Comparing Prediction Methods in Anomaly Detection: An Industrial Evaluation

Ralf Greis
University of Luxembourg and
POST Luxembourg
Luxembourg
ralf.greis@post.lu

Thorsten Ries
POST Luxembourg
Luxembourg
thorsten.ries@post.lu

Cu D. Nguyen
POST Luxembourg
Luxembourg
cu.nguyen@post.lu

ABSTRACT

This paper studies the popular Holt-Winters, SARIMA, and Kalman-Filter prediction methods on industrial time series in the cyber-security context. The analyzed datasets represent various data sources from internal networks, telecommunications, firewalls and email traffic, all with particular characteristics that require the utilisation of different algorithms and settings. This includes problems present in training data, being missing, wrong, or containing outliers. The obtained results provide an insight into the performance of the methods on typical industrial datasets, thus, enabling security practitioners to choose a "best fit" method for their use cases and guiding them in choosing the optimal size of training data and providing recommendations for the best settings of the given methods.

KEYWORDS

Holt-Winters, SARIMA, Kalman-Filter, Anomaly Detection

1 INTRODUCTION

Cyber-attacks are nowadays one of the most prominent risks for companies. Criminals have been inventing sophisticated tactics to penetrate into organizations and steal sensitive data. Various companies like Yahoo, Sony, Adobe, Equifax, or J.P. Morgan have been victims of data theft [9], which very often creates serious reputational and financial damage. Therefore, security teams rely on working intrusion detection mechanisms to mitigate such risks. Due to the huge amount of data, the challenge is to detect such intrusions in (near) real-time.

As a proven method, *anomaly detection* has the main advantage as it typically does not require extensive knowledge regarding attacks a priori. There exists a large body of research on anomaly detection methods [4]. Some are based on statistical prediction methods, while others rely on machine learning. Even though the family of machine-learning methods is gaining more and more attention, prediction methods are still a preferable choice in many contexts as they are fast (enabling frequent retraining to cope with the fast pace of service and also attack development) and in most cases require no expensive hardware. All techniques have in common the need of processing complex and huge quantities of time series data.

Valuable sources for anomaly detection are application and system logs but also network traffic, which itself consists of time series from various sources with different characteristics like *missing* data points, *wrong* data, the presence of *outliers*, and high *seasonality*.

MiLeTS '18, August 2018, London, United Kingdom
2018.

Given these preconditions, we experimentally assess three prediction methods, *Holt-Winters*, *Kalman-Filter*, and *SARIMA* on industrial security datasets. Our goal is to study the quality of the methods and how well they perform on the given datasets. The outcome provides an invaluable insight regarding method selection, the amount of data required for training, and the treatment of problematic data. To the best of our knowledge, this is the first experimental study in this context.

The obtained results show that one can achieve a prediction with an error rate of approximately 10%, dependent of the properly selected and configured method. The training data should contain between three to five weeks of historical data when we consider hourly traffic. Furthermore, it has been shown that the prediction rate of Holt-Winters can be significantly improved by treating training data to deal with missing, wrong data, and outliers by replacing them with seasonal means. Kalman-Filter and SARIMA resist well with such problems. Following the evaluation of the results, it has been shown that the Kalman-Filter method is generally a recommended method as it yields a good result in most situations, even on original datasets with outliers or missing/wrong data.

The remainder of the paper is structured as follows. Section 2 discusses time series data, their problems, and gives an overview of the three investigated prediction methods. Sections 3, 4, and 5 provide further details about the methods. Section 6 details our research questions and experiment settings, followed by Section 7 where we report all our findings. Section 8 discusses the implementation of the method in an industrial tool and their application in detecting real-world attacks. Finally, Section 9 concludes the paper.

2 BACKGROUND

2.1 Time Series Data in Security Context

A time series is a list data points in time order. Most often it contains successive values taken with a fixed time interval, e.g, every hour. In security context, we encounter frequently the use of time series data to capture the numbers of events of interest or traffic volume through a network interface over time. Such data gives the analyst invaluable insights about what has been happening to his networks or systems. Furthermore, statistical methods and machine learning models can be carried out on the data to predict the future and to detect anomalous events.

