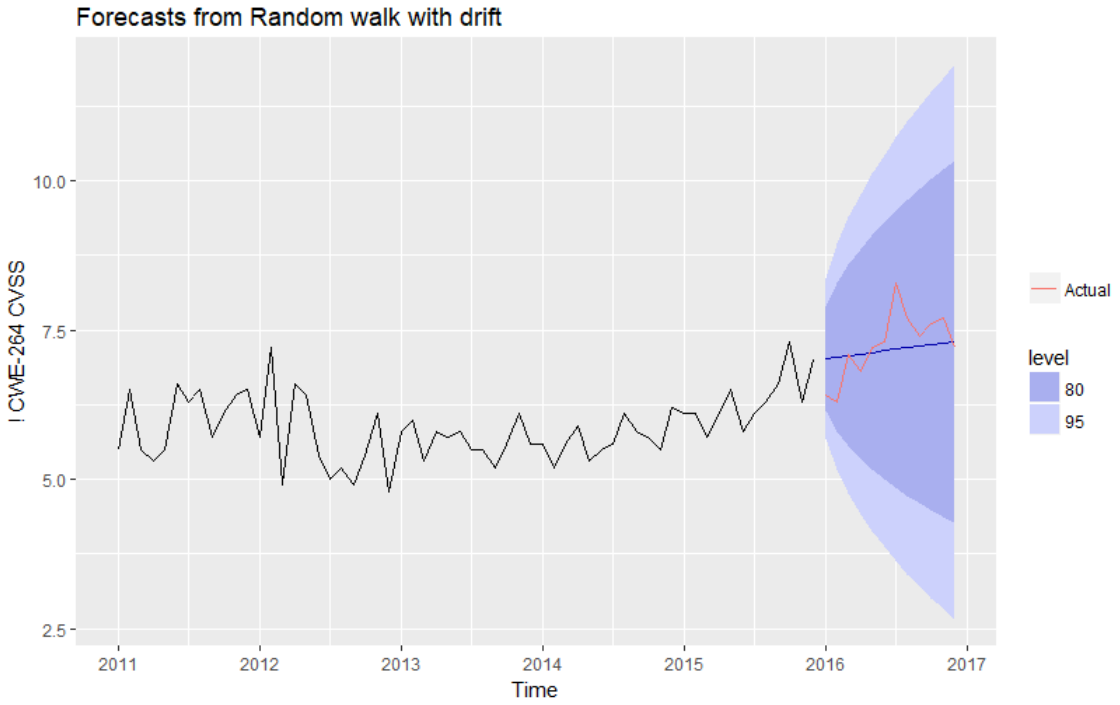

Figure 11: The 2016 CWE-264 CVSS Forecasts

Figure 10 presents the point forecasts, which form a steady line, surrounded by forecast intervals that stay within the limits $0 \ldots 10$ of CVSS scores. These intervals, however, cannot be taken seriously as the ACF plot in Figure D.10 contains a spike at lag 16, which points out autocorrelation in the residuals, and the Shapiro-Wilk test's null hypothesis got rejected, which shows that the residuals are not normally distributed.

The point forecasts of CWE-362 monthly mean CVSS scores stay in the "Medium" severity category. The test set's scores, on the other hand, fluctuate in 2016. The weakness type has "High" severity in January, March, June, and November. It has "Low" severity in July and August and "Medium" severity in other months. The forecast accuracy measures for CWE-362's model MAE ≈ 1.41, RMSE ≈ 1.59, MAPE $\approx 26.57\%$ and MASE ≈ 0.654 include the worst MAPE of the overall best accuracy results (Listing 4.1). MASE < 1 shows that CWE-362's average seasonal naïve forecasts on the training set are worse than NNAR$(1,1,2)_{12}$ forecasts on the test set. CWE-362 stands for "Race Conditions"[24] [3].

On eleven occasions out of 25, the benchmark models from Chapter 3.1.1 gave the best 2016 forecast accuracy results. **Drift model** was chosen by 'nvdr' for **CWE-264**. When the training set is passed to the 'forecast' package's [9] function

[24] `http://cwe.mitre.org/data/definitions/362.html`

`BoxCox.lambda`, then it suggests to use $\lambda = 1.999924 \neq 1$. However, the Box-Cox transformation appeared not to make the forecast accuracy better and was not chosen to be used by 'nvdr'. The forecast accuracy measures for the selected model were MAE ≈ 0.382, RMSE ≈ 0.493, MAPE $\approx 5.27\%$ and MASE ≈ 0.62. Based on Equation (8), the equation for calculating the point forecasts h steps in the future in the case of CWE-263 with training set length $T = 60$, December 2015 mean CVSS $y_{60} = 7.0$ and January 2011 mean CVSS $y_1 = 5.5$ can be written as

$$\hat{y}_{T+h|T} = 7.0 + h\left(\frac{7.0 - 5.5}{59}\right) = 7 + h\left(\frac{1.5}{59}\right) = 7 + \frac{3h}{118}. \tag{55}$$

The point forecasts were suggesting "High" severity with slightly rising monthly mean CVSS for CWE-264 during 2016. The test set revealed that the severity was mostly "High" in 2016. In January, February and April, the severity was actually "Medium".

It is not possible to reject the Shapiro-Wilk test's null hypothesis that the residuals are normally distributed. The residuals are not independently distributed as the Ljung-Box test's null hypothesis got rejected. Consequently, the forecasting intervals for CWE-264 in Figure 11 cannot be taken seriously. CWE-264 stands for "Permissions, Privileges, and Access Control"[25] [3].

Seasonal naïve model was chosen by 'nvdr' for **CWE-20**. The data was transformed using $\lambda = -0.9999242$, which essentially means a reciprocal transformation. The point forecasts are calculated with the transformed monthly data, where $m = 12$ and the training set size $T = 60$, inputted into the Equation (7), which gives

$$\hat{y}_{T+h|T} = y_{60+h-(\lfloor (h-1)/12 \rfloor + 1)12}, \tag{56}$$

where h obtains the values $1 \ldots 12$, when forecasting the CVSS scores of 2016. Therefore, $y_{49}, y_{50}, \ldots y_{60}$ correspond to $\hat{y}_{T+1|T}, \hat{y}_{T+2|T}, \ldots \hat{y}_{T+12|T}$. That means using the CVSS scores from January to December of 2015, the last year of the training set, as forecasts of the CVSS scores of the corresponding months in 2016. Because of the applied Box-Cox transformation, the transformed 2015 monthly values are used for 2016. Later, these values are back-transformed and bias-adjusted. Hence, the point forecast for 2016 in Figure 12 are very similar to the scores of 2015, but slightly different.

The Ljung-Box test's null hypothesis was not rejected when checking the CWE-20 model's residuals. The Shapiro-Wilk test's null hypothesis was also not rejected. Therefore, the forecasting intervals can be taken seriously. The lower limits of both 80% and 95% forecast intervals give the possibility of "Medium" severity CWE-20. The upper limit of 95% intervals show the prospect of "High" severity

[25] `http://cwe.mitre.org/data/definitions/264.html`

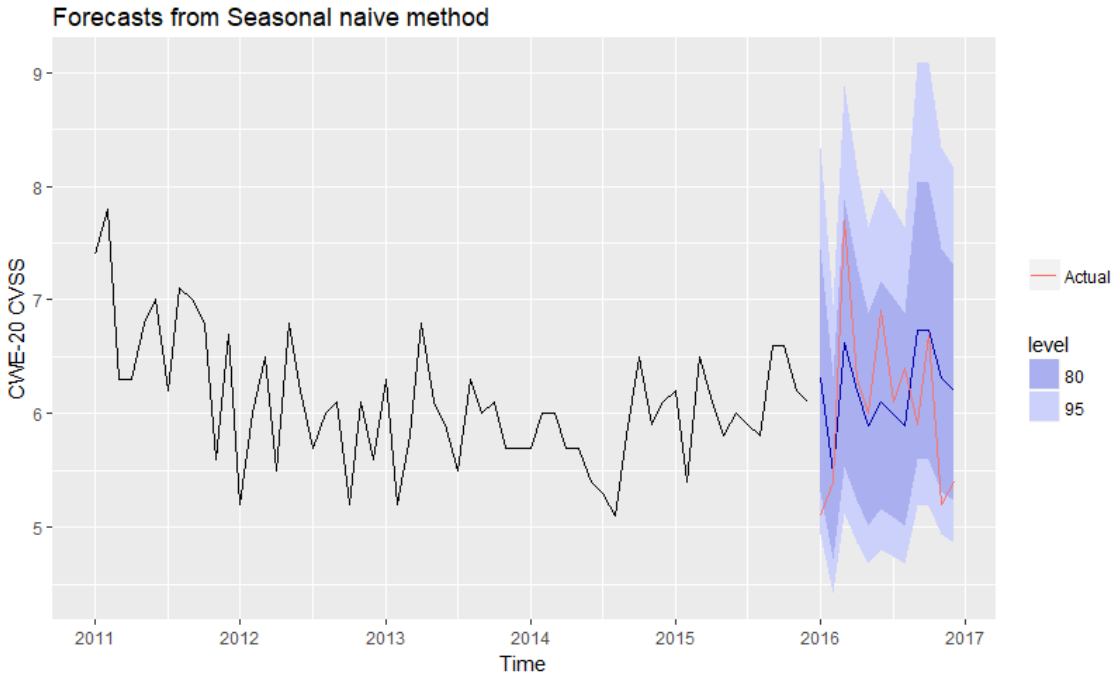

Figure 12: The 2016 CWE-20 CVSS Forecasts

in all months except February, while the upper limit of 80% intervals show the prospect of "High" severity in all months except February, May and August. The point forecasts remain in the "Medium" severity category throughout the year. The forecast intervals cover the test data. The test data shows that CWE-20 had actually "'Medium" severity in all months of 2016 except March with "High" severity. The forecast accuracy measures for CWE-20's model were MAE ≈ 0.563, RMSE ≈ 0.714, MAPE $\approx 9.53\%$ and MASE ≈ 0.907. CWE-20 stands for "Input Validation"[26] [3].

Seasonal naïve model was also chosen by 'nvdr' for **CWE-89**, **CWE-352** and **CWE-77**. The training data was monthly data and similarly to CWE-20's case (Equation (56)), the point forecasts of 2016 are essentially monthly copies of the values from the corresponding months of 2015. The Box-Cox transformation with $\lambda = 0.8237407$ was used for the training data of CWE-77, which resulted in forecast values that are not exactly the copies of the CVSS values of 2015 as it was for CWE-20. The null hypothesis of the Shapiro-Wilk test about residuals' normal distribution was rejected in the cases of CWE-77 and CWE-352 and was not rejected in the case of CWE-89. The null hypothesis of the Ljung-Box test about residuals' independent distribution was rejected in the cases of CWE-89

[26] `http://cwe.mitre.org/data/definitions/20.html`

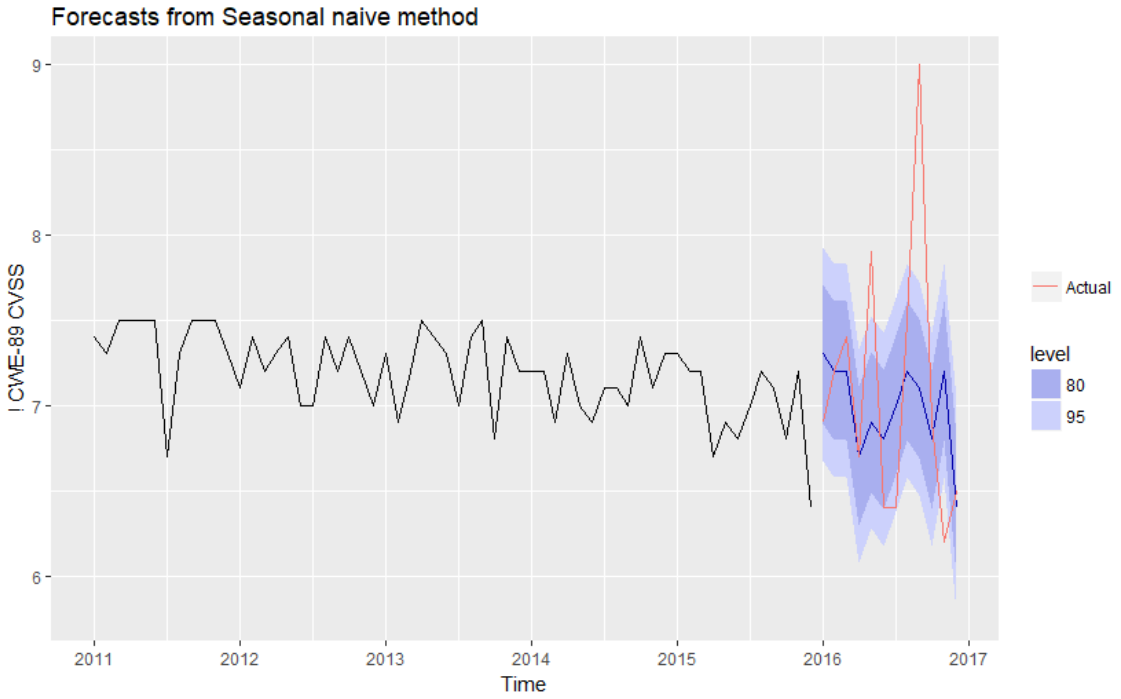

Figure 13: The 2016 CWE-89 CVSS Forecasts

and CWE-77 and not rejected in the case of CWE-352. Consequently, the forecast intervals of CWE-77 and CWE-89 cannot be taken seriously. For CWE-352's seasonal naïve model, however, acceptable forecast intervals, were generated from bootstrapped residuals.

CWE-89's point forecasts indicated "Medium" severity in April, May, June, October and December of 2016 and "High" severity in other months (Figure 13). The test set revealed that CWE-89 indeed switched between "High" and "Medium" severity in 2016 but not entirely the way as forecasted. CWE 352's point forecasts (Figure 14) continued to be all from "Medium" severity level as in 2015. They failed to anticipate a drop in mean CVSS score within "Medium" severity in November and the following rise to "High" severity in December. The CWE-352 model's forecast intervals also did not cover those two spikes in actual values in 2016. Finally, CWE-77's point forecasts did not match the test data CVSS scores' sudden falls to 0.0 in March and July (Figure 15). CWE-77's point forecasts indicated "High" severity in 2016. December 2015 had CVSS score 0.0 and December 2016 forecast was not defined. The actual score was "Medium" in June 2016, but the other actual CWE-77's scores were indeed "High". Since the value of the mean of the residuals of CWE-77 was 1.57, which is not close to zero, then 1.57 should be added to all forecasts in order to remove bias [8, ch. 3.3]. This adjustment does

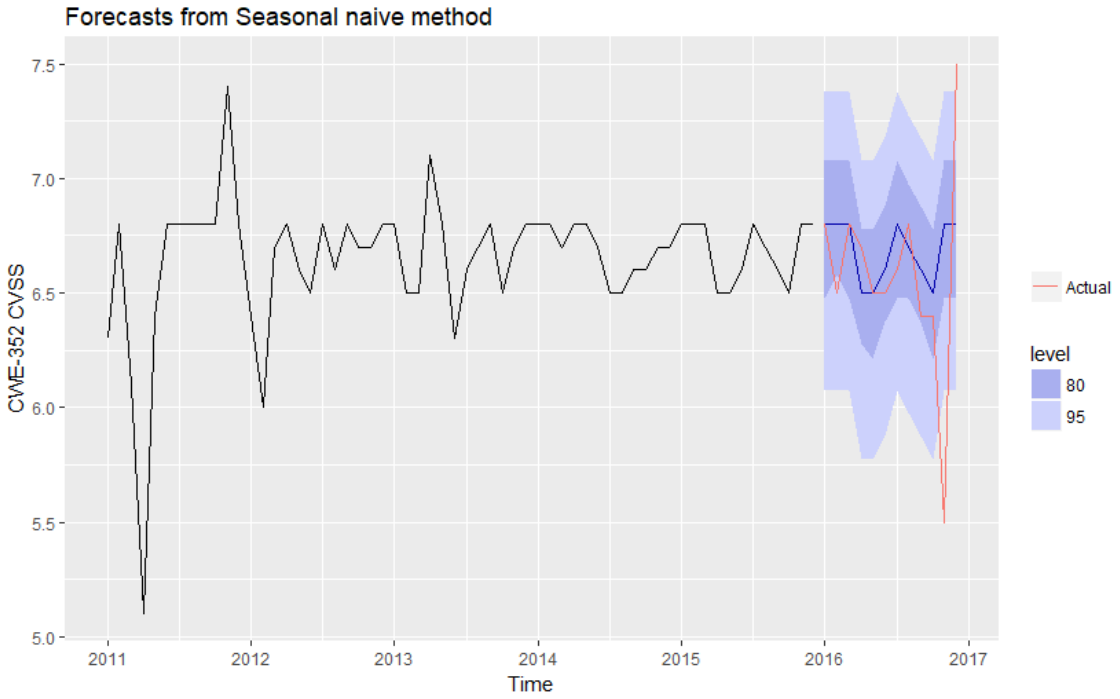

Figure 14: The 2016 CWE-352 CVSS Forecasts

not change the fact that CWE-77's point forecasts indicated "High" severity in 2016.

The forecast accuracy measures, MAE and RMSE, were below 0.7 for both models of CWE-89 and CWE-352 (Listing 4.1). On the other hand, CWE-77's model had MAE ≈ 2.28 and RMSE ≈ 3.59. That kind of forecasting error might be too big given a scale of $0 \ldots 10$ and its severity category ranges [7]. Although CWE 77's model seemed to be worse than the models of CWE-89 and CWE-352 based on the two scale-dependent errors, the scaled error, MASE, is better in the case of CWE-77's model. It has MASE lower than one, while the other two have MASE bigger than one. Therefore CWE-77's seasonal naïve model gives more accurate forecasts than the average seasonal naïve forecasts on CWE-77's training set, while that situation for CWE-89 and CWE-352 models is the other way around. This paragraph did not take into account non-zero mean of CWE-77 residuals. If this bias is removed by adding 1.57 to CWE-77's forecasts, then the forecast accuracy values go worse: MAE ≈ 3.23 and RMSE ≈ 4.46. CWE-89 stands for "SQL Injection"[27], CWE-352 is "Cross-Site Request Forgery (CSRF)"[28]

[27] `http://cwe.mitre.org/data/definitions/89.html`
[28] `http://cwe.mitre.org/data/definitions/352.html`

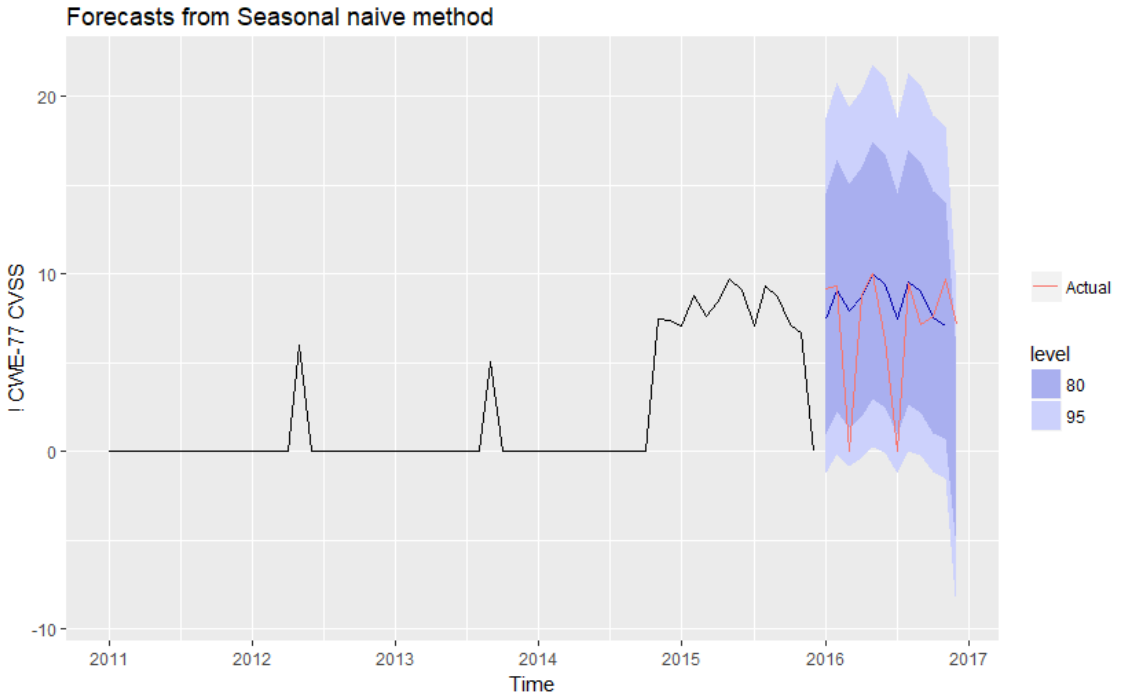

Figure 15: The 2016 CWE-77 CVSS Forecasts

and CWE-77 is "Command Injection"[29] [3].

According to the results from 'nvdr', the overall most accurate forecasts for **CWE-200** (the last row in Figure D.4), **CWE-287** (the second to last row in Figure D.5) and **CWE-19** (the last row in Figure D.6) were produced by **naïve** models. As explained in Chapter 3.1.1, the naïve method takes the last observation's value from the training set and sets it to be equal to all point forecasts in the future. This is exactly how CWE-19's point forecasts are obtained. However, in the cases of CWE-200 and CWE-287, the Box-Cox transformation and bias-adjustments have been used in the naïve model, which means all the point forecast are not equal to the training set last observation's value.

It was not possible to reject the null hypotheses of the Shapiro-Wilk test and Ljung-Box test in the cases of the residuals of naïve models for CWE-200 and CWE-287. CWE-19 model's residuals also did pass the Ljung-Box with p-value $>$ 0.05 but failed the Shapiro-Wilk test with p-value $\leq$ 0.05. Therefore, CWE-19 model's forecast intervals were generated from bootstrapped residuals. The forecast intervals of CWE-200, CWE-287 and CWE-19 started within the CVSS range $0 \ldots 10$ and ended up being wide and partially or completely out of that range as the time went on.

[29] `http://cwe.mitre.org/data/definitions/77.html`

The point forecasts of CWE-200 were in the bottom end of "Medium" severity from January 2016 to October 2016 and in the "Low" severity category in November and December. The point forecasts of CWE-287 indicated "High" severity in January to August of 2016 and "Medium" severity in September to December. The point forecasts of CWE-19 showed "Medium" severity. The test set values were quite close to point forecasts. One obvious difference between forecasts were CWE-19's severity score 6 against its actual score 8.5 in December. CWE-200 stands for "Information Leak / Disclosure"[30], CWE-287 means "Authentication Issues"[31] and CWE-19 means "Data Handling"[32] [3].

According to the results from 'nvdr', the overall most accurate forecasts for **CWE-22** (the last row in Figure D.4) and **CWE-17** (the second row in Figure D.5) were produced by **mean** models. The point forecasts are found by calculating the mean of past observations. The CWE-22 data is transformed using the Box-Cox transformation with parameter $\lambda = 1.999924$ before forecasting. Both null hypotheses of the Ljung-Box test and the Shapiro-Wilk tests are not rejected in the case of CWE-22, while the hypotheses are rejected in the case of CWE-17. Therefore, only the forecasting intervals of CWE-22 might not be inaccurate. According to these intervals, it is expected that 95% of CWE-22 future values belong to a mean CVSS score range from 4.5 to 7.1: "Medium" or "High" severity. CWE-22's point forecasts in 2016 have the value 5.9 ("Medium" severity). The test set shows that CWE-22 actually had "Medium" severity throughout the 2016 except in March with "Low" score.

CWE-17's point forecasts are 1.5 in all months of 2016: "Low" severity. The actual values showed "Medium" severity from January to May 2016 and "Low" severity in other months. The forecast accuracy measures for CWE-22 are MAE $\approx$ 1.05, RMSE $\approx$ 1.86 and MASE $\approx$ 1.30, while MAE and RMSE numbers for CWE-17 are worse but MASE the same: MAE $\approx$ 2.33, RMSE $\approx$ 2.57 and MASE $\approx$ 1.30. The MAE and RMSE above 2 might indicate too big errors on the CVSS score's scale. MASE above one shows that in the both models' cases, average naïve forecasts on the respective training sets are more accurate than the forecasts on the corresponding test sets. CWE-22 stands for "Path Traversal"[33] and CWE-17 means "Code"[34] [3].

TBATS$(1, \{0, 0\}, -, \{12, 2\})$ was chosen by 'nvdr' for **CWE-399**. The first parameter is the parameter of the Box-Cox transformation. The transformation was not necessary as the parameter is equal to one: the shape of the data is not changed. The next two parameters are both equal to zero, which means that the

[30] `http://cwe.mitre.org/data/definitions/200.html`
[31] `http://cwe.mitre.org/data/definitions/287.html`
[32] `http://cwe.mitre.org/data/definitions/19.html`
[33] `http://cwe.mitre.org/data/definitions/22.html`
[34] `http://cwe.mitre.org/data/definitions/17.html`

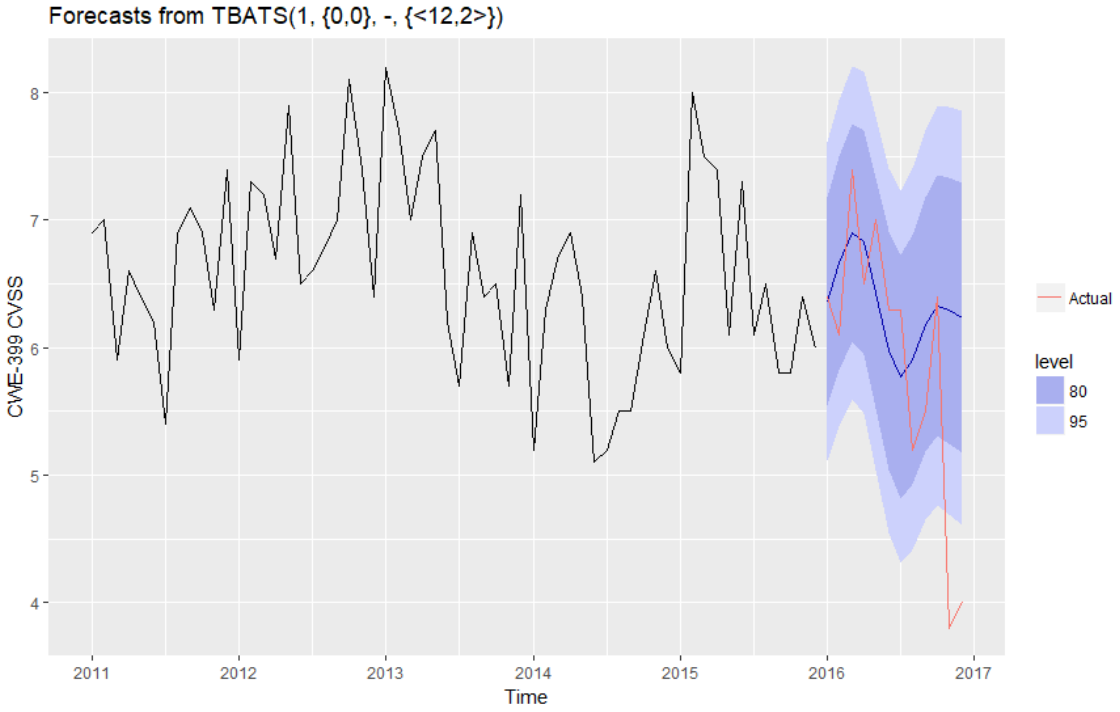

Figure 16: The 2016 CWE-399 CVSS Forecasts

ARMA error term shows no autocorrelation. The third parameter for damped trend was not used. The seasonal period 12 corresponds to the number of months in a year. Two Fourier-like terms were chosen to model the seasonality. The residuals of the selected model pass the Shapiro-Wilk test and Ljung-Box test with p-value > 0.05 in both cases.

The 2016 point forecast of CWE-399 severity stay in the "Medium" category (Figure 16). It is expected that the severity score is the highest at the start of the year, the smallest in the middle and somewhere between the smallest and highest scores at the end of the year. The forecast intervals also follow that tendency. The forecast intervals do not make an expectation that CWE-399's severity could become "Low". They do, however, indicate that the severity could enter into the "High" category. The test set revealed that the severity during most of the year was indeed "Medium", but was "High" in March and May; "Low" in November and December. The forecast accuracy results from line 7 in Listing 4.1 do not seem to be bad on a scale of $0 \ldots 10$. CWE-399 stands for "Resource Management Errors"[35] [3].

BATS$(1, \{0, 0\}, -, -)$ was chosen by 'nvdr' for **CWE-94**. The first parameter,

[35] `http://cwe.mitre.org/data/definitions/399.html`

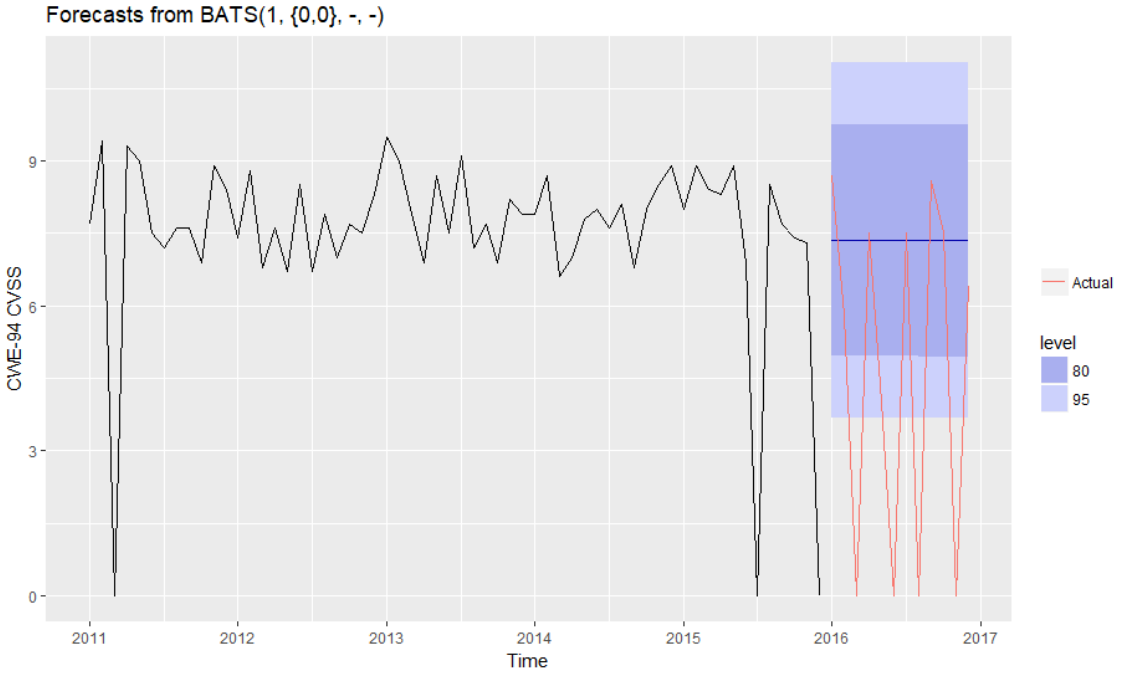

Figure 17: The 2016 CWE-94 CVSS Forecasts

which is equal to one, shows that the Box-Cox transformation was not necessary. The ARMA error term parameters are equal to zero: the error term is not using any lagged observations or lagged errors. The trend is not damped and no seasonal periods are used. The forecast intervals might be inaccurate as it was possible to reject the hypothesis that the model's residuals were normally distributed.

All point forecasts of CWE-94 were set to equal 7.35373, which is "High" severity. The actual mean monthly CVSS values were fluctuating between all three severity categories in 2016. This resulted in MAE ≈ 3.19, RMSE ≈ 4.42 and MASE ≈ 2.36. The forecast errors (as big as indicated by MAE and RMSE on a scale of $0 \ldots 10$ with the given three severity categories) make it hard to use the forecasts in risk assessment. CWE-94 stands for "Code Injection"[36] [3].

ARIMA(0,1,3) using the Box-Cox transformation with $\lambda \approx -1$ was selected by 'nvdr' for **CWE-189**. The model uses 0 lagged observations and 3 lagged errors as predictors (these numbers were chosen by minimising AIC_c by 'forecast' package's [9] `auto.arima`). The training set was needed to be differenced one time in order to make it stationary (the degree of differencing was determined by KPSS unit root tests in `auto.arima`). The coefficients $\theta_1 = -1.1859$, $\theta_2 = 0.2636$ and $\theta_3 = 0.3506$ for $e_{t-1} \ldots e_{t-3}$ in Equation (28) were also estimated by `auto.arima`.

[36] `http://cwe.mitre.org/data/definitions/94.html`

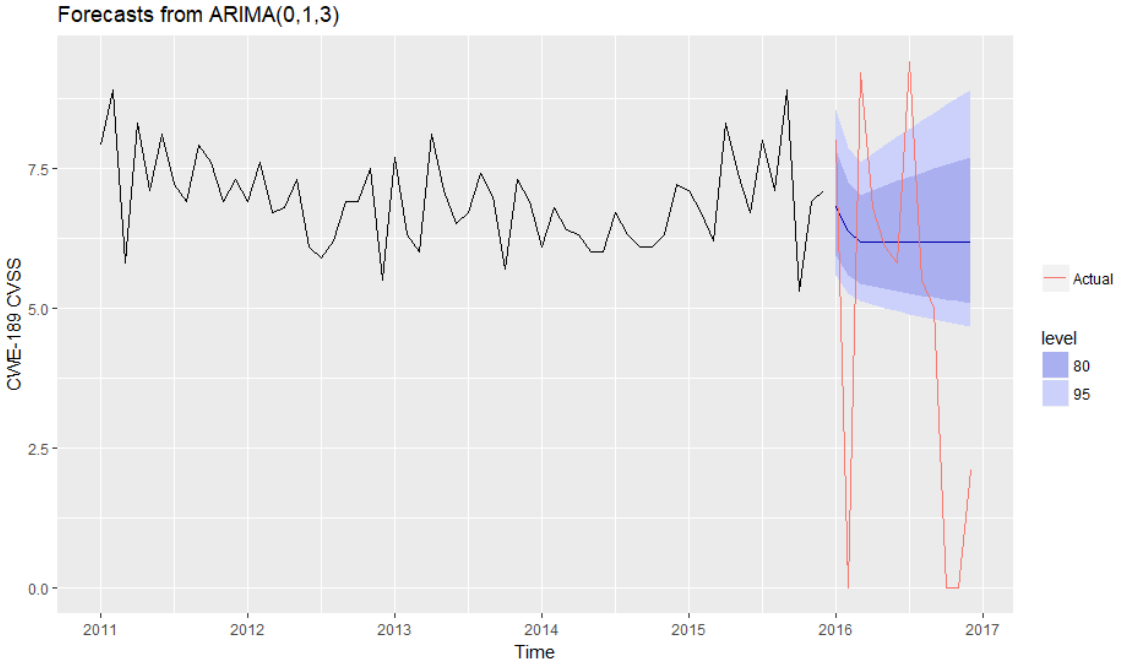

Figure 18: The 2016 CWE-189 CVSS Forecasts

Since the intercept $c = 0$ and the degree of first differencing $d = 1$, then the forecasts will end up being a non-zero constant in the long-term future [8, ch. 8.5]. In CWE-189's case, the point forecasts starting from March are all equal to 6.181341. All the point forecasts expect CWE-189's severity to be "Medium" in 2016 (Figure 18). The null hypotheses of Ljung-Box test and Shapiro-Wilk did not get rejected. The forecast intervals expected CWE-189 to have "Medium" or "High" severity. The test set revealed that to be partially true as in February, October, November and December the severity was "Low". The forecast accuracy with MAE ≈ 2.76, RMSE ≈ 3.62 and MASE ≈ 3.02 is quite low in the given scale. CWE-189 stands for "Numeric Errors"[37] [3].

Linear regression model was selected by 'nvdr' for **CWE-284**. The Box-Cox transformation parameter $\lambda \approx 1$ indicated that the transformation was not necessary. The 'forecast' package's `tslm` function was used with trend and seasonal predictors. As a result, one coefficient for trend, 11 coefficients for seasonal

[37] `http://cwe.mitre.org/data/definitions/189.html`

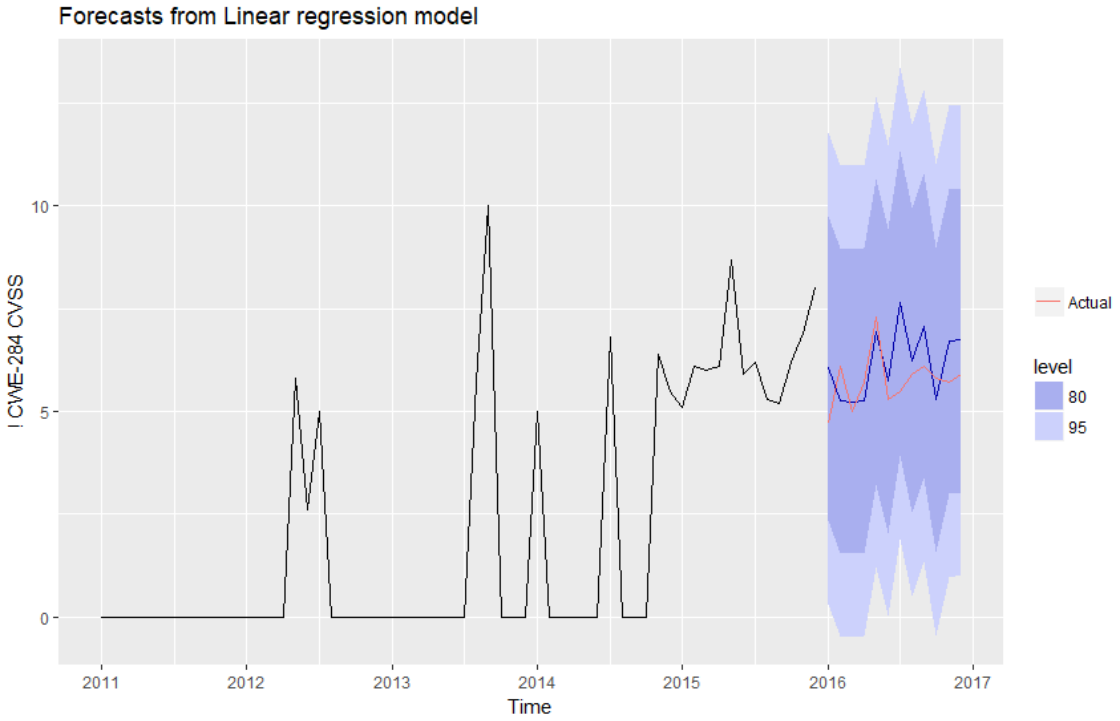

Figure 19: The 2016 CWE-284 CVSS Forecasts

predictors and an intercept were estimated. This gives

$$\begin{aligned} y_t = -1.7873 + 0.1123t - 0.9123s_{2,t} - 1.0446s_{3,t} - 1.1369s_{4,t} \\ +0.4308s_{5,t} - 0.8814s_{6,t} + 0.9062s_{7,t} \\ -0.6060s_{8,t} + 0.1216s_{9,t} - 1.7906s_{10,t} \\ -0.4829s_{11,t} - 0.5552s_{12,t} + \varepsilon_t. \end{aligned} \tag{57}$$

The residuals from CWE-284's model showed correlation in Breusch-Godfrey test. Therefore, the forecast intervals are inaccurate. The point forecast indicated "High" severity in July and in September; otherwise "Medium" severity. The test set had months with "Medium" severity except May which was "High". The MAE, RMSE and MASE were below one (row 13 in Listing 4.1). CWE-284 stands for "Improper Access Control"[38] [3].

Linear regression model was also selected by 'nvdr' for **CWE-190**. The Box-Cox transformation was applied with parameter $\lambda \approx 0.65$. The estimated

[38] `http://cwe.mitre.org/data/definitions/284.html`

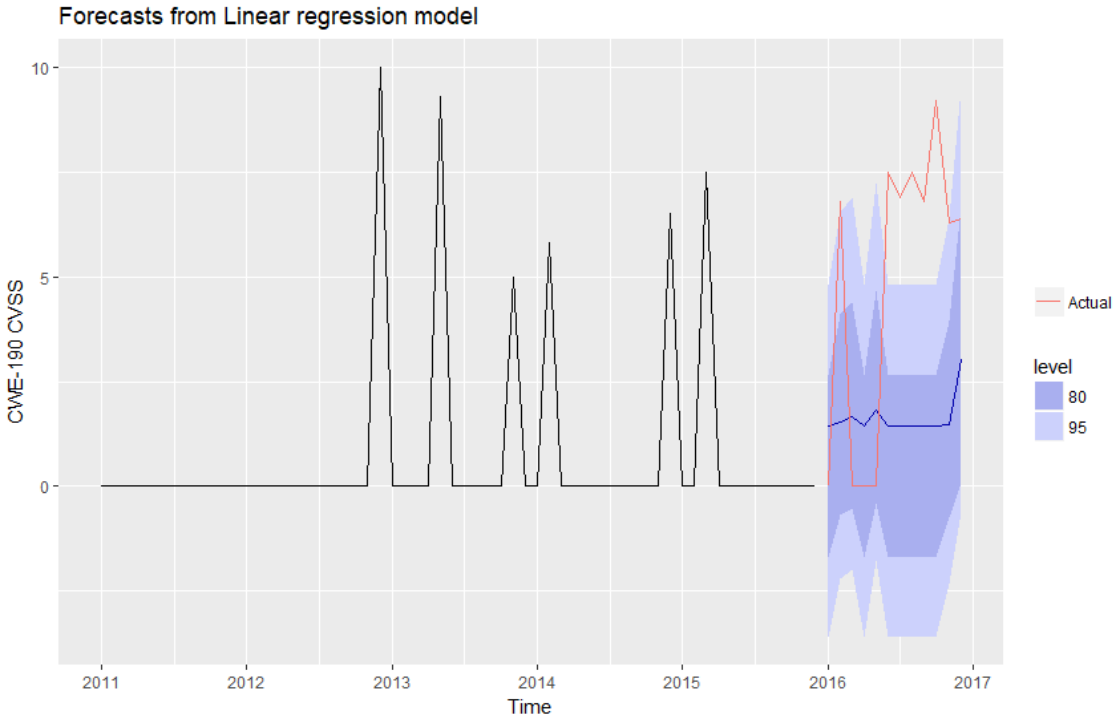

Figure 20: The 2016 CWE-190 CVSS Forecasts

coefficients with a trend and seasonal predictors resulted in

$$\begin{aligned} y_t = -1.78094 + 0.01011t + 0.95543s_{2,t} + 1.12218s_{3,t} - 0.03033s_{4,t} \\ +1.27464s_{5,t} - 0.05055s_{6,t} - 0.06066s_{7,t} \\ -0.07077s_{8,t} - 0.08088s_{9,t} - 0.09099s_{10,t} \\ +0.77508s_{11,t} + 2.30811s_{12,t} + \varepsilon_t. \end{aligned} \tag{58}$$

The Breusch-Godfrey test was not able to reject the hypothesis that the residuals are not correlated. The model's residuals were not normally distributed according to the Shapiro-Wilk test. As a result, the forecast intervals cannot be taken seriously. The point forecasts do not stay near the actual high severity values near the end of 2016 (Figure 20). The CWE-190 model has the highest MAE and RMSE values (both above 4.2) in Listing 4.1, indicating that this is an inaccurate model for CWE-190's CVSS scores. CWE-190 stands for "Integer Overflow or Wraparound"[39] [3].

Bagged ETS model was selected by 'nvdr' for **CWE-255**. The grey area in Figure 21 shows the area between the ensemble forecasts' minimum and maximum

[39] `http://cwe.mitre.org/data/definitions/190.html`

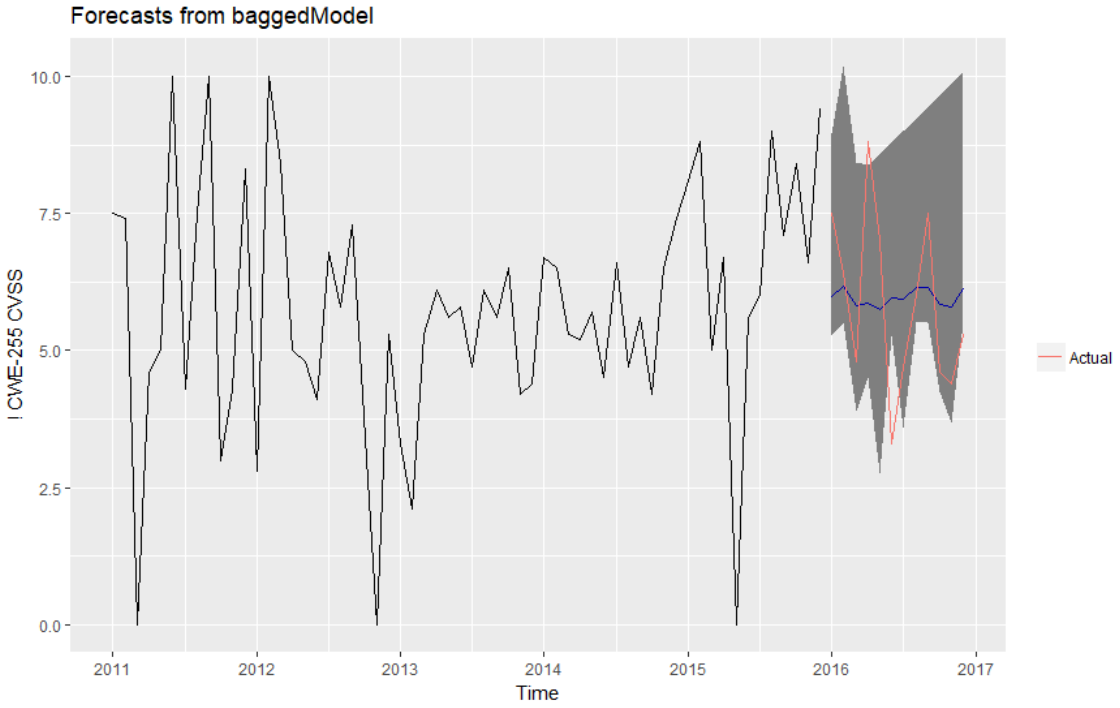

Figure 21: The 2016 CWE-255 CVSS Forecasts

values. The test set's actual values belong to that interval almost all the time. In 2016, the point forecasts showed "Medium" severity, but CWE-255 was actually "Low" in June and "Medium" or "High" in other months. The model had MASE below one, MAE below 1.4 and RMSE below 1.6 (row 15 in Listing 4.1). The ACF plot in Figure D.13 showed no significant spikes: indicating that the model managed to capture the available info adequately. CWE-255 stands for "Credentials Management"[40] [3].

Basic structural models were chosen by 'nvdr' for **CWE-416** (the third row in Figure D.5), **CWE-78** (the fourth row Figure D.5) and **CWE-134** (the last row in Figure D.5). All the models produced wide forecast intervals going beyond the scale of $0 \ldots 10$. The residuals of CWE-416's model were not normally distributed, while the Shapiro-Wilk test's null hypothesis was not rejected in the cases of the other two models.

Bootstrapped residuals are not used for generating forecast intervals for BSM models. Therefore, CWE-416's forecast intervals can be inaccurate. The ACF plot of CWE-78 (Figure D.12) shows no significant autocorrelation in the residuals. The ACF plot of CWE-134 has some spikes only just above the significant lines at lags

[40] `http://cwe.mitre.org/data/definitions/255.html`

one and ten (Figure D.11). As a result, the forecast intervals of CWE-78 and CWE-134 can be taken seriously. These intervals, however, show big uncertainty: CWE-78's forecast intervals cover CVSS scores from "Low" to "High" severity as do CWE-134's 95% forecast intervals. CWE-78's point forecast indicate "High" severity, which is mostly true according to the CWE-78's test set. CWE-134's point forecast indicate mainly "Low" severity in 2016. The test set confirms it with two "High" months as exceptions. CWE-416's point forecasts pointed out "Low" severity at the start of 2016 and "Medium" at the end. The test set had "Low" severity at the start of 2016 and "High" at the end.

All three BSM models had MASE lower than one (Listing 4.1). CWE-416's model had MAE and RMSE approaching to four. On the other hand, CWE-78 and CWE-134 models' MAE scores were below two and at least one unit lower than the respective RMSE. CWE-416 stands for "Use After Free"[41], CWE-78 means "OS Command Injections"[42] and CWE-134 means "Format String Vulnerability"[43] [3].

4.5.2 Forecasting 2017

On 1 January 2018, a new set of CVE-2011 to CVE-2017 XML files were downloaded from NVD. The training sets and test sets downloaded in October 2017 and used in Chapter 4.5.1, were merged into new training sets (covering 2011–2016) for forecasting 2017 with the same types of models that turned out to be having the best forecast accuracy in Chapter 4.5.1. The forecasts of 2017 were evaluated (Listing 4.2) with the test set: the actual values of 2017 that were obtained on 1 January 2018. The corresponding forecast plots with actual values as red lines are presented in Figures D.14, D.15, D.16.