Forecasting is the basis of anomaly detection. Here, historical time series data are used to predict or infer what should happen at a point in time or at the moment. Predicted values are then compared with what actually happens (i.e., with actual values), and a significant difference between the two indicates an anomalous event.

A time series dataset, when being used in anomaly detection, is often divided into two subsets: *training* (aka *fitting*) and *testing*. The former is used for training a machine learning or statistical model, while the latter is used either for assessing the model or detecting anomalous data points within the subset. The training subset takes the large portion of the dataset in most of the cases as one would want to learn as much as possible from the given data at hand. The testing subset, on the contrary, typically contains a few recent data points of the time series.

In practice, however, one has to face a number of challenges while using time series data. The first issue links to *missing data points*. In other words, there could be gaps in data due to the unavailability of the system under observation (e.g., because of a maintenance activity) or due to measurement gaps. The second aspect concerns *wrong data* that are incorrect values in a dataset. They can appear if there were errors made while gathering data. Sometimes datasets are not allowed to have missing values and those are then set to obvious wrong values so that they can be easily detected. Third, *outliers* in a training dataset are a particular issue. Outliers are values from a dataset that are too high or too low compared to the other values. Their appearance is normally caused due to special events like strikes, cyber-attacks, or natural catastrophes. These challenges affect the generality property of predictive methods, leading to undesired prediction. Hence, during the training phase, one should apply necessary treatments to detect and treat missing data, wrong data, and outliers in the data used for learning.

If a dataset shows a repeating pattern for constant periods, like for example every year, every week, or every month, then it is called *seasonal*. Such repeating patterns are often observed in real-world examples. In that case, we can use this knowledge to improve the prediction. One only has to find out the seasonality, i.e., the length of the repeating pattern that is equal to the number of data points of such a period. Normally, the seasonality is very small (four, seven, 12 if we have quarterly, weekly, monthly repeating data, respectively). In our case, however, we deal with hourly data and have to treat high seasonality values that is a challenge with the SARIMA prediction method.

2.2 Anomaly Detection Methods for Time Series

Many anomaly detection methods exist today. Some are based on machine learning and more especially on regression models, clustering, regression or SVM [4] (Support Vector Machine). In the context of time series, however, statistical prediction methods have been receiving more and more attention. On the one hand, it is due to the fact that they require no prior knowledge about the label (i.e., normal or anormal) of each data point in the dataset, as it is a requirement for the majority of machine learning methods (except the *unsupervised* ones). Hence, it is cheaper to apply a statistical prediction method. On the other hand, prediction methods are generally fast with the positive consequence that frequent or even online retraining is affordable.

There exists a large family of prediction methods like the naive methods, moving average methods, ARMA/ARIMA/SARIMA methods, exponential smoothing methods, Kalman-Filter based methods, and more advanced methods based on neural network models or vector autoregression. The most discussed methods are also those used for forecasting in general and especially if the datasets show a seasonal pattern as in our case: the Holt-Winters exponential smoothing methods [11, 12], the SARIMA method [1, 24], and methods based on the Kalman-Filter [19].

The development of the three methods started between 1944-1960 and have been improved over time. The Holt-Winters methods were developed by Charles Holt and Peter Winters and are used in a nearly unchanged way since then. Some historical information and explanation about exponential smoothing methods can be found in [11, 12, 16, 17]. The development of the SARIMA (Seasonal Auto-Regressive Integrated Moving Average) method started by the invention of the AR (Auto Regression) and the MA (Moving Average) methods. We refer the reader to the following two books that give mathematical detail about SARIMA [1, 24]. The Kalman-Filter was invented by Rudolf E. Kalman [19]. It is mainly used to forecast or estimate results from noisy data. It has a lot of applications like for example GPS, satellite navigation devices, smartphones, or computer games. Its famous use was helping the navigation of the Apollo mission to the moon.

2.3 Related Work

Predictive analytics on time series have been investigated in various fields of research [2–4, 7]. It has also been recognized by the industry for detecting data breaches in cyber security contexts [13].

The Holt-Winters prediction method has been applied to time series in [15] for anomaly detection on websites. Holt-Winters was used to forecast the number of pageviews and pageload time. Forecast data are then compared to actual ones for detecting anomalies. The author also dealt with missing values using the mean of previous observations. Our work differs significantly in that as we consider three different methods on many more datasets. We also study the application in practice.