[41] `http://cwe.mitre.org/data/definitions/416.html`
[42] `http://cwe.mitre.org/data/definitions/78.html`
[43] `http://cwe.mitre.org/data/definitions/134.html`

Listing 4.2: 2017 Forecast Accuracy Results (Forecasts by 2016 Best Model Types)

```
      cwe                          method    MAE   RMSE    MAPE   MASE
 1: CWE-119                 ETS(A,N,N) 0.9212 1.0181 13.7862 1.3682
 2:  CWE-79          NNAR(12,1,6)[12] 0.1203 0.1370  3.0222 0.8108
 3: CWE-264   Random walk with drift 0.5570 0.6909  8.5100 0.8273
 4:  CWE-20    Seasonal naive method 0.8225 1.1071 14.0746 1.3446
 5: CWE-200           Naive method 0.2944 0.3762  7.1267 0.6591
 6: CWE-310          NNAR(13,1,7)[12] 1.0228 1.3849 19.3717 0.9025
 7: CWE-399 TBATS(1, {0,0}, -, {<12,2>}) 1.0780 1.2542 17.1131 1.2135
 8:  CWE-89    Seasonal naive method 0.6333 0.7757  9.0240 2.0994
 9: CWE-352    Seasonal naive method 0.3083 0.4113  4.7501 1.3504
10:  CWE-22                    Mean 0.4028 0.4680  7.6268 0.4347
11: CWE-189  ARIMA(1,0,2)(1,1,0)[12] 2.3407 2.9303     Inf 1.7044
12:  CWE-94    BATS(1, {0,0}, 1, -) 3.9540 4.0359 56.9575 1.9721
13: CWE-284  Linear regression model 1.5089 1.6412 26.4400 0.6252
14: CWE-287           Naive method 1.1719 1.3391 17.0040 1.0479
15: CWE-255            baggedModel 0.7863 1.0200 16.5594 0.3380
16: CWE-254          NNAR(2,1,2)[12] 0.3663 0.4294  6.9868 0.3035
17:  CWE-17                    Mean 2.1924 2.4491     Inf 0.9824
18: CWE-416   Basic structural model 0.9063 1.0914 13.6725 0.2116
19:  CWE-78   Basic structural model 0.9602 1.1357 10.9547 0.2663
20: CWE-134   Basic structural model 4.1500 5.0494     Inf 1.1846
21: CWE-190  Linear regression model 3.1221 3.2925 46.5227 1.2866
22:  CWE-77    Seasonal naive method 1.3310 1.7576 17.9477 0.4948
23: CWE-362          NNAR(1,1,2)[12] 1.0536 1.1317 18.0817 0.5148
24:  CWE-59                 ETS(A,N,N) 1.6787 2.0538 38.7391 0.6371
25:  CWE-19           Naive method 2.5000 2.5994 43.6481 1.2931
      cwe                          method    MAE   RMSE    MAPE   MASE
```

With a training set from years 2011 to 2016, the `auto.arima` function used by 'nvdr' chose seasonal ARIMA(1,0,2)(1,1,0)$_{12}$ for CWE-189 (row 11 in Listing 4.2), as opposed to the non-seasonal ARIMA(0,1,3) when the training set from period 2011–2015 was used (row 11 in Listing 4.1). In order to compare the forecasting accuracy results for 2016 and 2017, Listing D.1 shows the differences obtained by subtracting the accuracy scores of Listing 4.1 from the scores of Listing 4.2. When the difference in Listing D.1 is negative, then the forecasts accuracy values of 2017 generated by the types of models selected in Chapter 4.5.1 were smaller and therefore indicating higher accuracy. In the cases of 14 CWEs, at least one accuracy measure was smaller in Listing 4.2 than in Listing 4.1. This means that for 14 CWEs, the model types that showed the best combined 2016 forecast accuracy based on training sets covering 2011–2015 and test sets covering the year of 2016, showed better forecast accuracy in 2017 when using training sets covering 2011–2016 (downloaded in October 2017) and test sets covering the year of 2017 (downloaded in January 2018).

Listing 4.2 displays some CWEs, which accuracy measures might show too big forecasting error on a scale of $0 \ldots 10$, where the score range for "Medium" severity[44] is 2.9 units wide. CWE-134 forecasts have MAE ≈ 4.15 and RMSE ≈ 5.05. CWE-94 forecasts have MAE ≈ 3.95, RMSE ≈ 4.04 and MAPE $\approx 56.96\%$.

[44] $4 \ldots 6.9$

CWE-190 forecasts have MAE ≈ 3.12, RMSE ≈ 3.29 and MAPE ≈ 46.52%. The measures for CWE-19 and CWE-189 forecasts are not good enough as well in Listing 4.2. CWE-94, CWE-190 and CWE-189 had problematic accuracy numbers also back in Listing 4.1, where the best model types were discovered for 2016 with a purpose to use them later for producing the forecasts for 2017. The first row in Figure D.16 shows how CWE-190's actual mean monthly severity scores of 2017 stay almost completely within 80% forecast interval but are higher than the point forecasts. The second row in Figure D.15 shows how BATS model's point forecasts for CWE-94's severity were lower than the actual values in 2017.

The analysis[45] of the data downloaded on 1 January 2018 covering CVE-2011 to CVE-2017 and focusing on the period of 2011 to 2017, revealed the same 25 CWEs that were selected for forecasting in Chapter 4.5.1. Nine additional CWEs were also discovered during the analysis. The Venn diagram in Figure 22 shows how these nine additional vulnerability categories got grouped.

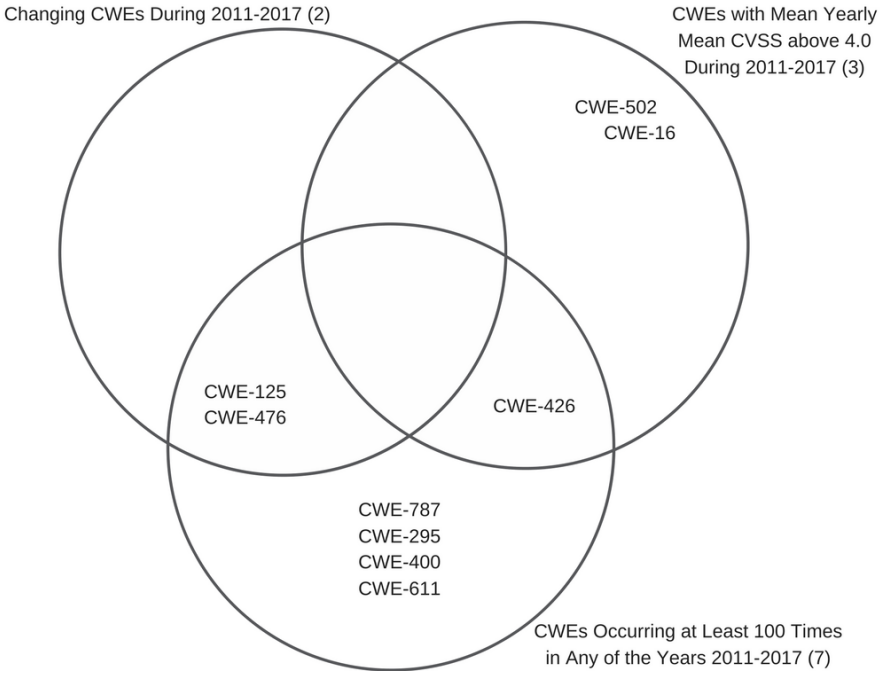

Figure 22: Venn Diagram for Additions from Period 2011-2017

Using the data downloaded on 1 January 2018, divided into training sets covering years 2011 to 2016 and test sets covering the year of 2017, the best model types for the 34 CWEs were found according to the combination of forecast accuracy measures (as described in Chapter 4.3). The accuracy numbers are presented

[45]three CWE subset selection techniques from Chapter 4.3

in Listing 4.3. The corresponding point forecasts as blue lines, actual values as red lines and forecast intervals as blue shades are presented in Figures D.17, D.18, D.19 and D.20.

Listing 4.2 and Listing 4.3 present forecast accuracy scores for the year of 2017. The same test sets were used for both of them when calculating the accuracy values. The training sets cover the years 2011 to 2016 on both occasions: Listing 4.2's training data as of 7 October 2017 and Listing 4.3's data as of 1 January 2018. A subset of 25 CWEs is present in both listings. For Listing 4.2, the model types were chosen based on the combined 2016 forecast accuracy measures in Chapter 4.5.1, which were then used to forecast 2017. For Listing 4.3, the model types were chosen based on the combined 2017 forecast accuracy measures. These types of models will be used to forecast severity scores of 2018 later in Chapter 4.5.3.

The types of best models from Listing 4.2 do not match the best models from Listing 4.3 with one exception. Only CWE-200's forecasts are generated by Naïve model in Listing 4.2 and in Listing 4.3. This shows that the types of models for 25 CWEs that were considered as the best in 2016 are not the best in 2017.

Listing 4.3: Best Forecast Accuracy Results for 2017

	cwe	method	MAE	RMSE	MAPE	MASE
1:	CWE–119	Naive method	0.37500000	0.4462809	5.386683	0.5569307
2:	CWE–79	Linear regression model	0.08121736	0.1024320	2.035639	0.5475327
3:	CWE–264	ARFIMA(1,0,1)	0.37283559	0.4741895	5.711266	0.5537162
4:	CWE–200	Naive method	0.19166667	0.2783882	4.837254	0.4291045
5:	CWE–20	Naive method	0.38175480	0.5370734	6.376364	0.6241223
6:	CWE–399	ARFIMA(2,0,0)	0.58142235	0.7191906	10.081456	0.6545092
7:	CWE–310	Basic structural model	0.81971075	1.1920859	15.063772	0.7232742
8:	CWE–284	BATS(1, {0,0}, –, –)	0.29956607	0.3471762	5.304435	0.1241296
9:	CWE–89	ETS(A,A,N)	0.18465679	0.2170827	2.617375	0.6121220
10:	CWE–352	baggedModel	0.14806487	0.1819293	2.310083	0.6484593
11:	CWE–22	baggedModel	0.23269656	0.2866719	4.270779	0.2511114
12:	CWE–189	ETS(A,N,N)	1.98709753	2.6498527	Inf	1.4469157
13:	CWE–94	Mean	0.45717593	0.5428850	6.685522	0.2280179
14:	CWE–287	ARIMA(0,0,1) with non–zero mean	0.32850538	0.3779543	4.817176	0.2937455
15:	CWE–254	baggedModel	0.34613118	0.4280210	6.655734	0.2868490
16:	CWE–125	baggedModel	0.35879457	0.5683681	6.134811	0.2237804
17:	CWE–255	Random walk with drift	0.71866197	0.9625254	13.544382	0.3088805
18:	CWE–416	BATS(1, {0,0}, –, –)	0.45122888	0.5761656	6.570453	0.1053453
19:	CWE–476	baggedModel	0.35585877	0.4319187	6.551802	0.3177310
20:	CWE–77	BATS(1, {0,0}, –, –)	0.41666667	0.5688198	5.386913	0.1548947
21:	CWE–190	Random walk with drift	0.57382629	0.7138271	9.106968	0.2364669
22:	CWE–19	NNAR(2,1,2)[12]	0.66893517	0.7864337	11.811581	0.3460009
23:	CWE–787	Random walk with drift	0.66408451	0.8622575	10.897496	1.2490618
24:	CWE–17	Naive method	1.38333333	2.8029746	100.000000	0.6198656
25:	CWE–295	Basic structural model	1.14061757	1.3295490	24.802723	2.2586487
26:	CWE–426	Naive method	2.19166616	2.6408643	Inf	2.4905297
27:	CWE–400	Naive method	1.25728629	1.5565527	21.946271	2.5571925
28:	CWE–611	ARIMA(0,1,1)	0.56847605	0.6663001	9.948420	0.9370484
29:	CWE–78	Linear regression model	0.67921719	0.8313963	8.262578	0.1884098
30:	CWE–134	ARFIMA(1,0.22,0)	3.10736802	3.2557086	Inf	0.8869747
31:	CWE–362	Linear regression model	0.63251881	0.7434977	11.354391	0.3090483
32:	CWE–59	BATS(1, {1,0}, –, –)	1.58098236	1.8716219	40.491002	0.5999933
33:	CWE–502	Naive method	0.35833333	0.4907477	4.940694	0.5810811
34:	CWE–16	Naive method	1.99166667	4.0797263	100.000000	0.7155689
	cwe	method	MAE	RMSE	MAPE	MASE

Although the evaluated CWE-200's forecasts are generated by Naïve model in Listing 4.2 and Listing 4.3 and the test sets are identical, the MAE, RMSE, MAPE and MASE values are not equal, indicating that the forecasts are different. The CWE 200's training sets as of October 2017 and as of January 2018 were checked for differences as NVD might have updated something related to 2011–2016 period, but no mean monthly CVSS score differences were found: the used training sets were exactly the same. The dissimilarity was caused by the fact that the CWE-200 forecasts that were evaluated in Listing 4.2 had used Box-Cox transformation with bias-adjustment but the CWE-200 forecasts that were evaluated in Listing 4.3 had not used the transformation.

Listing 4.3 shows that the forecast accuracy measures MAE, RMSE and MASE for the majority of CWEs' severity forecasts have values below one. Many MAPE values are below 10%. However, MAPE values for the forecasts of CWE-16 and CWE-17 are 100%. On both occasions, the point forecasts are equal to zero, which eventually leads to 100% error.

CWE-134, CWE-19, CWE-94, CWE-189 and CWE-190 severity forecast errors in Listing 4.2 were quite high given the severity scale range $0 \ldots 10$. The corresponding forecasts for 2017 were generated by these types of models that showed the best combined (MAE, RMSE, MAPE and MASE) forecast accuracy in 2016. When the best types of models were later found in 2017 based on the best combined 2017 forecast accuracy, the forecast errors of CWE-19, CWE-94 and CWE-190 severity forecast were better: MAE, RMSE and MASE were below one (Listing 4.3).

4.5.3 Forecasting 2018

The initial data used in Chapter 4.5.1 revealed 25 CWEs which severity was decided to be forecasted (Figure 4). They were discovered by applying the three CWE selection techniques from Chapter 4.3. The processing of new data obtained on 1 January 2018 gave the same 25 CWEs and nine additional CWEs (Figure 22). Forecasts were generated to find out the potential severity of these 34 software security weaknesses in 2018. The best model types for forecasting 2018 were determined based on the data downloaded on 1 January 2018 in Chapter 4.5.2 by finding the types of models which gave the lowest values for the maximum number of accuracy measures (Listing 4.3). Table D.1 helps to understand the meanings of the CWEs mentioned in this chapter.

On eight occasions, the **naïve** models were chosen by 'nvdr' to generate forecasts. The forecast intervals cannot be taken seriously in the cases of CWE-119, CWE-200, CWE-17, CWE-426 and CWE-16. The models' residual caused the null hypothesis of the Ljung-Box test to be rejected in these cases.

The point forecasts for **CWE-119** by **naïve** model are equal to 7.4 in 2018 (1st row in Figure 23). This means "High" severity. The training set's monthly mean CVSS values have shown mostly "High" severity in each of the past years. The year of 2017 had more "Medium" severity months than 2011–2016: the first two years and 2015 had none, 2013 and 2016 had two, 2014 one, while 2017 had six "Medium" severity months. The Naïve model took the CVSS value of December 2017 and forecasted it to be each month's severity score in 2018.

The severity of **CWE-200** is expected to be "Medium" with mean monthly CVSS score 4.6 according to the **naïve** model (4th row in Figure 23). It has taken the last value of 2017 and it expects that to be the severity score in 2018. The training set shows that CWE-200's severity has been mostly "Medium" during 2011–2017, sometimes "'Low" as well. The mean monthly CVSS scores seem to fluctuate less towards the end of the training period when compared with the start of the period.

"High" severity is forecasted for **CWE-20** by **naïve** model (5th row in Figure 23). The Box-Cox transformation with parameter $\lambda \approx 0$ is applied – it is as

taking a logarithm. The forecast intervals give the chance of "High" or "Medium" severity all through 2018. The lower limits of 95% intervals enter the "Low" severity range starting from August 2018. During the past from 2011 to 2017, CWE-20 has mostly had mean monthly severity scores belonging to the "Medium" severity category. Furthermore, in 2017, it was "High" only in December.

CWE-17 has had "Low" severity during most of the period covered by the training set (4th row in Figure 25). From November 2014 to May 2016, the severity was steadily around "Medium" scores but then faded away with only some higher spikes afterwards. The **naïve** model uses the mean monthly severity score 0.0 from December 2017 as a forecast for all future months of 2018.

The severity scores of four out of nine additional CWEs from Figure 22 were forecasted by **naïve** models. Those vulnerabilities are **CWE-426** ("Untrusted Search Path"[46]), **CWE-400** ("Uncontrolled Resource Consumption"[47]), **CWE-502** ("Deserialization of Untrusted Data"[48]) and **CWE-16** ("Configuration"[49]) [3]. In 2018 according to point forecasts, CWE-426 (1st row in Figure 25) and CWE-502 (3rd row in Figure 26) are expected to have "High" severity, while CWE-400's severity would be "Medium" (2nd row in Figure 25) and CWE-16's severity would be "Low" (5th row in Figure 26). The bootstrapped 95% forecast intervals for CWE-400 and CWE-502 are wide and cover "Low", "Medium" and "High" severity score range in 2018. The 80% intervals cover smaller score range at the start of 2018. The Box-Cox transformation with $\lambda \approx 0.86$ was used in the case of CWE-400.

Linear regression models generated forecasts on three occasions: for CWE-79, CWE-78 and CWE-362. The forecast intervals for the first two CWEs's severity can be incorrect as the Shapiro-Wilk test's null hypothesis was rejected: the models' residuals were not normally distributed.

According to the **linear regression** model for **CWE-79**, which point forecasts' equation uses a trend predictor and 11 seasonal predictors,

$$\begin{aligned} y_t = 8.474243 - 0.007313t - 0.171953s_{2,t} - 0.720297s_{3,t} - 0.096619s_{4,t} \\ -0.326423s_{5,t} - 0.596938s_{6,t} - 0.193238s_{7,t} \\ -0.651590s_{8,t} - 0.402874s_{9,t} - 0.104878s_{10,t} \\ -0.446099s_{11,t} - 0.210238s_{12,t} + \varepsilon_t, \end{aligned} \tag{59}$$

CWE-79 would have either "Low" or "Medium" severity in 2018 (2nd row in Figure 23). The forecasted CVSS scores are only just above 4.00 in January, February, April, July, October and December. The scores of the other months

[46] https://cwe.mitre.org/data/definitions/426.html
[47] https://cwe.mitre.org/data/definitions/400.html
[48] https://cwe.mitre.org/data/definitions/502.html
[49] https://cwe.mitre.org/data/definitions/16.html

are approximately equal to 3.9. The first month of 2018 has been forecasted to have higher severity than the following eleven months. The negative coefficients in front of the 11 seasonal predictors in Equation 59 show that on average, the first month of a year has a higher severity CWE-79 than the other months. The mean monthly CWE-79 severity score has stayed close to four in the training set with two distinct spikes: CVSS 3.5 in March 2013 and CVSS 4.4 in October 2016 (2nd row in Figure 23). The data was transformed by the Box-Cox transformation with the parameter $\lambda \approx 2.0$.

Linear regression model's point forecast equation for calculating the severity of **CWE-78** can be represented as

$$\begin{aligned} y_t = 13.25441 + 0.17464t - 0.01361s_{2,t} - 7.02014s_{3,t} - 2.64683s_{4,t} \\ +0.42963s_{5,t} - 3.97067s_{6,t} - 3.67620s_{7,t} \\ -2.09047s_{8,t} - 0.22292s_{9,t} - 0.37365s_{10,t} \\ -2.64165s_{11,t} - 3.98850s_{12,t} + \varepsilon_t. \end{aligned} \tag{60}$$

The estimated coefficients before the seasonal predictors indicate that on average, the fifth month of a year has higher severity than the first month, while the other months have lower severity than the first month. In 2018, CWE-78 is forecasted to have "High" severity in all months (4th row in Figure 25). The mean monthly CVSS score is expected to be the highest in October and the lowest in March. The data was transformed using the Box-Cox transformation with $\lambda \approx 1.8$. The training set from the period 2011–2017 has mostly "High" mean monthly CVSS score. There exist some spikes, where the score is 0.0. Last such spikes appeared in March and July 2016.

The severity of **CWE-362** is forecasted to be "High" in February, March and June 2018 and "Medium" in other months of 2018 (1st row in Figure 26). The coefficients before seasonal predictors in equation

$$\begin{aligned} y_t = 6.9089 + 0.0986t + 12.1199s_{2,t} + 9.2951s_{3,t} + 5.1139s_{4,t} \\ +8.2994s_{5,t} + 9.7499s_{6,t} - 1.1158s_{7,t} \\ +1.8975s_{8,t} + 2.7596s_{9,t} + 5.7737s_{10,t} \\ +2.9131s_{11,t} - 0.1861s_{12,t} + \varepsilon_t \end{aligned} \tag{61}$$

show that on average, the seventh month and the twelfth month of a year have lower severity than the first month, while the other months have higher severity than the first month. The Box-Cox transformation with $\lambda \approx 2$ was applied to stabilise the variance of the training set before estimating the forecasting model. The plot (1st row in Figure 26) shows that over time, CWE-362 has had smaller downwards spikes of mean monthly CVSS scores.

ARFIMA(1,0,2) model's point forecasts foresee "Medium" severity in 2018 for **CWE-264** (3rd row in Figure 23). The autoregressive fractionally integrated moving average model do not use differencing in the specific case. One lagged observation and 2 lagged errors have the role of predictors. The Box-Cox transformation with the parameter $\lambda \approx 1.4$ was applied for stabilising the variance of the training set. From the mid-2012 up to July 2016, the time series plot's mean monthly scores had an upwards trend (3rd row in Figure 23). Later, the monthly scores gradually declined. The ARFIMA model's point forecasts indicate that the downwards trend continues in "Medium" category in 2018. The forecast intervals cannot be taken seriously because the ACF plot of the residuals (Figure D.21) has a spike at lag 9, which is slightly crossing the threshold line.

CWE-399 should have steadily increasing "Medium" severity scores in 2018 (1st row in Figure 23). The forecasts are produced by a **ARFIMA(0,0.29,0)** model. Its orders of the autoregressive part and the moving average part are equal to zero. The estimated difference coefficient $d = 0.29$ is a non-integer, something that is possible in the case of an ARFIMA model but not in the case of an ARIMA model. The forecast intervals might be inaccurate as the ACF plot in Figure D.22 shows that on two occasions the correlations go outside the threshold. The mean monthly CVSS scores during 2011–2017 have stayed in "Medium" or "High" categories with one exception: CWE-399 had a "Low" score in November 2016. On the other hand, May 2012, October 2012, January 2013 and February 2015 have been with the highest severity: 7.9, 8.1, 8.2 and 8.0 respectively. To stabilise the variance of the training set, the Box-Cox transformation with the parameter $\lambda \approx 2$ was used.

ARFIMA(0,0.06,0) model forecasts that the mean monthly severity score of **CWE-134** in 2018 would be about 3, which means "Low" (5th row in Figure 25). No lagged errors or lagged observations have been used as predictors. The forecast intervals cannot be taken seriously as the model's residuals led to the null hypothesis rejection of Shapiro-Wilk test and showed autocorrelation in ACF plot in Figure D.24. The training set shows that CWE-134's monthly mean CVSS score have alternated between 0.0 and "High" or "Medium" scores in the past (5th row in Figure 25).

The **Basic structural model** (BSM) forecasts that the mean monthly severity of **CWE-310** would be close to 4.9, "Medium" severity, in 2018. The Shapiro-Wilk test's null hypothesis did not get rejected and the ACF-plot (Figure D.23) did not show significant lines outside the threshold. The forecast intervals show the possibility of "Low" severity or "Medium" severity in 2018. CWE-310, however, has had "High" mean monthly severity in the past. The highest score from the training set was in December 2017, the last observation of the training set.

Another **BSM** generated forecasts of the severity of **CWE-295**, which stands

for "Improper Certificate Validation"[50] [3]. In 2018, this vulnerability is expected to have a "Medium" severity according to point forecasts (5th row in Figure 25). Forecast intervals are not taken seriously as the model's residuals are not normally and independently distributed. Figure D.25 shows a spike in the ACF plot. The mean monthly CVSS score of CWE-295 has mostly been equal to 0.0 during 2011–2017. Since December 2016, there have been only higher scores than 0.0.

BATS models generated forecasts on four occasions: for **CWE-284**, **CWE-416**, **CWE-77** and **CWE-59**. On none of those occasions, the forecast intervals could be taken seriously. In the cases of CWE-284, CWE-416, the mean monthly severity score values in 2018 are forecasted to belong to the "Medium" score range. CWE-59, on the other hand, is expected have "Medium" severity in January and February and "Low" severity in other months. CWE-77's 2018 mean monthly CVSS score's point forecasts are calculated to be equal to 7.7, which is "High" severity. BATS(1,{0,0},-,-) models were used for CWE-284, CWE-416 and CWE-77, while BATS(1,{1,0},-,-) was used for CWE-59. BATS(1,{0,0},-,-) model uses no Box-Cox transformation, no ARMA error, no damped trend and no seasonal periods. BATS(1,{1,0},-,-) model takes use of an ARMA error term that uses 1 lagged observation. CWE-284 plot is in Figure 23 (3rd row), CWE-416 and CWE-77 can be found from Figure 24 (3rd row and 5th row) and CWE-59's time series plot is in Figure 23 (2nd row).

ETS(A,N,N) models generated the mean monthly CVSS forecasts for **CWE-89** and for **CWE-189**. ETS(A,N,N) model is a simple exponential smoothing model with additive errors, which has no trend component and no seasonal component. Both vulnerability types were forecasted to have "Medium" severity in 2018: CWE-89's monthly mean CVSS approximately 6.99 in each month (4th row in Figure 23) and CWE-189's monthly severity value approximately 4.8 (2nd row in Figure 24). However, the mean of CWE-189 model's residuals is not zero. Instead, it is -1.3. To remove that bias, -1.3 should be added to all CWE-189's forecasts. This results in mean monthly CVSS score approximately 3.5, which represents "Low" severity category. CWE-189 model's residuals are not normally and independently distributed. Its forecast intervals may be therefore inaccurate. However, CWE-89 model's forecast intervals from bootstrapped residuals can be taken seriously. Both CWE-89 model's 80% and 95% forecast intervals cover scores from "Medium" and "High" category during 2018.

For both CWE-89 and CWE-189 data, Box-Cox transformation using the parameter $\lambda \approx 2$ was applied. In the past, CWE-89 has mostly had mean monthly CVSS scores around 6 and 7. However, in September 2016, the score spiked to 9.0 (4th row in Figure 23). The past of CWE-189 from the period 2011–2017 shows bigger fluctuations towards the end of the period (2nd row in Figure 24).