The ARIMA prediction method, which is a sister method of SARIMA discussed in the present paper, has been applied in [6, 8, 23] for anomaly detection. Moayedi et al. used ARIMA on network traffic data in order to isolate anomalies [6]. The data used were simulated data with some artificial attacks that increased the network traffic at some time intervals. The authors showed that ARIMA was capable of detecting those attacks. Wang et al. also used ARIMA on network traffic data obtained from a network of a university campus [23]. Historical data were used for training in order to forecast expectation value at the current moment. It is then compared to actual data in order to detect inconsistencies. The authors claimed to have dealt with abnormal data in the training data by replacing them with forecast values so that subsequence detection will be more accurate. However, the impact of such treatment has not been investigated. Shirani et al. used ARIMA on incoming the size of SOAP messages sending to a web service over time to detect anomalies that are likely linked to XML DoS or brute-force attacks [8]. The method yielded a high accuracy (97%) and a low false positive rate (1.5%) on a dataset made available by Amazon[1].

Prior to the current work, we have done a preliminary analysis and found that the SARIMA model, a variant of ARIMA, that considers seasonality in time series outperforms ARIMA in all the

[1] http://mlsp2012.conwiz.dk/index.php?id=43

datasets investigated. In fact, we observe a clear weekly and hourly pattern of seasonality in traffic data, e.g., having a peak at 8AM and 1PM on working days; weekend days have less traffic than working days. Therefore, SARIMA has been assessed in our experimental studies.

Apart from considering SARIMA instead of ARIMA, our work differs from the related ones on a number of aspects. First, we assess experimentally three different methods, Holt-Winters, Kalman-Filter, and SARIMA on time series security datasets. Second, we implement and deploy Holt-Winters and SARIMA into an industrial platform called Splunk[2], which comes with Kalman-Filter, making them all available to practitioners. Third, we work with a number of diverse and real-world datasets, not artificial ones. Hence, the results are likely more generalizable in practice.

3 HOLT-WINTERS

Several Holt-Winters exponential smoothing methods exist. It depends on the shape of the data to decide which one to use. Our data is regular, i.e., all the seasons have the same shape and similar values. So the mean values of the seasons are rather constant. This means that our data is not really additive or multiplicative, but kind of both and in that case the multiplicative forecasting method works the best. In a preliminary work we compared different Holt-Winters exponential smoothing methods on our data and came also to the result that the multiplicative method works the best. Hence, we will only consider the multiplicative Holt-Winters method, more precisely a damped version of the multiplicative method. The strength of this method is that it is straightforward and easy to understand and to use. Multiplicative triple-exponential-smoothing methods divide the original data into three parts: the level, the trend, and the seasonal component. When they are multiplied together one gets back the original values. We show the main formulas proposed by Hyndman in [5] and explain the main variables.

$$\begin{aligned}
\hat{x}_{t+h|t} &= [L_t + (\phi + \phi^2 + \ldots + \phi^h)B_t]S_{t-m+h_m^+}, \\
L_t &= \alpha \frac{x_t}{S_{t-m}} + (1-\alpha)(L_{t-1} + \phi B_{t-1}), \\
B_t &= \beta(L_t - L_{t-1}) + (1-\beta)\phi B_{t-1}, \\
S_t &= \gamma \frac{x_t}{L_{t-1} + \phi B_{t-1}} + (1-\gamma)S_{t-m},
\end{aligned}$$

where

- $h \in \mathbb{N}$ denotes the *time step* to be forecasted.
- $\hat{x}_{t+h|t}$ denotes the estimated (forecast) values. The index indicates the estimation of x_i at time $i = t + h$ knowing the values of x_i for $i = 0, \ldots, t$.
- L_t is called the *local mean level* or *smoothed value* of the seasonally adjusted time series at time t. L_t is the weighted average of the current observation x_t without seasonality and the estimate of the previous observation without seasonality. As the seasonality is taken out of the data, greater variations are taken out and so we get back a more smoothed version of our initial data.
- B_t is called the *trend* or *slope* of the seasonally adjusted time series at time t. It is the weighted average of the difference between the current smoothed value and the last one and the previous slope. As the smoothed values contained already no seasonality also the B_t contain no seasonal pattern.
- S_t is called the *seasonal component* of the time series at time t. It is the weighted average of the current seasonal component and the seasonal component of the value at the previous season. It represents the seasonal effect on the time series.
- $\alpha, \beta, \gamma, \phi \in [0, 1]$ are called the *smoothing parameters*. α, β, γ tell us how much importance is given in the more recent past and how much importance is given in the further past. The role of ϕ is to damp the values to avoid too high forecasts.

The most important part of these methods is to estimate good smoothing parameters because they have a high impact on the result. A very often used method is to set the initial values at (0.3, 0.3, 0.3, 0.3) and use a hill-climbing method to minimize the *Sum of Squared Errors* of the data. In this work, we use the BFGS [10, 20] hill-climbing algorithm with the starting values (0.0, 1.0, 0.0, 1.0). We found out that this combination provides the best results for our data with $\alpha \sim 0$, $\beta \sim 1$, $\gamma \in [0.5, 0.8]$, and ϕ near 1 or near 0. This means that the recent past values are nearly ignored for the local mean level ($\alpha \sim 0$), while only the latest values are taken into account for the trend ($\beta \sim 1$). Finally, the seasonal component considers the whole value range with an emphasis on recent past values ($\gamma \in [0.5, 0.8]$).

4 KALMAN-FILTER

Kalman-Filter is a forecasting method that has been devised to deal with noise in training data [21]. Figure 1 shows an iteration of the formulas.

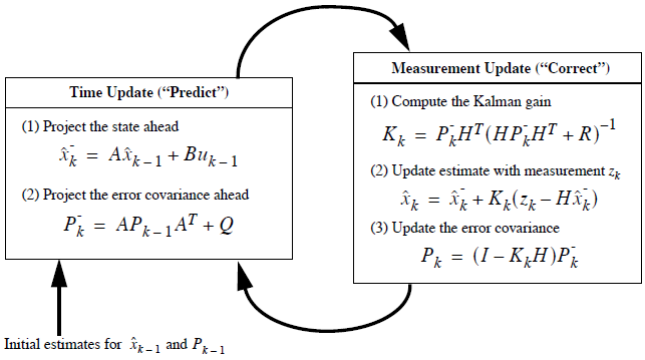

Figure 1: General iteration of the Kalman-Filter method. Source: [21].

For each point in the training set, the Kalman-Filter method makes a forecast $\hat{x}_k^-$ and corrects this value using the actual value $\hat{x}_k$. This correction is done by calculating the so-called *Kalman gain* K_k which indicates the reliability of the forecasted value and its required correction. With these iterations, the method adapts automatically the parameters to improve the forecast so that the difference between the forecasted and the actual value gets smaller and smaller. The strength of this method is that only a few iterations are required. In our case, most of the parameters represent real

[2]https://www.splunk.com

values or vectors, but in multidimensional cases they represent matrices. For more details about these equations, we refer the reader to [14, 19, 21].

Nowadays more than one version of Kalman-Filter exist, but they are all based on the original formulas and are still close to them. The formulas in Figure 1 can even be applied for higher dimensional examples. In a preliminary phase we tested different versions of Kalman-Filter, implemented in Splunk[3], without having the possibility to take a look at the code. Hence, we have no detailed knowledge about whether there is any customization. Nevertheless, we did try different options provided by Splunk and used one combination (*X11 with LLPs*) that delivers the best result on our data in our evaluation.

5 SARIMA

As already mentioned previously, the SARIMA method is an improved version of a mix between the AR and the MA method. The AR method describes values as a linear combination of the past values whereas the MA method describes values as a linear combination of its past errors. ARMA combines both and ARIMA and SARIMA are the improvements for different shapes of data. The main formulas are proposed by Wei in [24]. The SARIMA(p, d, q)(P, D, Q, s) method writes for $p,\ d,\ q,\ P,\ D,\ Q,\ s \in \mathbb{R}$