[50] `https://cwe.mitre.org/data/definitions/295.html`

Bagged ETS models forecasted "Medium" severity for **CWE-352**, **CWE-22**, **CWE-254**, **CWE-125** ("Out-of-bounds Read"[51] [3]) and **CWE-476** ("NULL Pointer Dereference"[52] [3]). The bagged ETS models' plots can be found in Figures 23 and 24, where they feature grey areas surrounding the point forecasts. These are the areas between minimum and maximum values of the bagged ETS ensemble forecasts. The areas are wide[53] for CWE-254, CWE-125 and CWE-476, while they are narrow[54] for CWE-352 and CWE-22. The training set parts of the plots of CWE-254, CWE-125 and CWE-476 look similar: many zeros at the beginning of the period 2011–2017, which then disappear towards the end of the period and become replaced with consecutive values higher than zero.

Mean models' forecasts, the means of past observations, were calculated for **CWE-94** and for **CWE-287** (3rd and 4th row in Figure 24). In the case of CWE-94, the point forecasts indicate "High" severity for 2018; the forecast intervals may be inaccurate because of the results from Shapiro-Wilk and Ljung-Box tests. In the case of CWE-287, the point forecasts indicate "Medium" severity for 2018; the forecast intervals can be taken seriously. 95% and 80% forecast intervals of CWE-287's model show the possibility that instead of "Medium" severity, the mean monthly scores could be "High". CWE-287's training set was transformed by Box-Cox transformation with the parameter $\lambda \approx 2$ before the model estimation, while the transformation was not applied in the case of CWE-94's observations. CWE-94's training set has multiple visible downward spikes in Figure 24 (3rd row). CWE-287's training set has one big downwards spike in Figure 24 (4th row) in May 2015.

The 2018 mean monthly CVSS score of **CWE-255** is decreasing gradually from 4.7 to 4.3 in the "Medium" severity category, while **CWE-190** and **CWE-787** ("Out-of-bounds Write"[55] [3]) show gradually increasing "High" severity scores. These forecasts were generated by **drift** models, which forecast intervals could not be taken seriously because of the residual analysis results. The Box-Cox transformation was not used in any of these cases. CWE-255's plot is presented in the second row in Figure 24. The plots of CWE-190 and CWE-787 can be found from the first and third rows in Figure 25. The training sets of CWE-190 and CWE-787 contain mostly mean monthly CVSS scores equal to 0.0: their scores start to have consecutive numbers higher than 0.0 starting from mid-2016.

The severity of **CWE-19** in 2018 is forecasted by a feed-forward neural network autoregression model, $\mathbf{NNAR(2,1,2)}_{12}$, which has two nodes in the hidden layer and uses three input values: two lagged values and one monthly seasonal lagged

[51] `https://cwe.mitre.org/data/definitions/125.html`
[52] `https://cwe.mitre.org/data/definitions/476.html`
[53] At some point in 2018, they cover values of "Low", "Medium" and "High" score categories.
[54] Most of 2018, they cover only "Medium" severity score range.
[55] `https://cwe.mitre.org/data/definitions/787.html`

value. Links between the layers of nodes and bias constants altogether required the calculation of 11 weights in the model. The point forecasts of mean monthly CVSS scores of CWE-19 are approximately equal to 6.2 in every month of 2018 (2nd row in Figure 25). This means "Medium" severity for CWE-19. The uncertainty of those forecasts is not known as the forecast intervals may be inaccurate. The model's residuals are not normally distributed and the ACF plot in Figure D.26 has a significant spike at lag 10. The training set of CWE-19 shows that the mean monthly CVSS score in years 2011–2014 was mostly 0.0. Starting from November 2014, the scores have been above zero with two exceptions in September and November of 2015 (2nd row in Figure 25). The training set was transformed by Box-Cox transformation using the parameter $\lambda \approx 1$, which essentially means no transformation.

CWE-611 stands for "Improper Restriction of XML External Entity Reference ('XXE')"[56] [3]. The Box-Cox transformation with the parameter $\lambda \approx 0.70$ was applied to its training set of mean monthly CVSS scores before the estimation of **ARIMA(0,1,1)** model. It has differenced the data once and it uses one lagged error as a predictor with the estimated coefficient $\theta_1 = -0.2834$. All the point forecasts in 2018 are approximately equal to 5.3 (3rd row in Figure 25). This means "Medium" monthly severity for CWE-611. The forecast intervals based on bootstrapped residuals are wide and cover all severity score categories "Low", "Medium" and "High" almost throughout 2018. The 80% forecast intervals are narrower in January, February and March and cover only "Medium" scores. CWE-611's training set reveals that its severity has been continuously above 0.0 starting from August 2016 (3rd row in Figure 25).

[56] `https://cwe.mitre.org/data/definitions/611.html`

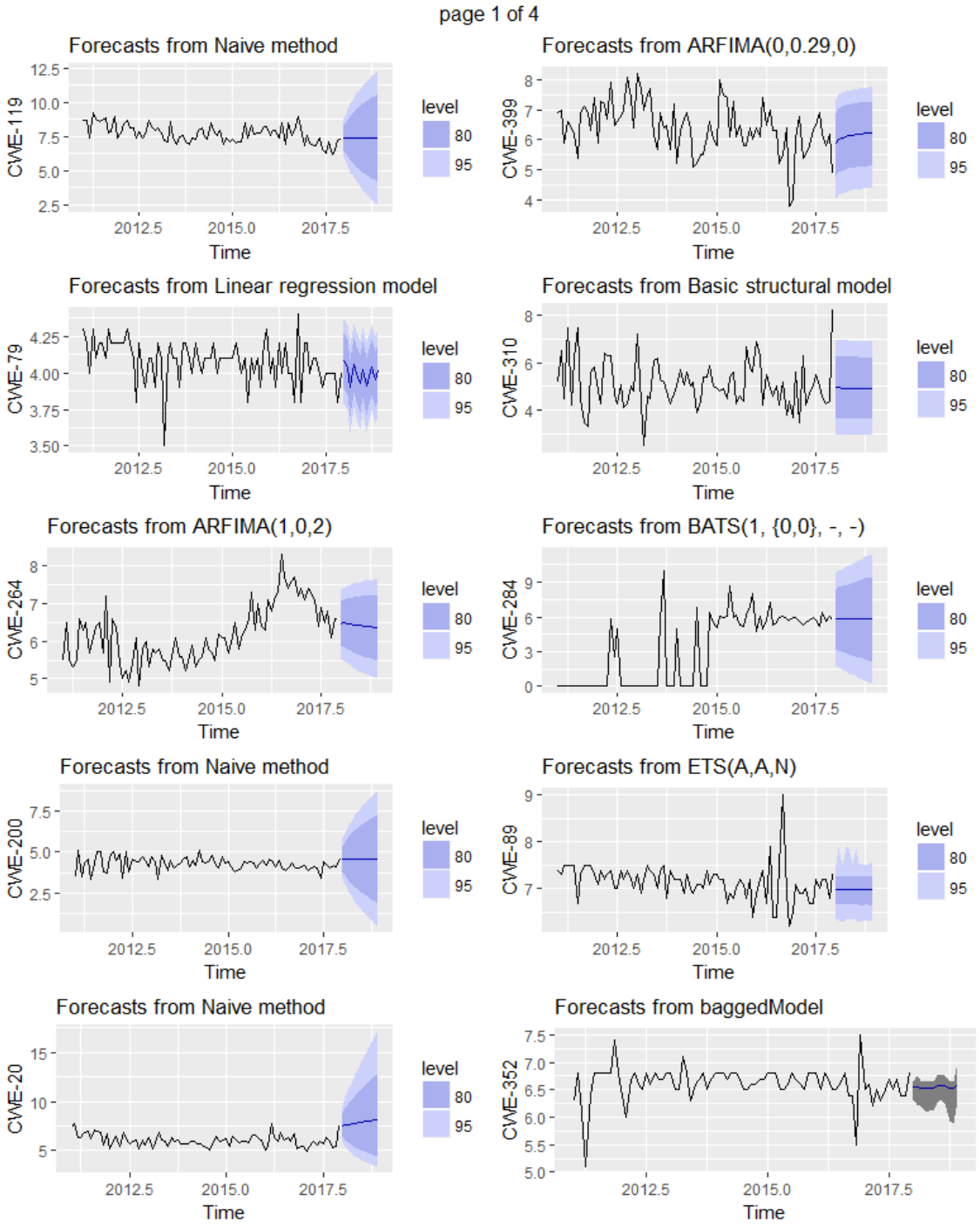

Figure 23: The 2018 Forecasts by 2017 Best Model Types (Page 1)

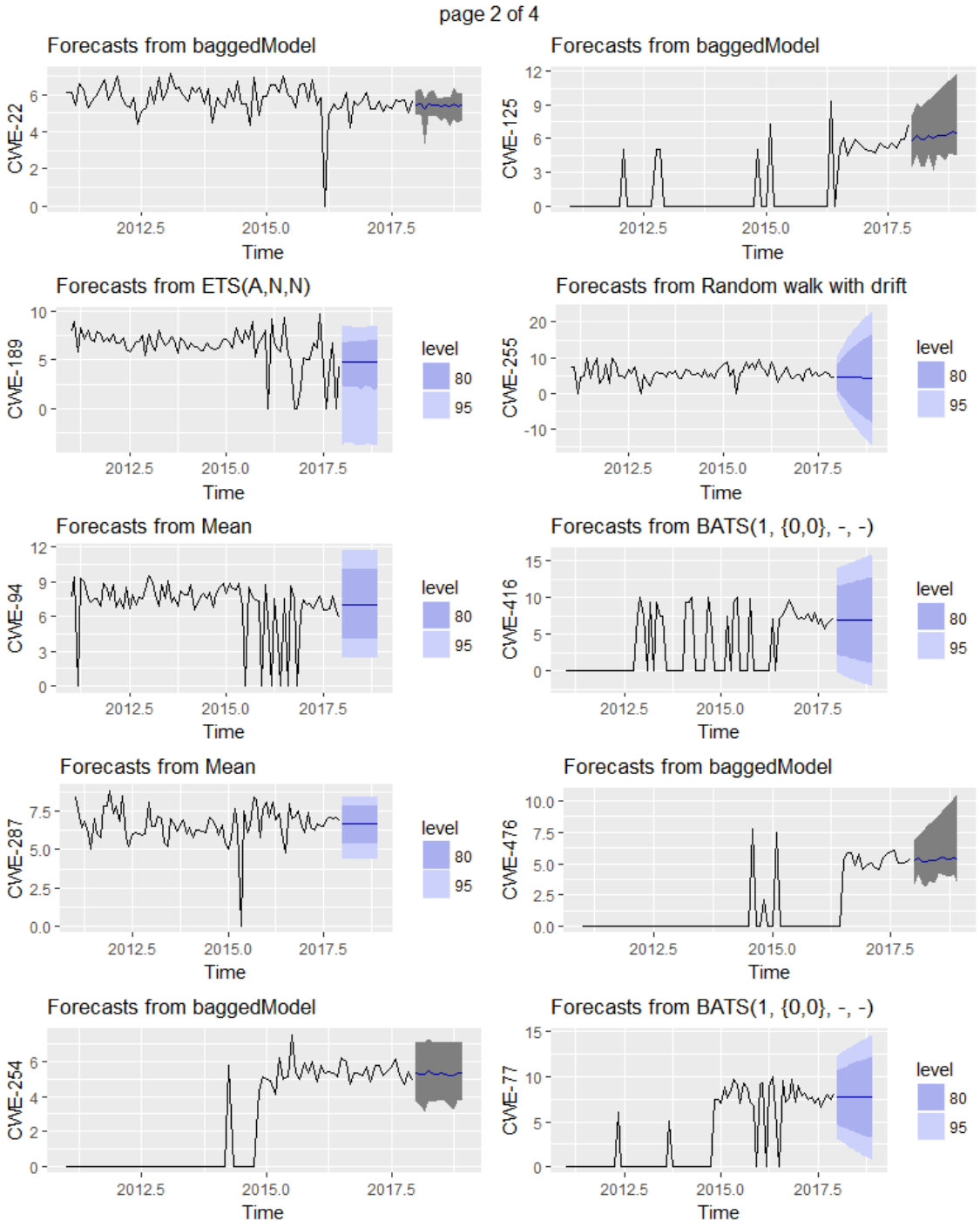

Figure 24: The 2018 Forecasts by 2017 Best Model Types (Page 2)

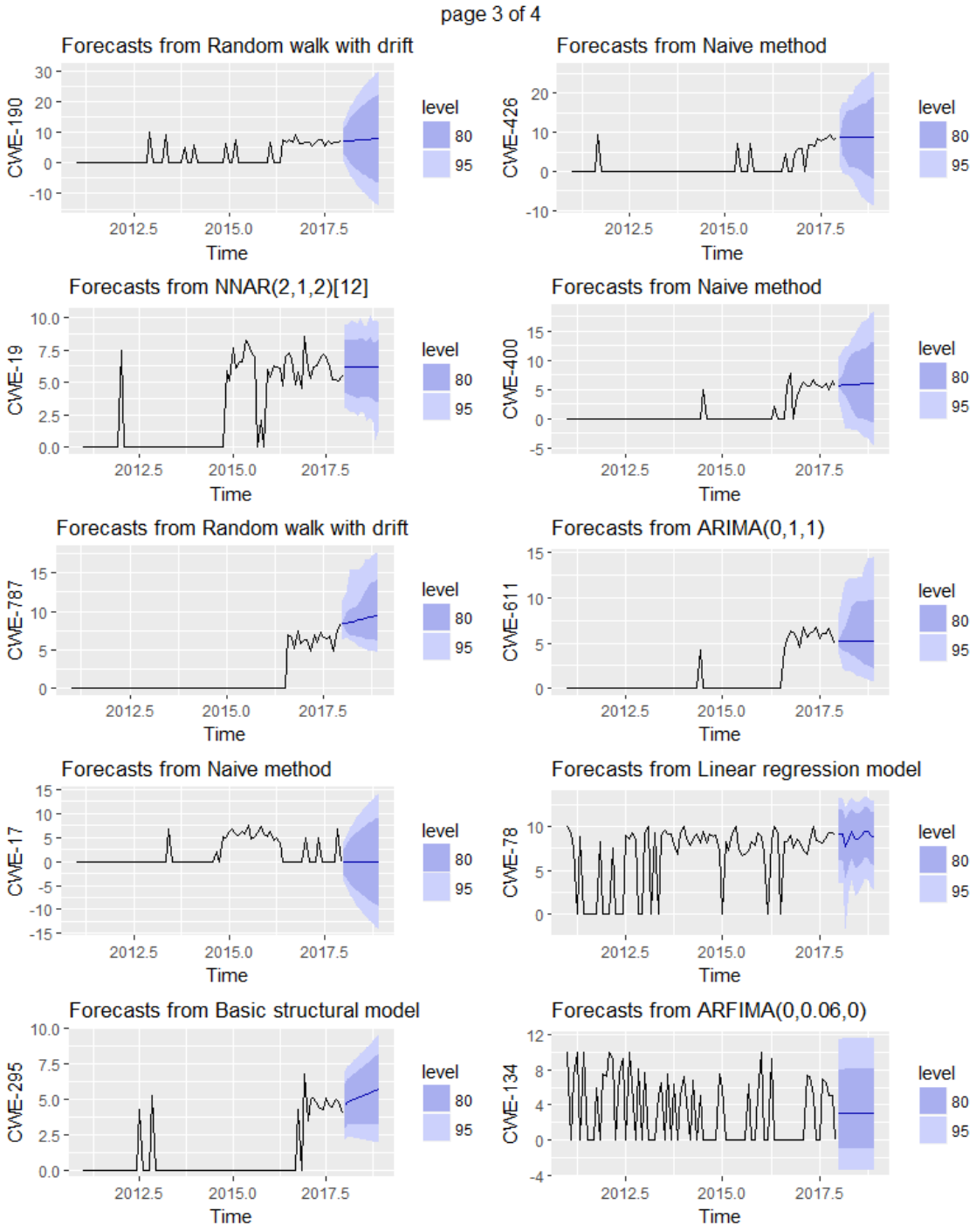

Figure 25: The 2018 Forecasts by 2017 Best Model Types (Page 3)

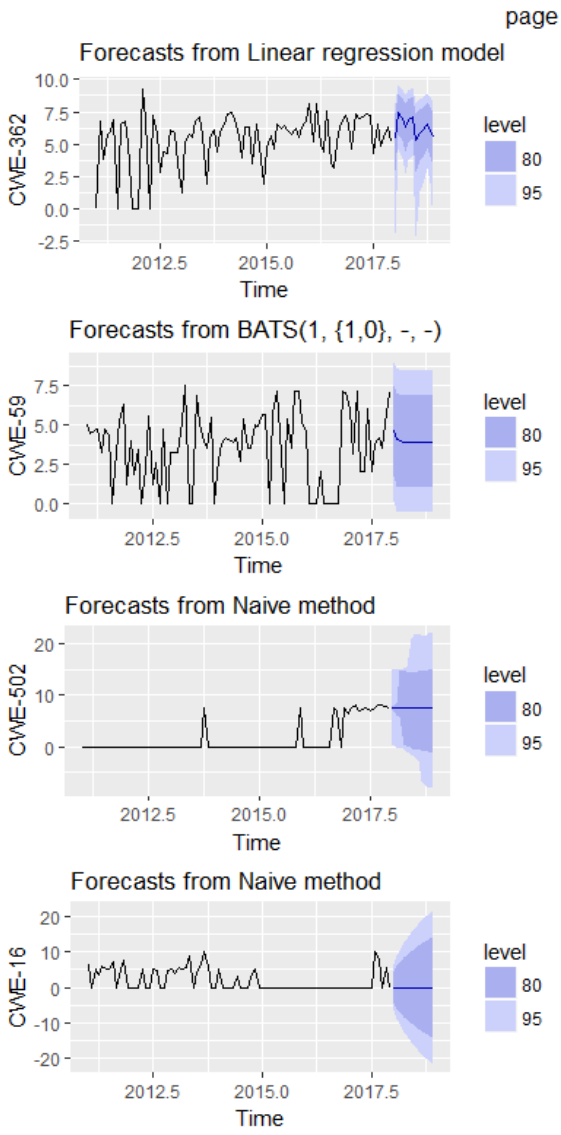

Figure 26: The 2018 Forecasts by 2017 Best Model Types (Page 4)

Table 12 presents some components of forecasts for the year 2018. The model's name that generated the forecasts is presented for each CWE ID. The table rows are grouped by these names. The summary of the mean monthly CVSS point forecasts are written into a colour-coded column. In that column, red marks "High" severity, yellow marks "Medium" severity and green means "Low" severity. If there are point forecasts belonging to multiple severity categories in 2018, then the colour of the highest one is used (for example, the case of CWE-79 in Table 12). The more specific summarising information about the point forecasts' score of the months of 2018 is given in the parentheses. For example, in the case of CWE-200, every month in 2018 was forecasted to have mean monthly CVSS 4.6 (blue line of 4th row's plot in Figure 25; 2nd row's yellow cell in Table 12).

The models' residuals were checked. The column "Mean OK" shows whether the residuals had zero mean. The column "Intervals OK" shows whether the residuals were uncorrelated and normally distributed. Sometimes the intervals were generated from bootstrapped residuals. The cells in the column "Intervals Meaning" are filled when the "Intervals OK" cells were filled with "Yes". The cells of the column "Intervals Meaning" usually show what severity score categories were covered by 95% intervals and 80% intervals. In the case of bagged ETS models, they show what severity score categories were covered by the area that is located between the minimum and maximum point forecast values of the ensemble.

Ten red cells were in Table 12. These corresponded to the following CWEs: CWE-119, CWE-20, CWE-94, CWE-77, CWE-190, CWE-787, CWE-426, CWE-78, CWE-362 and CWE-502. Three green cells in the same table were connected to CWE-17, CWE-134 and CWE-16. The rest of the cells in the column "Point Forecasts 2018" were yellow. Seven out of 21 CWEs, which had "Medium" point forecasts for 2018, had intervals surrounding the point forecasts, which included "High" mean monthly CVSS scores. These seven CWEs were CWE-89, CWE-287, CWE-254, CWE-125, CWE-476, CWE-400 and CWE-611. The total 17 expectedly "High" CWEs' previously calculated forecast accuracy results are plotted in Chapter D.1. The 2018 forecast accuracy can be eventually checked with the help of 'nvdr' as the time progresses and the unknown future becomes the known past with monthly data available to be extracted from NVD data feeds.

Table 12: 2018 Forecasts

CWE-...	Model	Mean OK	Intervals OK	Point Forecasts 2018	Intervals Meaning
119	Naïve	Yes	No	High (7.4)	-
200	Naïve	Yes	No	Medium (4.6)	-

Table 12: 2018 Forecasts Continued

CWE-...	Model	Mean OK	Intervals OK	Point Forecasts 2018	Intervals Meaning
20	Naïve	Yes	Yes	High (increasing gradually from 7.6 to 8.2)	95% intervals: Medium/High (Jan–July), Low/Medium/High (otherwise); 80% intervals: Medium/High
17	Naïve	Yes	No	Low (0)	-
426	Naïve	Yes	No	High (8.7)	-
400	Naïve	Yes	Yes (boots-trapped)	Medium (increasing gradually from 5.7 to 6.1)	95% intervals: Low/Medium/High; 80% intervals: Medium (Jan), Medium/High (Feb, Mar), Low/Medium/High (otherwise)
502	Naïve	Yes	Yes (boots-trapped)	High (7.5)	95% intervals: Low/Medium/High; 80% intervals: High (Jan), Medium/High (Feb, Mar), Low/Medium/High (otherwise)
16	Naïve	Yes	No	Low (0)	-
79	Linear regression	Yes	No	Low (Mar, May, Jun, Aug, Sep, Nov); Medium (otherwise)	-
78	Linear regression	Yes	No	High (Mar: 7.7; otherwise severity near 8 or 9)	-
362	Linear regression	Yes	Yes	High (Feb, Mar, June); Medium (otherwise)	95% intervals: Medium/High (Feb, June) Low/Medium/High (otherwise); 80% intervals: Medium/High (otherwise), Low/Medium/High (Jan, Jul, Aug, Sep, Dec)
264	ARFIMA(1,0,2)	Yes	No	Medium (decreasing gradually from 6.46 to 6.36)	-
399	ARFIMA(0,0.29,0)	Yes	No	Medium (increasing gradually from 5.8 to 6.2)	-
134	ARFIMA(0,0.06,0)	Yes	No	Low (near 3)	-
310	BSM	Yes	Yes	Medium (near 4.9)	95% intervals: Low/Medium; 80% intervals: Low/Medium
295	BSM	Yes	No	Medium (increasing gradually from 4.56 to 5.75)	-

Table 12: 2018 Forecasts Continued

CWE-...	Model	Mean OK	Intervals OK	Point Forecasts 2018	Intervals Meaning
284	BATS(1,{0,0},-,-)	Yes	No	Medium (5.8)	-
416	BATS(1,{0,0},-,-)	Yes	No	Medium (6.9)	-
77	BATS(1,{0,0},-,-)	Yes	No	High (7.7)	-
59	BATS(1,{1,0},-,-)	Yes	No	Medium (Jan, Feb); Low (otherwise	-
89	ETS(A,N,N)	Yes	Yes	Medium (near 6.99)	95% intervals: Medium/High; 80% intervals: Medium/High
189	ETS(A,N,N)	No	No	Medium (near 4.8)	-
352	Bagged ETS	Yes	Yes (ensemble limits)	Medium (near 6.5)	Medium
22	Bagged ETS	Yes	Yes (ensemble limits)	Medium (near 5.4)	Low/Medium (Mar); Medium (otherwise)
254	Bagged ETS	Yes	Yes (ensemble limits)	Medium (near 5)	Low/Medium/High
125	Bagged ETS	Yes	Yes (ensemble limits)	Medium (near 6)	Low/Medium/High (Jan, Mar, Apr, Jun, Sep); Medium/High (otherwise)
476	Bagged ETS	Yes	Yes (ensemble limits)	Medium (near 5)	Low/Medium (Jan); Medium/High (Feb, Jul, Aug, Sep, Nov); Low/Medium/High (otherwise)
94	Mean	Yes	No	High (7.03)	-
287	Mean	Yes	Yes	Medium (6.6)	95% intervals: Medium/High; 80% intervals: Medium/High
255	Drift	Yes	No	Medium (decreasing gradually from 4.7 to 4.3)	-
190	Drift	Yes	No	High (increasing gradually from 7.1 to 8.0)	-
787	Drift	Yes	No	High (increasing gradually from 8.4 to 9.5)	-
19	NNAR(2,1,2)$_{12}$	Yes	No	Medium (near 6.2)	-
611	ARIMA(0,1,1)	Yes	Yes (boots-trapped)	Medium (5.3)	95% intervals: Low/Medium/High; 80% intervals: Medium (Jan, Feb, Mar), Low/Medium/High (otherwise)

5 Conclusion

The following central research question was asked: "How to forecast monthly mean CVSS scores of a subset of CWE types?" It was asked with the motivation to help security risk management firms who use CVSS scores in their work. The forecasts could lead to improved customer risk level estimations.

An R package 'nvdr'[57] was developed. The software was then used to carry out the planned steps introduced in Chapter 4.3. The XML data from NVD as of 7 October 2017 were processed. The time series of monthly mean CVSS scores covering years 2011–2016 was created. The subset of potentially important CWE categories (Figure 4) was found. Thirteen different types of time series forecasting models were considered for each CWE.

The forecasts for 2016 were generated and the forecast accuracy was measured by using MAE, RMSE, MASE and MAPE. In the case of each CWE, the model that resulted in the lowest numbers for the most of these accuracy measures was chosen as the best model for the particular CWE (Listing 4.1 and Figures D.4, D.5, D.6). After that, the same types of models that were the best in 2016, were used with a new training set to generate monthly forecasts for 2017, which was unseen future at that time (Figures D.14, D.15, D.16). The new training set was formed by merging the previously used training set covering years 2011–2015 and the previously used test set covering the year 2016.