$$\phi(B)\Phi(B^s)[(1-B)^d\,(1-B^s)^D\,\hat{x}_t] = \theta(B)\Theta(B^s)e_t,$$

where

- $p,\ q,\ P,\ Q \in \mathbb{N}$ indicate how many periods we go back in time. Normally $p, q, P, Q \leq 3$.
- d and D are the *degree of differencing and the degree of seasonal differencing* respectively and normally $d, D \leq 1$. They are used to manipulate the data so that it has a good shape to use the SARIMA method.
- s is the *seasonal factor* and indicates the length of a season in the data.
- e_t is the error at time t.
- B^n is the *backshift operator*, i.e., a function shifting the data back n periods in time: $B^n x_t = x_{t-n}$.
- $\phi(B)$ and $\theta(B)$ are the *regular autoregressive and moving average factors* depending on p and q:
- $\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$, with $\phi_1,\ \ldots,\ \phi_p \in \mathbb{R}$,
- $\theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q$, with $\theta_1,\ \ldots,\ \theta_q \in \mathbb{R}$.
- $\Phi(B^s)$ and $\Theta(B^s)$ are the *seasonal autoregressive and moving average factors* depending on P and Q:
- $\Phi(B) = 1 - \Phi_1 B - \ldots - \Phi_P B^P$, with $\Phi_1,\ \ldots,\ \Phi_P \in \mathbb{R}$,
- $\Theta(B) = 1 - \Theta_1 B - \ldots - \Theta_Q B^Q$, with $\Theta_1,\ \ldots,\ \Theta_Q \in \mathbb{R}$.

To use the SARIMA method, one first has to determine the parameters p, d, q, P, D, Q, and s. This takes some time and analysis, so we refer the reader to [1] and [24] for more details about how to find the parameters and also about the SARIMA method in general.

Normally the value of s is typically clear by analyzing the time series representation (e.g., data plot over time). However, no perfect method exists so far for the other six parameters. d and D are estimated based on unit root test [5]. Finally, one can adopt the Hyndman-Khandakar algorithm [5] to optimize p, q, P, Q. Their initial values can be obtained thanks to the autocorrelation function (ACF) and partial autocorrelation (PACF) plots, then one need to vary them (usually with a step of 1 within the range from 0 to 3) to reach the lowest error rate. Using this method the following combination SARIMA(1, 1, 1)(1, 1, 1, s) gives the lowest error rate in our evaluation.

[3] https://docs.splunk.com/Documentation/Splunk/7.1.0/SearchReference/Predict

6 EXPERIMENT SETUP

6.1 Research Questions

The goal of this study is to experimentally investigate the performance of the three prediction methods, Holt-Winters, Kalman-Filter, and SARIMA. Specifically, we seek to answer the following research questions concerning the prediction capability of the methods and the impact of the quality and quantity of training data:

- **RQ1**: How effective are the methods in prediction?
- **RQ2**: How does the size of training data (aka the quantity of training data) impact the methods?
- **RQ3**: What is the influence of wrong, missing, or outliers in training data to the performance of the methods?

6.2 Setting

Table 1 summarizes the four time series datasets used in our experiments. These come from different IT and telecommunication areas of an incumbent ISP and are relevant security data, required for visibility and incident detection. The first dataset, DNS, pertains to DNS queries that translate hostnames to IP addresses. The second one, Firewall, contains data of network security events. The third one, Email, links to the number of email sent and received through our email gateway. Finally, Diameter contains mobile phone signaling events from our mobile network. Note that in the course of this study, only quantitative data, i.e., number of events, has been used. Hence, the data sets don't contain personal data.

Table 1: The available datasets used in our experiments.

Dataset	Timespan	Description
DNS	10 weeks	DNS requests
Firewall	13 weeks	Network traffic
Email	10 weeks	Emails exchanged through
Diameter	13 weeks	Call signaling in a mobile network

All of them share the same characteristics as detailed below:

- The datasets cover periods between 80 and 100 days with hourly counts, i.e., the value at 1AM is the count of all the events that happened between 0-1AM.
- All the datasets show a seasonality of one week, i.e., of $24*7 = 168$ points.
- Initial visualization points to obvious daily up and down: high values during the daily hours and low values during the night. The weekdays and the weekends show the same pattern respectively.
- The weekdays have a higher overall mean than the weekends.