On 1 January 2018, the up-to-date data sets from NVD were downloaded, processed and used to check the accuracy of the previously generated unchecked 2017 forecasts (Listing 4.2). The whole up-to-date data, covering years 2011–2017 were then analysed such as the data covering years 2011–2016. Earlier, 25 potentially important CWEs were identified (Figure 4). Now, nine additional potentially important CWEs were found (Figure 22). Monthly forecasts for 34 CWEs (25 + 9) were generated for 2017 (Figures D.17, D.18, D.19, D.20). The forecast accuracy was measured again by using MAE, RMSE, MASE and MAPE. The training set covered the years 2011–2016 and the test set covered the year 2017. Based on the results, the best types of models were found for each CWE in 2017 (Listing 4.3). After that, the same types of models that were the best in 2017, were used to generate monthly forecasts for 2018, which was unseen future at that time (Figures 23, 24, 25 and Table 12).

CVSS version 2.0 score can take values from range $0 \ldots 10$. This can be divided into three categories: "Low", "Medium" and "High". Ten CWEs out of 34 potentially important CWEs will have at least some month's mean monthly CVSS score of "High" severity according to the point forecasts for 2018. These CWEs are CWE-119, CWE-20, CWE-94, CWE-77, CWE-190, CWE-787, CWE-426, CWE-

[57] `https://github.com/realerikrani/nvdr/`

78, CWE-362 and CWE-502. Seven CWEs with "Medium" point forecasts for 2018 have a possibility of "High" severity in 2018 because of the intervals surrounding the point forecasts. These CWEs are CWE-89, CWE-287, CWE-254, CWE-125, CWE-476, CWE-400 and CWE-611.

The best model types found in one year are not necessarily the best model types for the next year. It was also evident in Chapter 4.5.2 as almost all the model types were different in Listings 4.2 and 4.3. However, in the thesis, such standpoint was taken: based on the forecast accuracy calculated on the test set, the types of the best models of one year could be considered as potentially the best models to generate forecasts for the unseen next year. With that point of view, in order to forecast monthly mean CVSS scores of a subset of CWE types, one could use the developed R package 'nvdr'. That way, the steps briefly described in the conclusion and thoroughly performed in the thesis project can lead to forecasts of monthly mean CVSS scores in the unknown future by using the best available known information from the past.

5.1 Opportunities for Further Research

The research can be extended in multiple ways. The current tool, the 'nvdr' package, is accessible to users who are willing to use R programming language and the package from the command line. A graphical user interface could be one way to improve usability. 'Shiny' R package[58] can be utilised for that purpose. An advanced drag-and-drop interface similar to Alteryx Designer[59] could be helpful. Another possibility would be the conversion of R code into another language's code such as Python, which allows to build a traditional web application with interactive components. The 'nvdr' package relies on many other R packages, most importantly on 'forecast' [9]. Alternative libraries exist in the case of Python. For example, PyFlux library provides the way to forecast h-step ahead future with ARIMA models[60]. It is also possible to instead write every model's code from scratch and try to improve performance or get a more deeper understanding of the models.

Combining different forecasts could lead to improved forecast accuracy [8, ch. 12.4]. The 'nvdr' package's point forecasts from ARFIMA, ARIMA, benchmark, ETS, TBATS and linear regression model could be combined: added together and divided by the total number of the models, which is six in the current case. The obtained MAE, RMSE and MAPE could be then compared with the current MAE, RMSE and MAPE.

[58] `https://shiny.rstudio.com/`

[59] `https://www.alteryx.com/products/alteryx-designer`

[60] `http://www.pyflux.com/arima-models/`

It could be possible to use dynamic regression models [8, ch. 9] that generate forecasts based on not only the history of the time series (as the ones used in the thesis), but also additional information other than the history of the time series. First, it would be necessary to come up with ideas about what additional data might improve the forecasts of monthly mean CVSS base scores. Those additional predictor variables should show some correlation. Next, the regularly spaced time series data set of that extra data should be available. Finally, the future values of these additional predictors must be obtained for forecasting CVSS h-steps into the future.

The thesis focused on monthly data. One possibility for further research would be to change that. Weekly or yearly mean CVSS scores could be forecasted instead of forecasting monthly mean CVSS scores. The thesis selected a subset of CWEs by applying three different techniques to the data. These techniques can be changed or left out with the selection process altogether.

The XML files from NVD with CVSS version 2 were used as input. NVD also provides JSON beta release. The JSON files additionally contain CVSS version 3 base scores for the latest years [7]. As a result, the JSON files might be preferred for CVSS version 3 users.

References

[1] The MITRE Corporation. About CVE. Accessed: 2017-10-22. [Online]. Available: https://cve.mitre.org/about

[2] National Institute of Standards and Technology (NIST). NVD Frequently Asked Questions. Accessed: 2017-10-02. [Online]. Available: https://nvd.nist.gov/general/faq

[3] ——. NVD Categories. Accessed: 2017-10-23. [Online]. Available: https://nvd.nist.gov/vuln/categories

[4] The MITRE Corporation. Frequently Asked Questions. Accessed: 2017-10-23. [Online]. Available: https://cve.mitre.org/about/faqs.html#year_portion_of_cve_id

[5] A. Schuster, "Ii. on the periodicities of sunspots," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 206, no. 402-412, pp. 69–100, 1906. [Online]. Available: http://rsta.royalsocietypublishing.org/content/206/402-412/69

[6] Forum of Incident Response and Security Teams (FIRST). A Complete Guide to the Common Vulnerability Scoring System Version 2.0. Accessed: 2017-10-23. [Online]. Available: https://www.first.org/cvss/v2/guide

[7] National Institute of Standards and Technology (NIST). Vulnerability Metrics. Accessed: 2017-10-23. [Online]. Available: https://nvd.nist.gov/vuln-metrics

[8] R. J. Hyndman and G. Athanasopoulos, "Forecasting: Principles and Practice," Online textbook, Accessed: 2018-01-18. [Online]. Available: https://otexts.org/fpp2

[9] R. Hyndman, C. Bergmeir, G. Caceres, M. O'Hara-Wild, S. Razbash, and E. Wang, *forecast: Forecasting functions for time series and linear models*, 2017, r package version 8.3. [Online]. Available: http://pkg.robjhyndman.com/forecast

[10] National Institute of Standards and Technology (NIST). NVD Data Feeds. Accessed: 2017-10-02. [Online]. Available: https://nvd.nist.gov/vuln/data-feeds

[11] O. Alhazmi, Y. Malaiya, and I. Ray, "Measuring, analyzing and predicting security vulnerabilities in software systems," *Computers &*

Security, vol. 26, no. 3, pp. 219 – 228, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404806001520

[12] S. S. Murtaza, W. Khreich, A. Hamou-Lhadj, and A. B. Bener, "Mining trends and patterns of software vulnerabilities," *Journal of Systems and Software*, vol. 117, pp. 218 – 228, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121216000790

[13] P. Johnson, D. Gorton, R. Lagerström, and M. Ekstedt, "Time between vulnerability disclosures: A measure of software product vulnerability," *Computers & Security*, vol. 62, pp. 278 – 295, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404816300955

[14] Y. Roumani, J. K. Nwankpa, and Y. F. Roumani, "Time series modeling of vulnerabilities," *Computers & Security*, vol. 51, pp. 32 – 40, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404815000358

[15] S. Zhang, D. Caragea, and X. Ou, *An Empirical Study on Using the National Vulnerability Database to Predict Software Vulnerabilities*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 217–231. [Online]. Available: https://doi.org/10.1007/978-3-642-23088-2_15

[16] M. S. Ahmed, E. Al-Shaer, M. Taibah, and L. Khan, "Objective risk evaluation for automated security management," *Journal of Network and Systems Management*, vol. 19, no. 3, pp. 343–366, Sep 2011. [Online]. Available: https://doi.org/10.1007/s10922-010-9177-6

[17] J. Geng, D. Ye, and P. Luo, *Predicting Severity of Software Vulnerability Based on Grey System Theory*. Cham: Springer International Publishing, 2015, pp. 143–152. [Online]. Available: https://doi.org/10.1007/978-3-319-27161-3_13

[18] D. Last, "Using historical software vulnerability data to forecast future vulnerabilities," in *2015 Resilience Week (RWS)*, Aug 2015, pp. 1–7.

[19] ——, "Consensus forecasting of zero-day vulnerabilities for network security," in *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*, Oct 2016, pp. 1–8.

[20] M. Tang, M. Alazab, and Y. Luo, "Big data for cybersecurity: Vulnerability disclosure trends and dependencies," *IEEE Transactions on Big Data*, vol. PP, no. 99, pp. 1–1, 2017.

[21] D. Mellado, C. Blanco, L. E. Sánchez, and E. Fernández-Medina, "A systematic review of security requirements engineering," *Computer Standards & Interfaces*, vol. 32, no. 4, pp. 153 – 165, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920548910000255

[22] R. J. Hyndman, A. Koehler, K. Ord, and R. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*, 1st ed., ser. Springer Series in Statistics. Springer-Verlag Berlin Heidelberg, 2008.

[23] R. J. Hyndman, "forecast 8.0," Hyndsight blog, Accessed: 2017-12-10. [Online]. Available: https://robjhyndman.com/hyndsight/forecast8/

[24] ——, "Forecasting: Principles and Practice: 9. State space models," Slides, Accessed: 2017-12-02. [Online]. Available: https://robjhyndman.com/uwafiles/9-StateSpaceModels.pdf

[25] C. Bergmeir, R. Hyndman, and J. Benítez, "Bagging exponential smoothing methods using stl decomposition and box–cox transformation," *International Journal of Forecasting*, vol. 32, no. 2, pp. 303 – 312, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207015001120

[26] A. M. D. Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513–1527, 2011. [Online]. Available: https://doi.org/10.1198/jasa.2011.tm09771

[27] B. R. W.N. Venables, *Modern Applied Statistics with S*, 4th ed., ser. Statistics and Computing. Springer, 2002.

[28] J. Janulevičius, "Method of information security risk analysis for virtualized systems," PhD dissertation, Vilnius Gediminas Technical University, 2016. [Online]. Available: http://dspace.vgtu.lt/handle/1/3026

[29] Forum of Incident Response and Security Teams (FIRST). Vulnerability Database Catalog. Accessed: 2017-10-01. [Online]. Available: https://www.first.org/global/sigs/vrdx/vdb-catalog

[30] Carnegie Mellon University. Vulnerability Notes Database. Accessed: 2017-10-02. [Online]. Available: https://www.kb.cert.org/vuls

[31] R. J. Hyndman, "Forecasting using R: 2. The forecaster's toolbox," Slides, Accessed: 2017-11-13. [Online]. Available: https://robjhyndman.com/talks/RevolutionR/2-Toolbox.pdf

[32] H. Wickham, "Advanced R," Online textbook, Accessed: 2018-01-21. [Online]. Available: https://adv-r.hadley.nz/

Appendix

A Appendix for Introduction

Table A.1: CVSS V2 Base Metric Group [6, ch. 3.2.1]

Metric	Metric Value	Score
AV	Local (L)	0.395
AV	Adjacent Network (A)	0.646
AV	Network (N)	1.0
AC	High (H)	0.35
AC	Medium (M)	0.61
AC	Low (L)	0.71
Au	Multiple (M)	0.45
Au	Single (S)	0.56
Au	None (N)	0.704
C	None (N)	0.0
C	Partial (P)	0.275
C	Complete (C)	0.660
I	None (N)	0.0
I	Partial (P)	0.275
I	Complete (C)	0.660
A	None (N)	0.0
A	Partial (P)	0.275
A	Complete (C)	0.660

B Appendix for Related Work

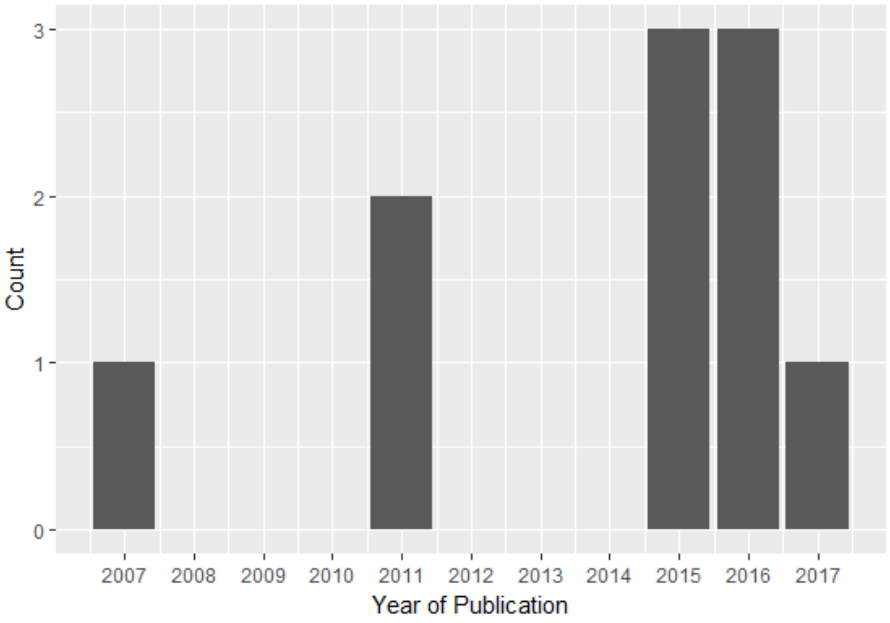

Figure B.1: How Many Studies Were Published in What Year?

C Appendix for Background

Table C.1: Forecasts by Exponential Smoothing Methods [8, ch. 7.4, Table 7.7]

(T,S)	Point Forecast $\hat{y}_{t+h\|t}$	Slope b_t	Level ℓ_t	Season s_t
(N,N)	ℓ_t		αy_t $+(1-\alpha)\ell_{t-1}$	
(N,A)	ℓ_t $+s_{t-m+h_m^+}$		$\alpha(y_t - s_{t-m})$ $+(1-\alpha)\ell_{t-1}$	$\gamma(y_t - \ell_{t-1})$ $+(1-\gamma)s_{t-m}$
(N,M)	ℓ_t $\times s_{t-m+h_m^+}$		$\alpha\left(\frac{y_t}{s_{t-m}}\right)$ $+(1-\alpha)\ell_{t-1}$	$\gamma\left(\frac{y_t}{\ell_{t-1}}\right)$ $+(1-\gamma)s_{t-m}$
(A,N)	$\ell_t + hb_t$	$\beta^*(\ell_t - \ell_{t-1})$ $+(1-\beta^*)b_{t-1}$	αy_t $+(1-\alpha)(\ell_{t-1}$ $+b_{t-1})$	
(A,A)	$\ell_t + hb_t$ $+s_{t-m+h_m^+}$	$\beta^*(\ell_t - \ell_{t-1})$ $+(1-\beta^*)b_{t-1}$	$\alpha(y_t - s_{t-m})$ $+(1-\alpha)(\ell_{t-1}$ $+b_{t-1})$	$\gamma(y_t - \ell_{t-1}$ $-b_{t-1})$ $+(1-\gamma)s_{t-m}$
(A,M)	$(\ell_t + hb_t)$ $\times s_{t-m+h_m^+}$	$\beta^*(\ell_t - \ell_{t-1})$ $+(1-\beta^*)b_{t-1}$	$\alpha\left(\frac{y_t}{s_{t-m}}\right)$ $+(1-\alpha)(\ell_{t-1}$ $+b_{t-1})$	$\gamma\left(\frac{y_t}{\ell_{t-1}+b_{t-1}}\right)$ $+(1-\gamma)s_{t-m}$
(A_d,N)	$\ell_t + \phi_h b_t$	$\beta^*(\ell_t - \ell_{t-1})$ $+(1-\beta^*)\phi b_{t-1}$	αy_t $+(1-\alpha)(\ell_{t-1}$ $+\phi b_{t-1})$	

Table C.1: Forecasts by Exp. Smoothing Methods Continued [8, ch. 7.4, Table 7.7]

(T,S)	Point Forecast $\hat{y}_{t+h\|t}$	Slope b_t	Level ℓ_t	Season s_t
(A_d,A)	$\ell_t + \phi_h b_t$ $+s_{t-m+h_m^+}$	$\beta^*(\ell_t - \ell_{t-1})$ $+(1-\beta^*)\phi b_{t-1}$	$\alpha(y_t - s_{t-m})$ $+(1-\alpha)(\ell_{t-1}$ $+\phi b_{t-1})$	$\gamma(y_t - \ell_{t-1}$ $-\phi b_{t-1})$ $+(1-\gamma)s_{t-m}$
(A_d,M)	$(\ell_t + \phi_h b_t)$ $\times s_{t-m+h_m^+}$	$\beta^*(\ell_t - \ell_{t-1})$ $+(1-\beta^*)\phi b_{t-1}$	$\alpha\left(\frac{y_t}{s_{t-m}}\right)$ $+(1-\alpha)(\ell_{t-1}$ $+\phi b_{t-1})$	$\gamma\left(\frac{y_t}{\ell_{t-1}+\phi b_{t-1}}\right)$ $+(1-\gamma)s_{t-m}$

Table C.2: ETS Models [8, ch. 7.5, Table 7.8]

(E,T,S)	Observation y_t	Slope b_t	Level ℓ_t	Season s_t
(A,N,N)	$\ell_{t-1} + \varepsilon_t$		$\ell_{t-1} + \alpha\varepsilon_t$	
(M,N,N)	$\ell_{t-1}(1+\varepsilon_t)$		$\ell_{t-1}(1+\alpha\varepsilon_t)$	
(A,N,A)	ℓ_{t-1} $+s_{t-m}$ $+\varepsilon_t$		$\ell_{t-1} + \alpha\varepsilon_t$	$s_{t-m} + \gamma\varepsilon_t$
(M,N,A)	$(\ell_{t-1}$ $+s_{t-m})$ $\times(1+\varepsilon_t)$		ℓ_{t-1} $+\alpha(\ell_{t-1}$ $+s_{t-m})\varepsilon_t$	s_{t-m} $+\gamma(\ell_{t-1}$ $+s_{t-m})\varepsilon_t$
(A,N,M)	ℓ_{t-1} $\times s_{t-m}$ $+\varepsilon_t$		ℓ_{t-1} $+\frac{\alpha\varepsilon}{s_{t-m}}$	s_{t-m} $+\frac{\gamma\varepsilon_t}{\ell_{t-1}}$
(M,N,M)	ℓ_{t-1} $\times s_{t-m}$ $\times(1+\varepsilon_t)$		ℓ_{t-1} $\times(1+\alpha\varepsilon_t)$	s_{t-m} $\times(1+\gamma\varepsilon_t)$

Table C.2: ETS Models Continued [8, ch. 7.5, Table 7.8]

(E,T,S)	Observation y_t	Slope b_t	Level ℓ_t	Season s_t
(A,A,N)	ℓ_{t-1} $+b_{t-1}$ $+\varepsilon_t$	b_{t-1} $+\beta\varepsilon_t$	ℓ_{t-1} $+b_{t-1}$ $+\alpha\varepsilon_t$	
(M,A,N)	$(\ell_{t-1}$ $+b_{t-1})$ $\times(1+\varepsilon_t)$	b_{t-1} $+\beta(\ell_{t-1}$ $+b_{t-1})\varepsilon_t$	$(\ell_{t-1}$ $+b_{t-1})$ $\times(1+\alpha\varepsilon_t)$	
(A,A,A)	ℓ_{t-1} $+b_{t-1}$ $+s_{t-m}$ $+\varepsilon_t$	b_{t-1} $+\beta\varepsilon_t$	ℓ_{t-1} $+b_{t-1}$ $+\alpha\varepsilon_t$	$s_{t-m}+\gamma\varepsilon_t$
(M,A,A)	$(\ell_{t-1}$ $+b_{t-1}$ $+s_{t-m})$ $\times(1+\varepsilon_t)$	b_{t-1} $+\beta(\ell_{t-1}$ $+b_{t-1}$ $+s_{t-m})\varepsilon_t$	ℓ_{t-1} $+b_{t-1}$ $+\alpha(\ell_{t-1}$ $+b_{t-1}$ $+s_{t-m})\varepsilon_t$	s_{t-m} $+\gamma(\ell_{t-1}$ $+b_{t-1}$ $+s_{t-m})\varepsilon_t$
(A,A,M)	$(\ell_{t-1}$ $+b_{t-1})$ $\times s_{t-m}$ $+\varepsilon_t$	b_{t-1} $+\dfrac{\beta\varepsilon_t}{s_{t-m}}$	ℓ_{t-1} $+b_{t-1}$ $+\dfrac{\alpha\varepsilon_t}{s_{t-m}}$	s_{t-m} $+\dfrac{\gamma\varepsilon_t}{\ell_{t-1}+b_{t-1}}$
(M,A,M)	$(\ell_{t-1}$ $+b_{t-1})$ $\times s_{t-m}$ $\times(1+\varepsilon_t)$	b_{t-1} $+\beta(\ell_{t-1}$ $+b_{t-1})\varepsilon_t$	$(\ell_{t-1}$ $+b_{t-1})$ $\times(1+\alpha\varepsilon_t)$	s_{t-m} $\times(1+\gamma\varepsilon_t)$
(A,A_d,N)	ℓ_{t-1} $+\phi b_{t-1}$ $+\varepsilon_t$	ϕb_{t-1} $+\beta\varepsilon_t$	ℓ_{t-1} $+\phi b_{t-1}$ $+\alpha\varepsilon_t$	
(M,A_d,N)	$(\ell_{t-1}$ $+\phi b_{t-1})$ $\times(1+\varepsilon_t)$	ϕb_{t-1} $+\beta(\ell_{t-1}$ $+\phi b_{t-1})\varepsilon_t$	$(\ell_{t-1}$ $+\phi b_{t-1})$ $\times(1+\alpha\varepsilon_t)$	

Table C.2: ETS Models Continued [8, ch. 7.5, Table 7.8]

(E,T,S)	Observation y_t	Slope b_t	Level ℓ_t	Season s_t
(A,A_d,A)	ℓ_{t-1} $+\phi b_{t-1}$ $+s_{t-m}$ $+\varepsilon_t$	ϕb_{t-1} $+\beta\varepsilon_t$	ℓ_{t-1} $+\phi b_{t-1}$ $+\alpha\varepsilon_t$	$s_{t-m}+\gamma\varepsilon_t$
(M,A_d,A)	$(\ell_{t-1}$ $+\phi b_{t-1}$ $+s_{t-m})$ $\times(1+\varepsilon_t)$	ϕb_{t-1} $+\beta(\ell_{t-1}$ $+\phi b_{t-1}$ $+s_{t-m})\varepsilon_t$	ℓ_{t-1} $+\phi b_{t-1}$ $+\alpha(\ell_{t-1}$ $+\phi b_{t-1}$ $+s_{t-m})\varepsilon_t$	s_{t-m} $+\gamma(\ell_{t-1}$ $+\phi b_{t-1}$ $+s_{t-m})\varepsilon_t$
(A,A_d,M)	$(\ell_{t-1}$ $+\phi b_{t-1})$ $\times s_{t-m}$ $+\varepsilon_t$	ϕb_{t-1} $+\frac{\beta\varepsilon_t}{s_{t-m}}$	ℓ_{t-1} $+\phi b_{t-1}$ $+\frac{\alpha\varepsilon_t}{s_{t-m}}$	s_{t-m} $+\frac{\gamma\varepsilon_t}{\ell_{t-1}+\phi b_{t-1}}$
(M,A_d,M)	$(\ell_{t-1}$ $+\phi b_{t-1})$ $\times s_{t-m}$ $\times(1+\varepsilon_t)$	ϕb_{t-1} $+\beta(\ell_{t-1}$ $+\phi b_{t-1})\varepsilon_t$	$(\ell_{t-1}$ $+\phi b_{t-1})$ $\times(1+\alpha\varepsilon_t)$	s_{t-m} $\times(1+\gamma\varepsilon_t)$

D Appendix for Contribution

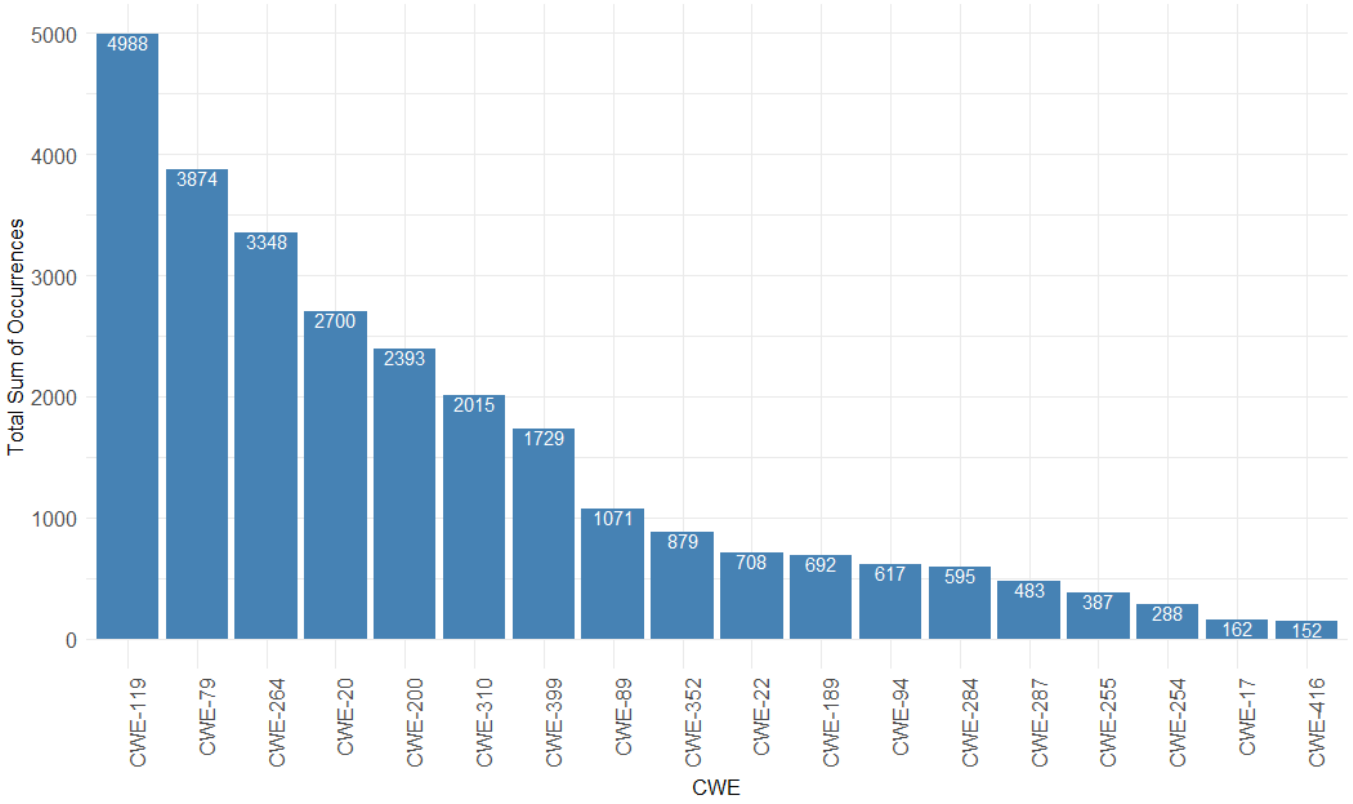

Figure D.1: CWEs Occurring at Least 100 Times in Any of the Years 2011 – 2016

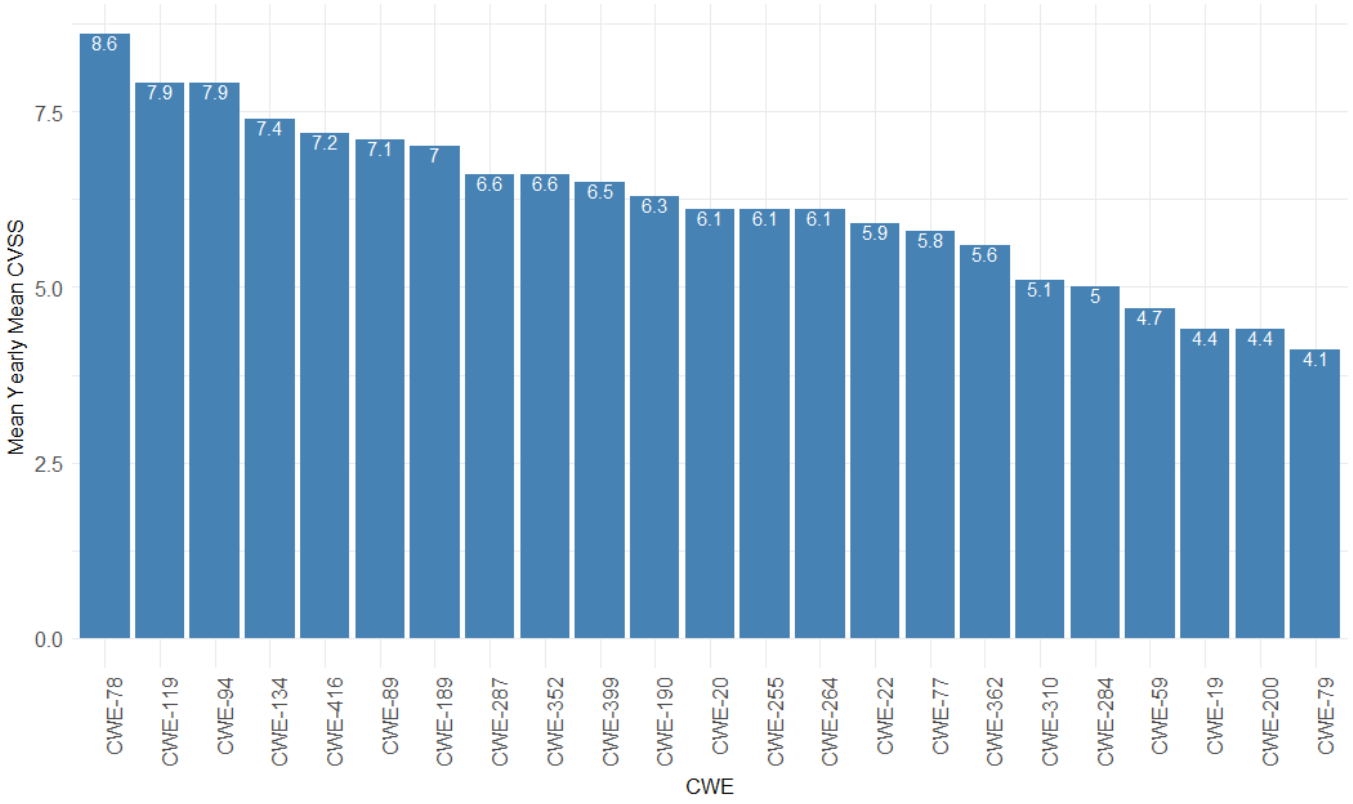

Figure D.2: CWEs with Mean Yearly Mean CVSS above 4.0 During 2011 – 2016

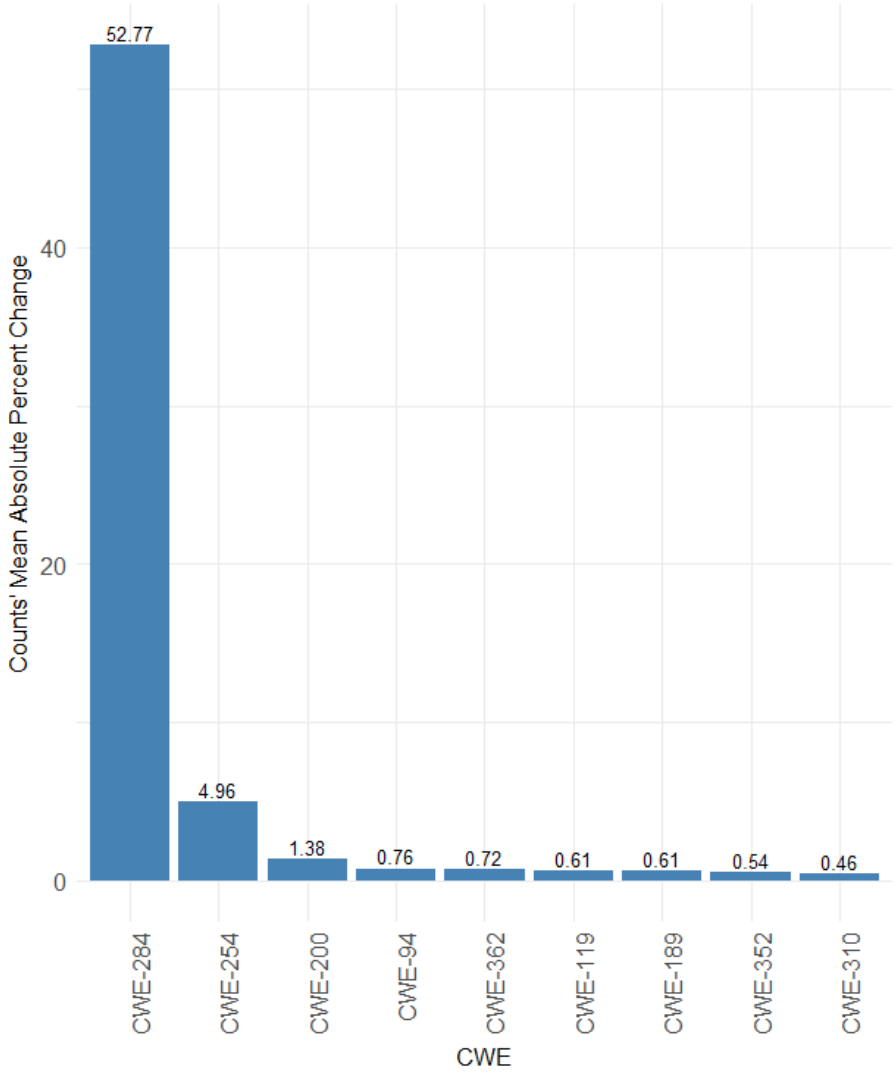

Figure D.3: Changing CWEs During 2011 – 2016

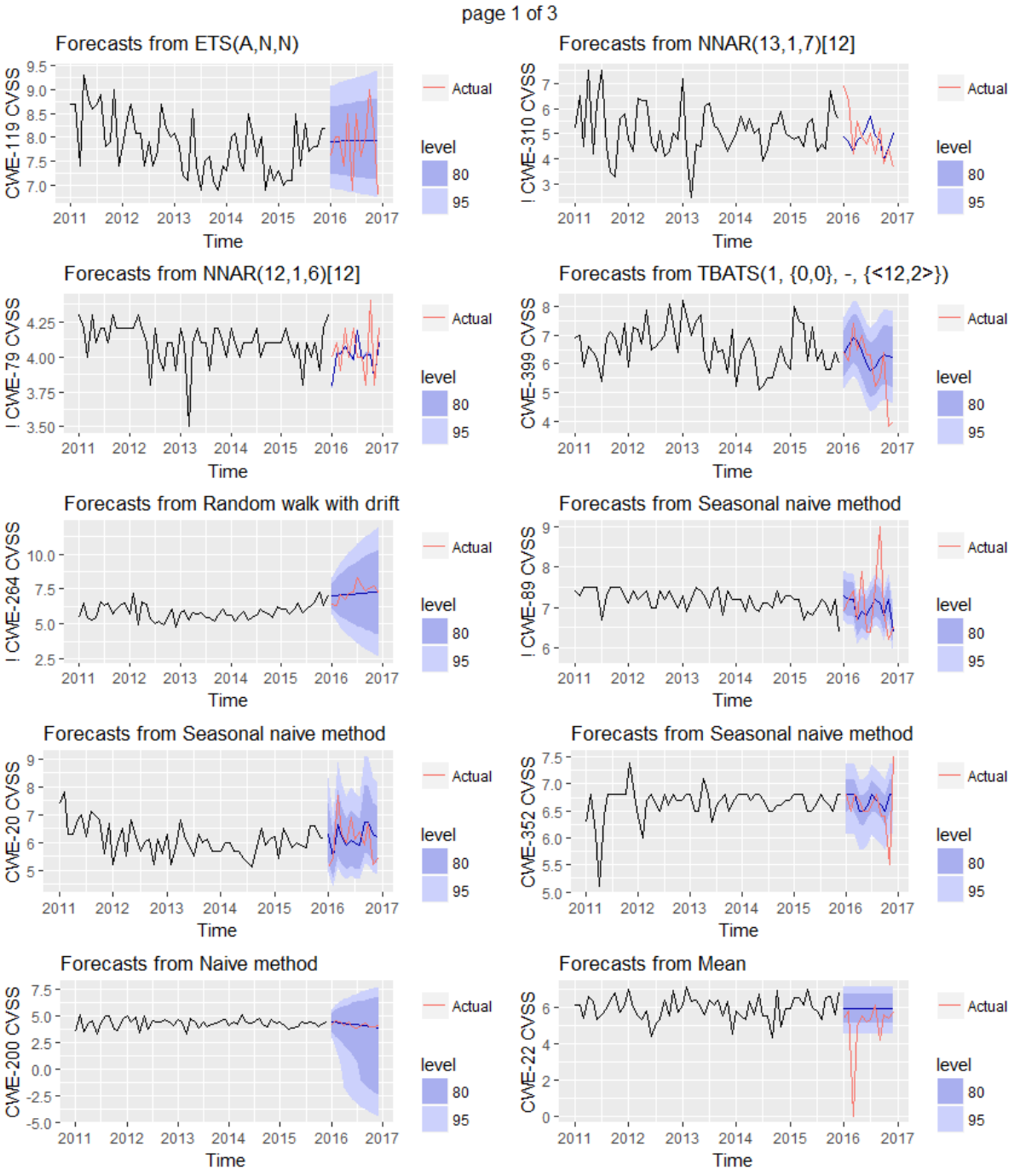

Figure D.4: The 2016 Forecasts (Page 1)

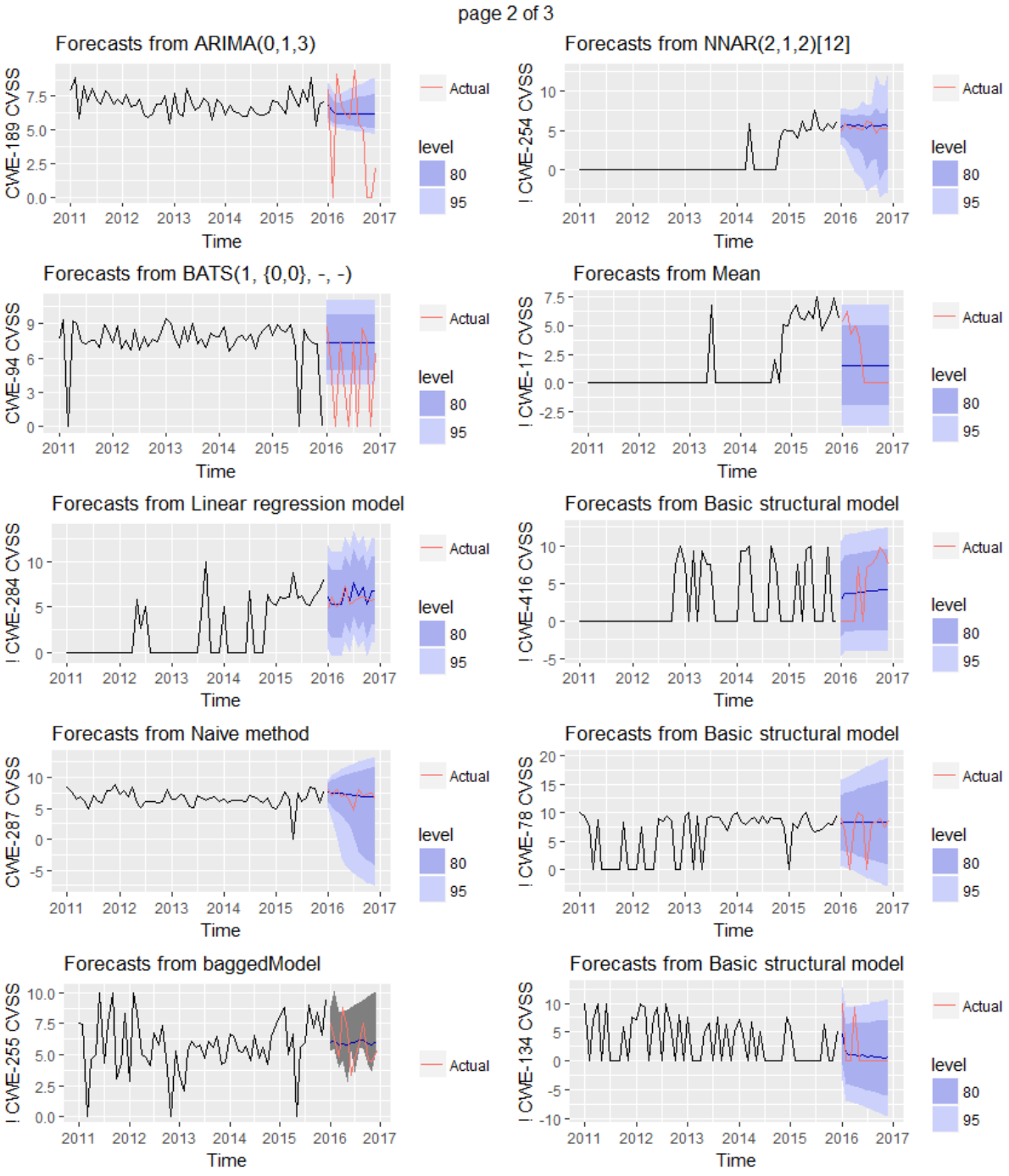

Figure D.5: The 2016 Forecasts (Page 2)

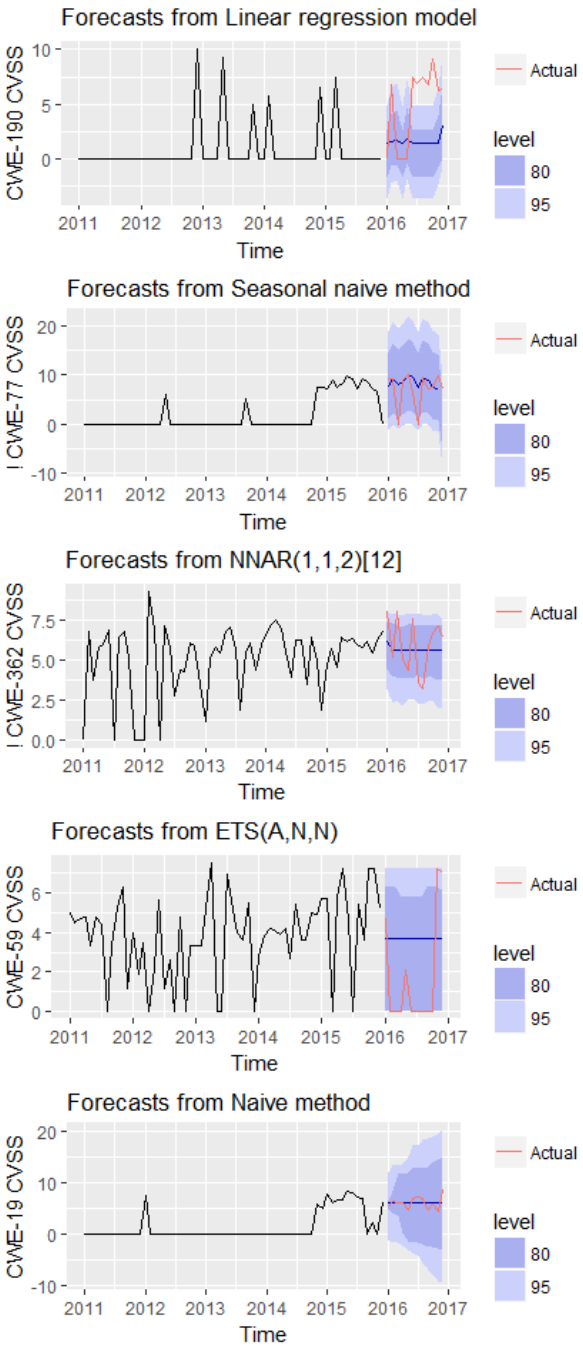

Figure D.6: The 2016 Forecasts (Page 3)

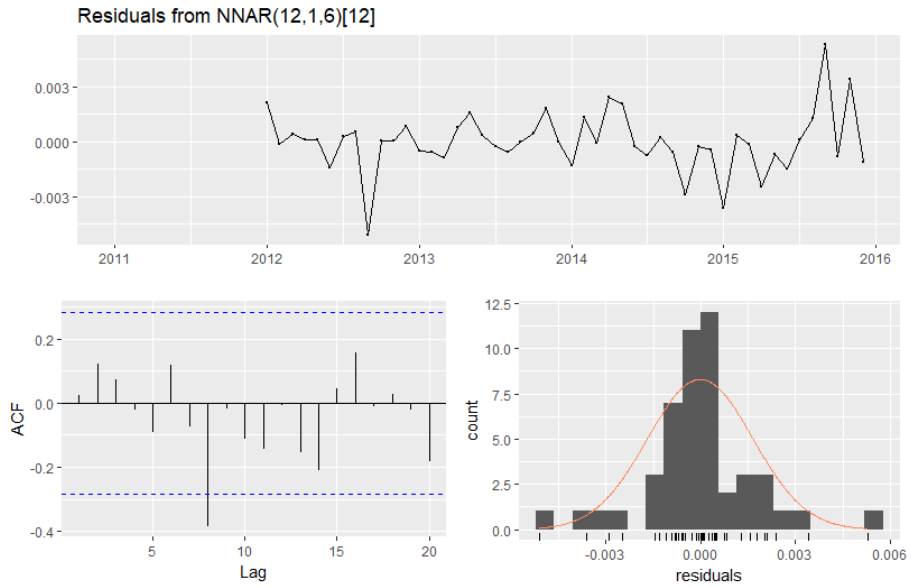

Figure D.7: CWE-79 Model's Residuals Diagnostics 2011–2015

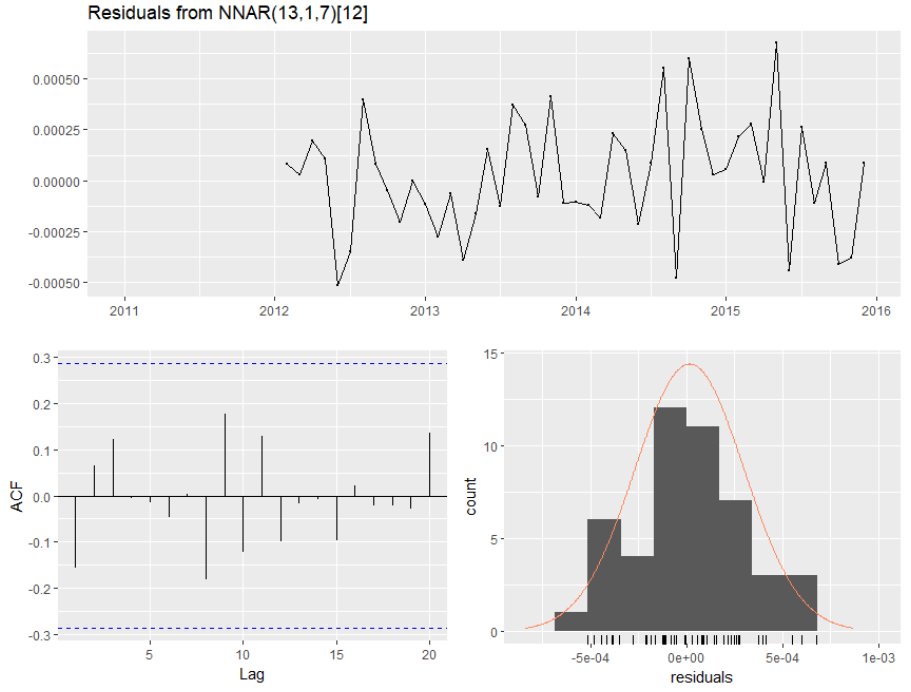

Figure D.8: CWE-310 Model's Residuals Diagnostics 2011–2015

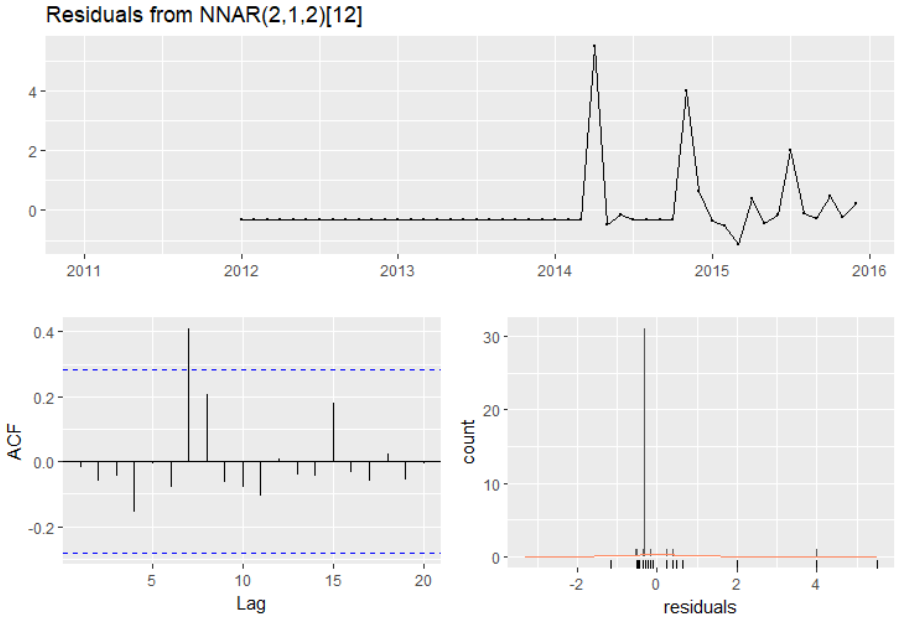

Figure D.9: CWE-254 Model's Residuals Diagnostics 2011–2015

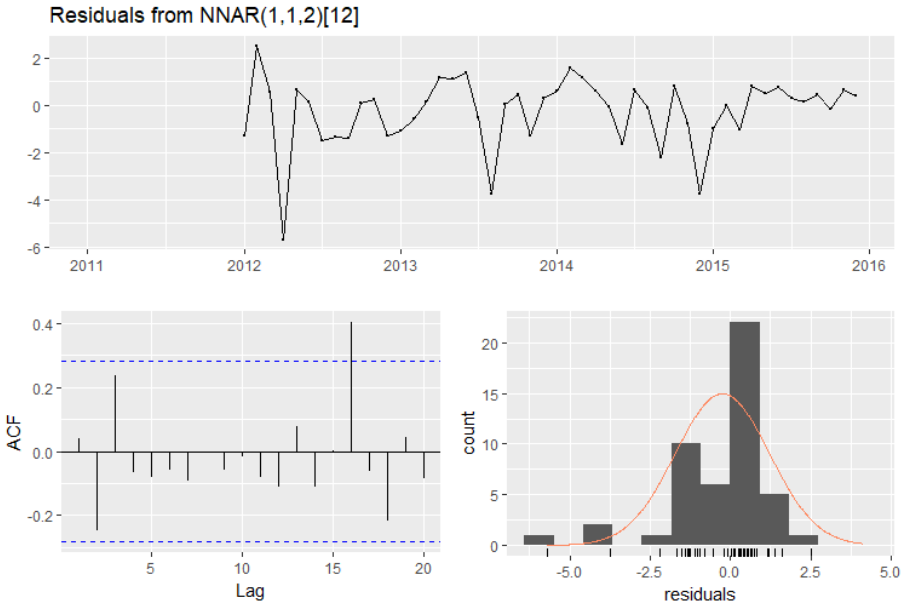

Figure D.10: CWE-362 Model's Residuals Diagnostics 2011–2015

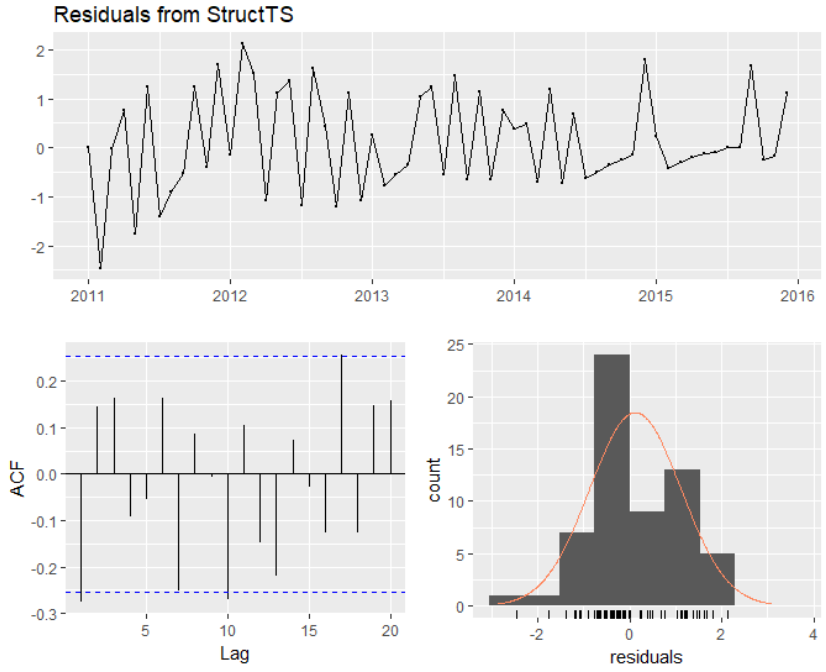

Figure D.11: CWE-134 Model's Residuals Diagnostics 2011–2015

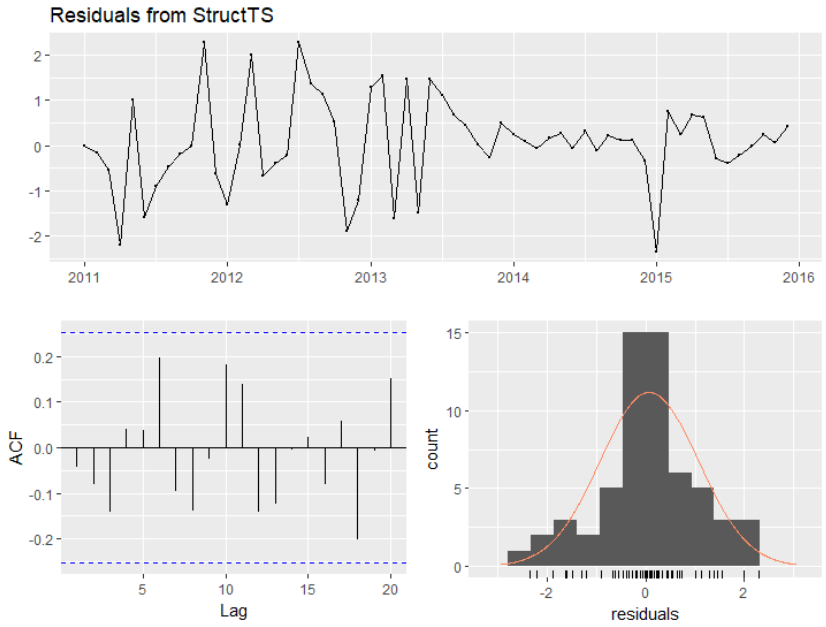

Figure D.12: CWE-78 Model's Residuals Diagnostics 2011–2015

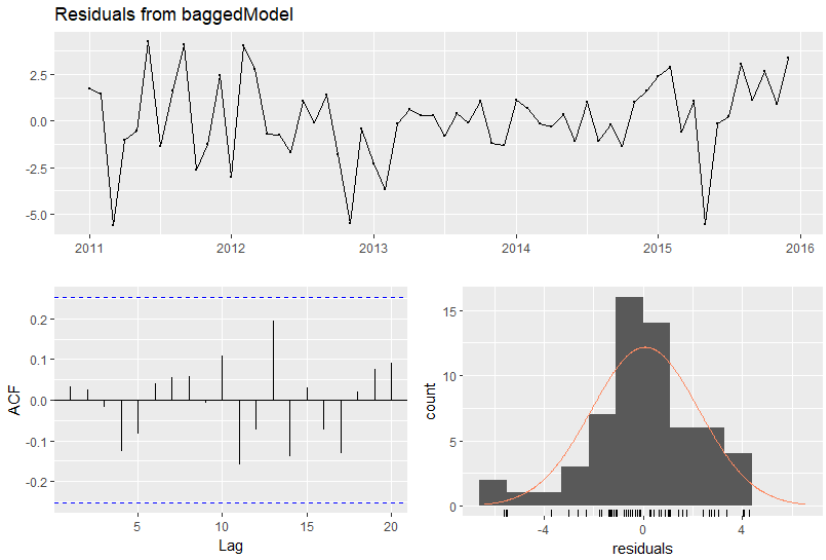

Figure D.13: CWE-255 Model's Residuals Diagnostics 2011–2015

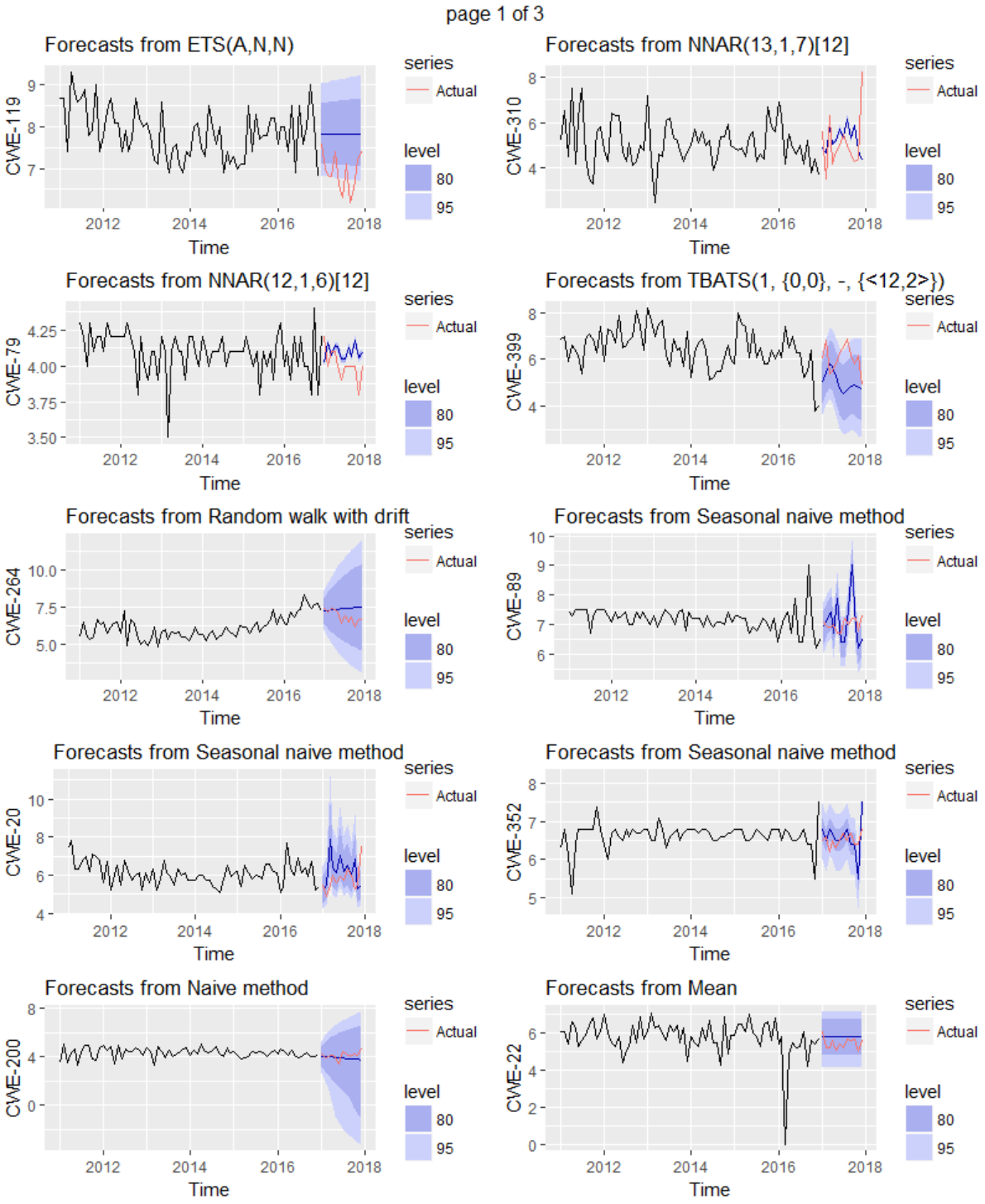

Figure D.14: The 2017 Forecasts by 2016 Best Model Types (Page 1)

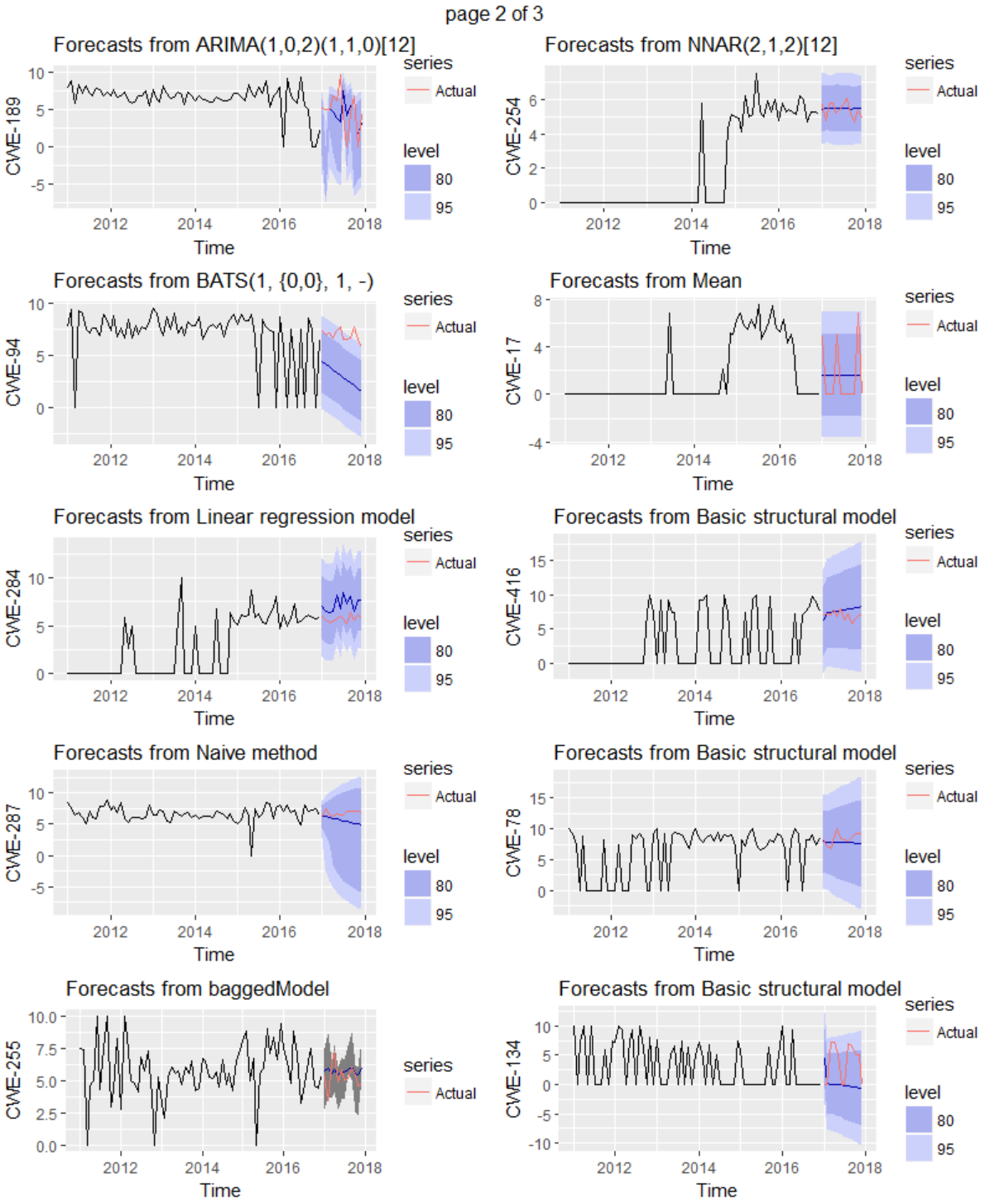

Figure D.15: The 2017 Forecasts by 2016 Best Model Types (Page 2)

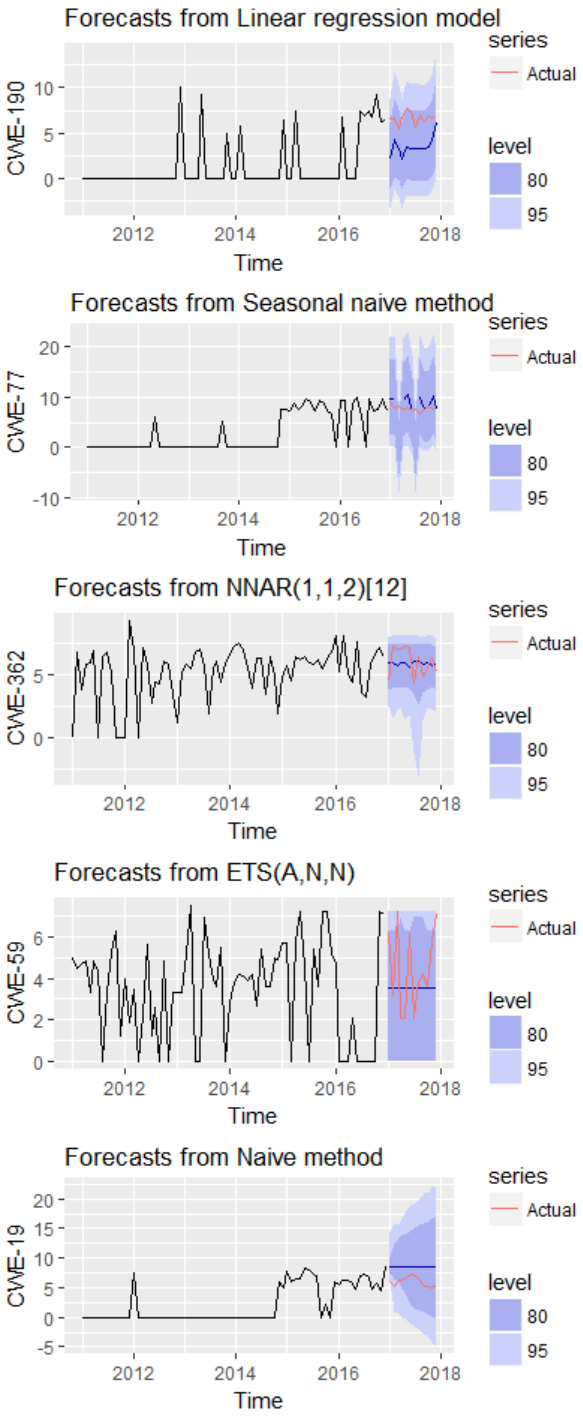

Figure D.16: The 2017 Forecasts by 2016 Best Model Types (Page 3)

Listing D.1: Values of Listing 4.1 subtracted from values of Listing 4.2

	cwe	MAE	RMSE	MAPE	MASE
1:	CWE–119	0.4259	0.3944	7.3332	0.6731
2:	CWE–79	−0.0247	−0.0393	−0.5160	−0.2766
3:	CWE–264	0.1751	0.1980	3.2398	0.2016
4:	CWE–20	0.2596	0.3932	4.5410	0.4379
5:	CWE–200	0.1377	0.1931	3.3856	0.3320
6:	CWE–310	0.2984	0.4334	5.2828	0.2489
7:	CWE–399	0.3251	0.1915	1.1317	0.3300
8:	CWE–89	0.1333	0.0431	2.2731	0.1159
9:	CWE–352	0.0416	−0.0378	0.4753	0.1314
10:	CWE–22	−0.6424	−1.3905	−Inf	−0.8684
11:	CWE–189	−0.4234	−0.6852	NaN	−1.3110
12:	CWE–94	0.7617	−0.3829	−Inf	−0.3926
13:	CWE–284	0.7164	0.6931	12.2609	0.3436
14:	CWE–287	0.5283	0.4612	6.7230	0.3614
15:	CWE–255	−0.5318	−0.5141	−8.2910	−0.2361
16:	CWE–254	−0.0721	−0.0745	−1.3538	−0.0187
17:	CWE–17	−0.1370	−0.1201	NaN	−0.3147
18:	CWE–416	−2.9553	−2.8383	−Inf	−0.7269
19:	CWE–78	−0.9905	−2.3557	−Inf	−0.2404
20:	CWE–134	2.3300	2.1213	NaN	0.7134
21:	CWE–190	−1.0842	−1.3973	−Inf	−1.2153
22:	CWE–77	−0.9512	−1.8328	−Inf	−0.3590
23:	CWE–362	−0.3533	−0.4553	−8.4893	−0.1389
24:	CWE–59	−1.5680	−1.3104	−Inf	−0.6858
25:	CWE–19	1.5833	1.4629	28.3606	0.8164
	cwe	MAE	RMSE	MAPE	MASE

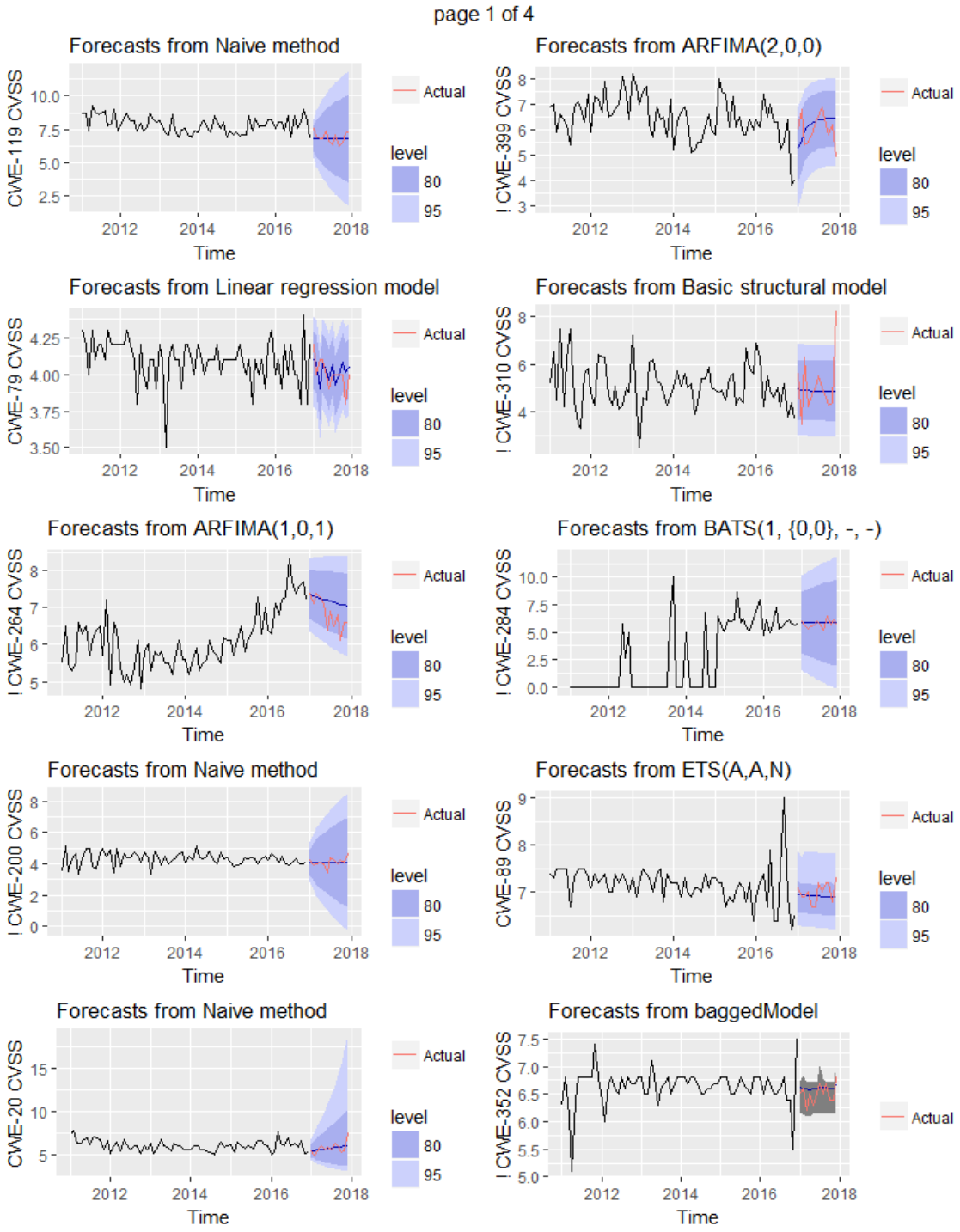

Figure D.17: The 2017 Forecasts (Page 1)

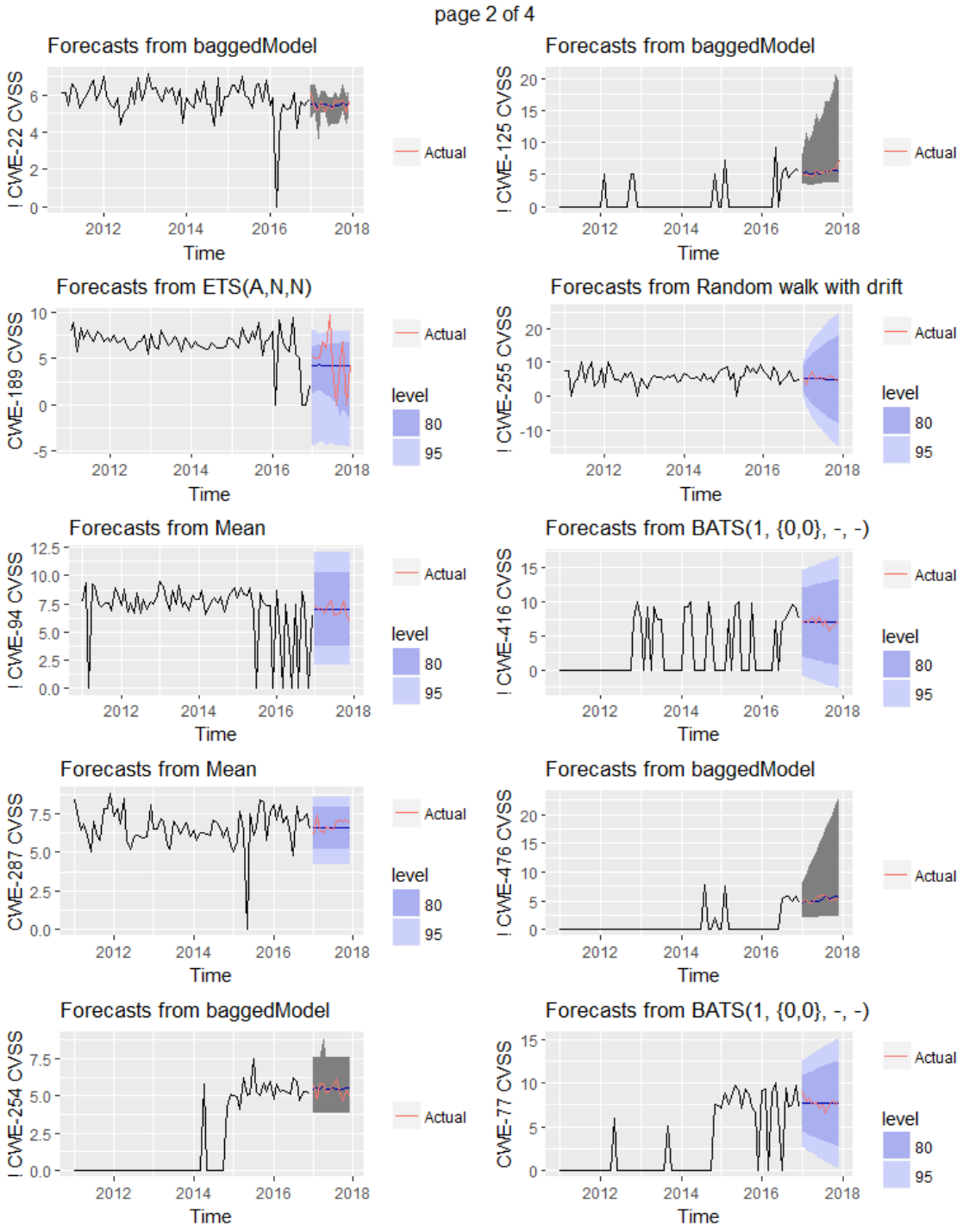

Figure D.18: The 2017 Forecasts (Page 2)

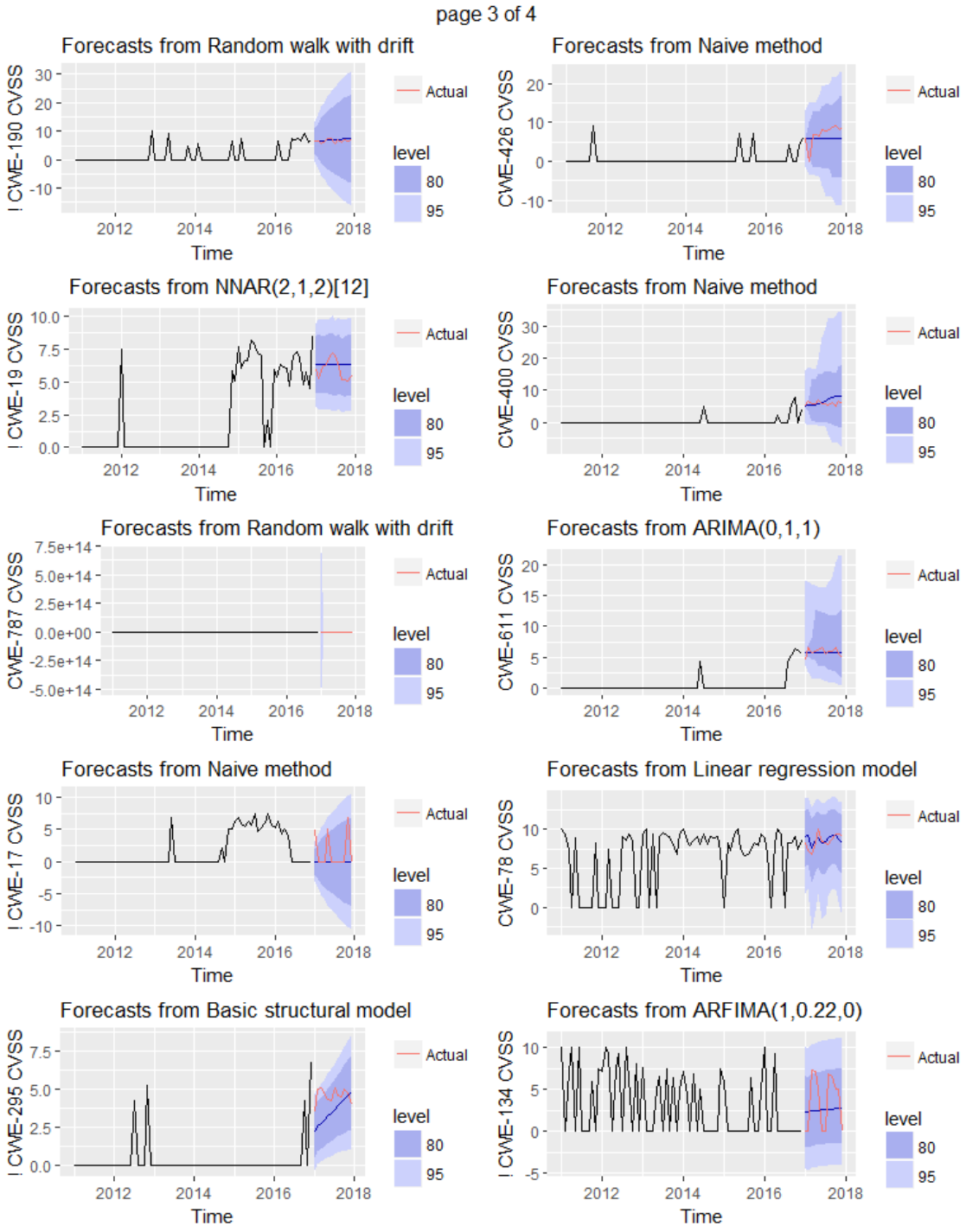

Figure D.19: The 2017 Forecasts (Page 3)

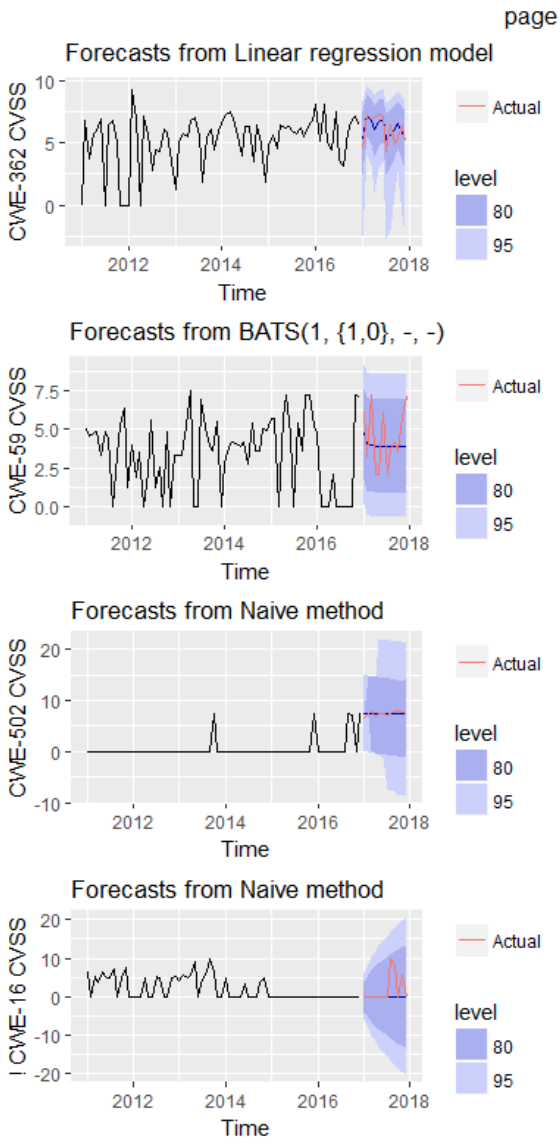

Figure D.20: The 2017 Forecasts (Page 4)

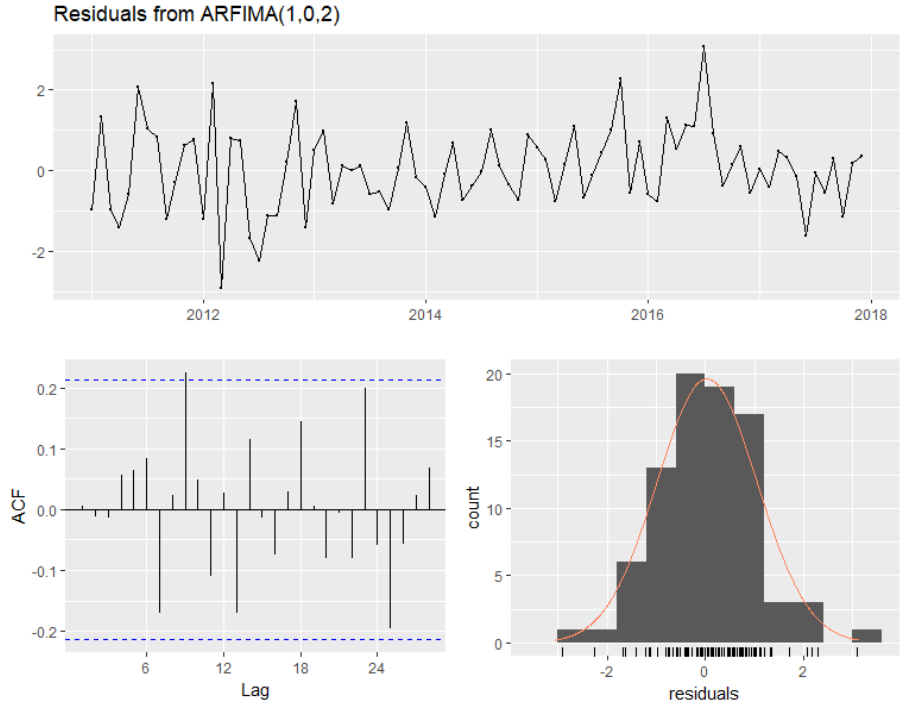

Figure D.21: CWE-264 Model's Residuals Diagnostics 2011–2017

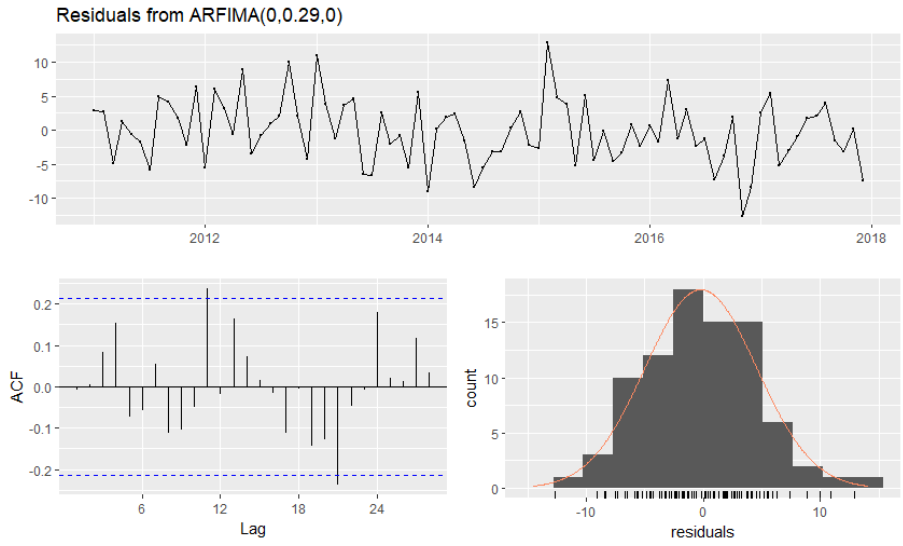

Figure D.22: CWE-399 Model's Residuals Diagnostics 2011–2017

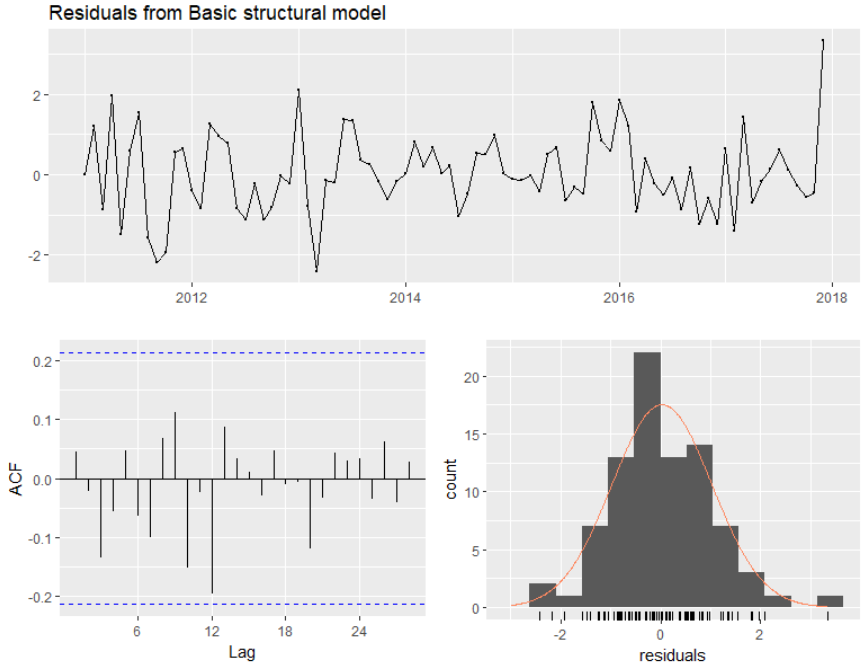

Figure D.23: CWE-310 Model's Residuals Diagnostics 2011–2017

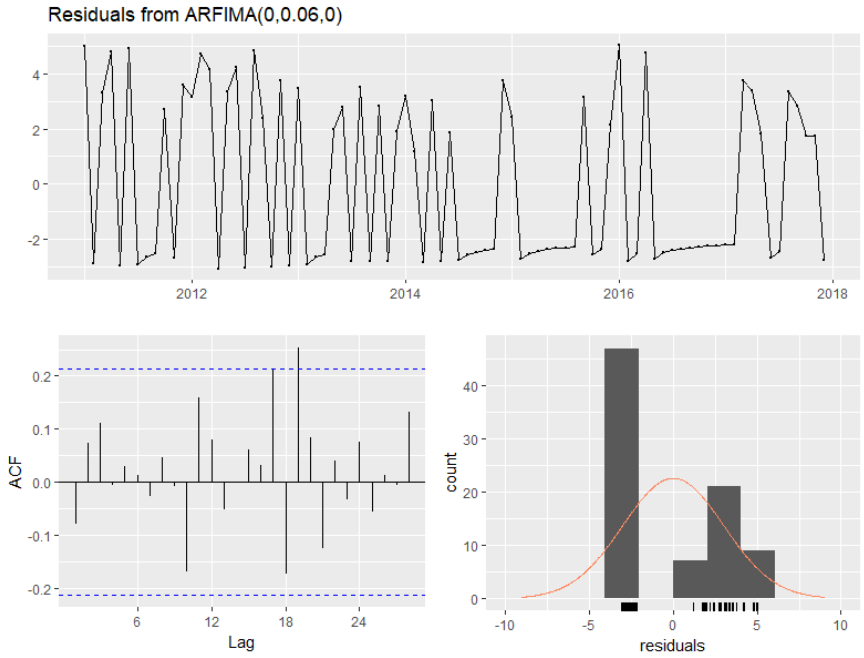

Figure D.24: CWE-134 Model's Residuals Diagnostics 2011–2017

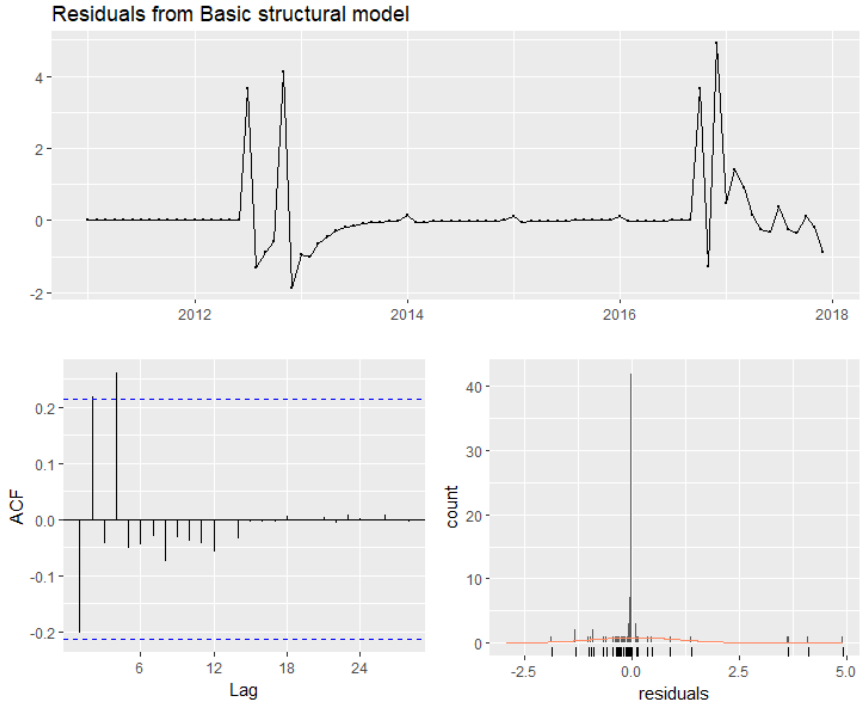

Figure D.25: CWE-295 Model's Residuals Diagnostics 2011–2017

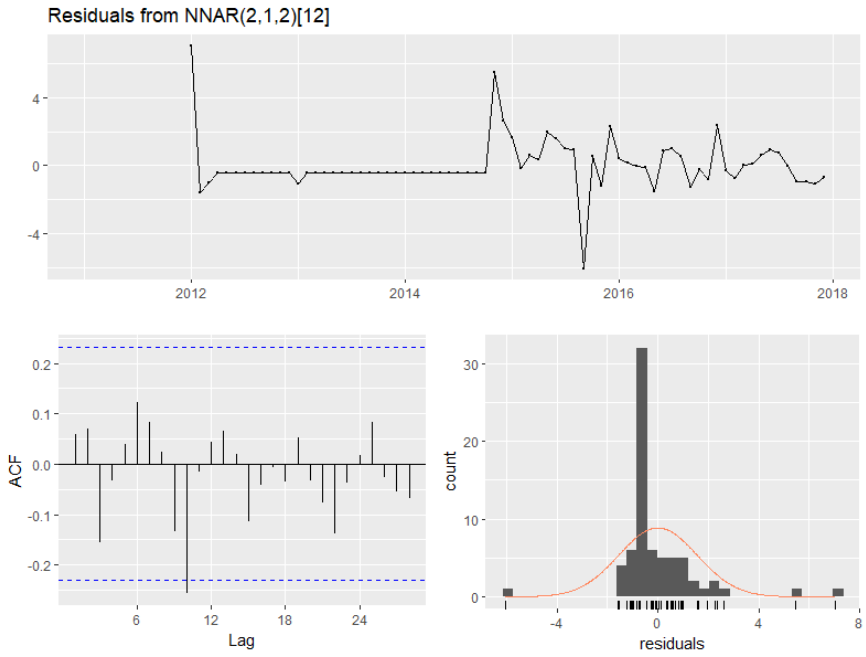

Figure D.26: CWE-19 Model's Residuals Diagnostics 2012–2017

Table D.1: CWE Meanings [3]

CWE-...	Meaning
119	“Buffer Errors”
264	“Cross-Site Scripting (XSS)”
200	“Information Leak / Disclosure”
20	“Input Validation”
399	“Resource Management Errors”
310	“Cryptographic Issues”
284	“Improper Access Control”
89	“SQL Injection”
352	“Cross-Site Request Forgery (CSRF)”
22	“Path Traversal”
189	“Numeric Errors”
94	“Code Injection”
287	“Authentication Issues”
254	“Security Features”
125	“Out-of-bounds Read”
255	“Credentials Management”
416	“Use After Free”
476	“NULL Pointer Dereference”
77	“Command Injection”
190	“Integer Overflow or Wraparound”
19	“Data Handling”
787	“Out-of-bounds Write”
17	“Code”
295	“Improper Certificate Validation”
426	“Untrusted Search Path”
400	“Uncontrolled Resource Consumption (’Resource Exhaustion’)”
611	“Improper Restriction of XML External Entity Reference (’XXE’)”
78	“OS Command Injections”
134	“Format String Vulnerability”
362	“Race Conditions”
59	“Link Following”
502	“Deserialization of Untrusted Data”
16	“Configuration”

D.1 2018 High Severity CWE Previous Accuracy

When considering the 2018 point forecasts and the intervals surrounding the point forecasts, 17 CWEs from the chosen subset of potentially important CWEs are expected to have "High" severity according to Chapter 4.5.3. Figure D.27 shows their previous forecasts' accuracy measures from Listings 4.1, 4.2 and 4.3 as grouped and flipped barplots.

In each of these plots in Figure D.27, the red bars mark the accuracy measure's numeric value for 2016 best forecasts, the blue bars represent the accuracy for 2017 forecasts based on the 2016 best model types using the new training set and the new test set, the green bars indicate the accuracy for 2017 best forecasts based completely on the data downloaded on 1 January 2018. The CWEs only with the green bar are those discovered later when analysing the up-to-date data (Figure 22).

The plots show that in each case, where both a green bar and a red bar are existent, the green bar is shorter than the red bar. This indicates that one additional year in the training set, more training data, seemed to have a positive effect on the forecast accuracy of the best models for these CWEs. The plots also show that in each case, where both the green bar and a blue bar are existent, the green bar is shorter than the blue bar.

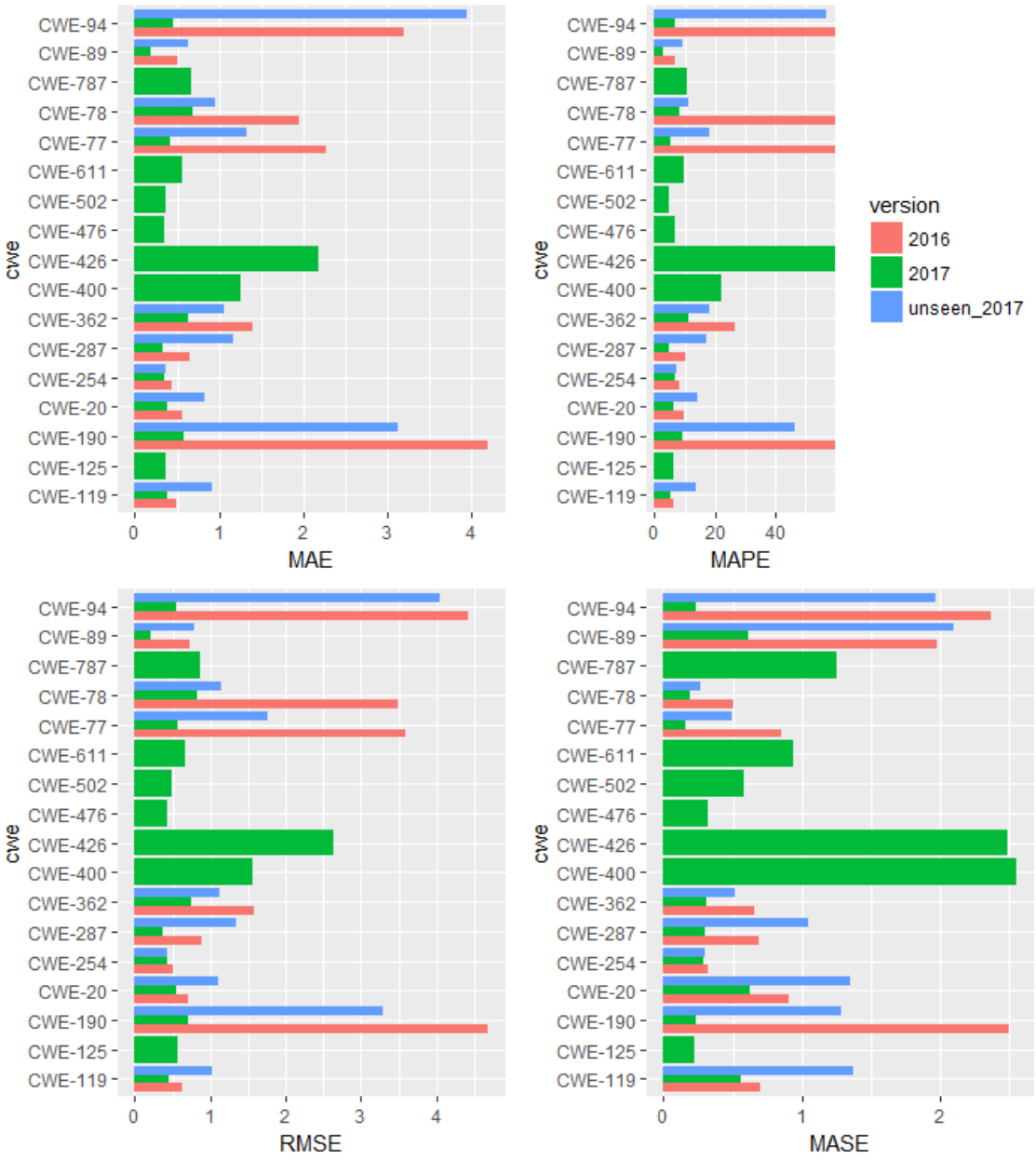

Figure D.27: Previous Forecast Accuracy for 2018 High Severity CWEs

E Reproduced MITRE's Copyright

Terms of Use[61]

LICENSE

The MITRE Corporation (MITRE) hereby grants you a non-exclusive, royalty-free license to use Common Weakness Enumeration (CWE™) for research, development, and commercial purposes. Any copy you make for such purposes is authorized provided that you reproduce MITRE's copyright designation and this license in any such copy.

DISCLAIMERS

ALL DOCUMENTS AND THE INFORMATION CONTAINED THEREIN ARE PROVIDED ON AN "AS IS" BASIS AND THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE MITRE CORPORATION, ITS BOARD OF TRUSTEES, OFFICERS, AGENTS, AND EMPLOYEES, DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

CWE is free to use by any organization or individual for any research, development, and/or commercial purposes, per these CWE Terms of Use. MITRE has copyrighted the CWE List, Top 25, CWSS, and CWRAF for the benefit of the community in order to ensure each remains a free and open standard, as well as to legally protect the ongoing use of it and any resulting content by government, vendors, and/or users. MITRE has trademarked ™ the CWE and related acronyms and the CWE and related logos to protect their sole and ongoing use by the CWE effort within the information security arena. Please contact cwe@mitre.org if you require further clarification on this issue.

[61] `http://cwe.mitre.org/about/termsofuse.html`

F Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Erik Räni**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   of my thesis

   Prediction Model for Tendencies in Cybersecurity

   supervised by Raimundas Matulevičius and Justinas Janulevičius

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 16 May 2018