Anomalies detected from such datasets could link to various cyber-attacks or frauds. Anomalies on DNS might indicate unusual

DNS traffic related to malwares trying to hide themselves (e.g, *Domain Generation Algorithms (DGA)*) or a degradation of DNS servers. Anomalies detected in Firewall indicate unusual traffic in network that might link to data exfiltration, malicious network scanning. Anomalies detected in Email pertain to email threats such as phishing campaigns or spamming. Last but not least, anomalies in Diameter might be related to mobile network threats such as the *SS7 attack* or *call frauds*.

We will use the three forecasting methods, Holt-Winters, SARIMA, and Kalman-Filter exactly in the way as described above. We recall the used parameters in Table 2. For the Diameter dataset we do not use the multiplicative Holt-Winters method as it contains many missing values. We therefore make an exception and use the additive Holt-Winters method.

Table 2: Used forecasting methods with their parameters.

Holt-Winters	$\alpha \sim 0$, $\beta \sim 1$, $\gamma \in [0.5, 0.8]$, ϕ near 1 or near 0
Kalman-Filter	The best option of Splunk
SARIMA	$(p, d, q, P, D, Q) = (1, 1, 1, 1, 1, 1)$

To measure the performance of the methods in terms of time, all experiments were carried out on a dedicated server having eight cores CPU 2.8GHz, 16Gb RAM.

6.3 Error Measures

To evaluate the prediction capability we fix the length of the testing dataset to one week and change the size of the training data of each dataset. We then evaluate the prediction capability of the three methods using two different error measures: *mean absolute error* (MAE) and the *mean absolute percentage error* (MAPE). Their formulas can be found in Table 3.

The MAE sums up the errors for each value in the testing dataset and then divides by the number of points to calculate the average error. So obviously the MAE is a scale-dependent error measure using the same scale as the dataset. It can only compare time series which are on the same scale and can be used to compare the results of different prediction methods on the same dataset.

The MAPE is a normalization of the MAE and describes the error in percentage. It is therefore scale-independent, hence can be used to compare the results of the same or different prediction methods on different datasets.

Table 3: Formulas of the error measures.

MAE	$\text{mean}(\lvert x_t - \hat{x}_t \rvert)$
MAPE	$\text{mean}\left(\left\lvert 100\,\frac{x_t - \hat{x}_t}{x_t}\right\rvert\right)$

7 RESULTS AND DISCUSSION

In this section, we report the experimental results comparing how the three methods, Holt-Winters, Kalman-Filter, and SARIMA perform on the four datasets with respect to the error measures (MAE and MAPE). Recall that on each dataset, we consider the data of the last week for testing, i.e., for measuring the error rates. The data of the other weeks are used for training with different controlled sizes. For DNS and Email, we consider four different training sizes of one, three, five, and ten weeks. For the others, we consider one additional size of 13 weeks as they have more data.

The four original datasets were made of hourly values over up to 13 weeks with a repeating pattern every week. So our seasonality was $24 * 7 = 168$. This is a very high value as it is normally 4, 7, or 12. We only found one work by Hyndman et al. [18] talking about this problem using the ARIMA method. The calculation with the SARIMA method took several hours, which questions its applicability. Therefore, we only used the Holt-Winters and the Kalman-Filter on the original datasets (SARIMA is applicable when we reduce the seasonality, which will be discussed shortly).

Table 4 shows the results for the different sizes of the training sets. During the calculations we observed that both methods need at least one season as training dataset. They run rather fast as both need just a few seconds to finish the complete process from training to prediction. We also see that Kalman-Filter gives in all cases better results than the Holt-Winters method. It is also worth mentioning that Holt-Winters and Kalman-Filter seem to have no problem with the high seasonality. Considering Holt-Winters, we can see that the training has an overall influence on the performance of the method: it yields the worst error rates when the size is equal to one week and the best one the size is three or five weeks. Also, the addictive variant of Holt-Winters performs worse than its multiplicative one. On the contrary, the results of Kalman-Filter on the different training size exhibit no clear pattern. For each dataset, the results are very close, independently of the length of the training set.

Since SARIMA cannot cope with the high seasonality within an acceptable execution time, we have pre-processed the datasets to change the interval from 1 h to 4 h by taking the sum of every four consecutive hours. As a result, the seasonality factor is reduced to 42 and we have only one-fourth of the datapoints. Now the computation of the SARIMA method took between 20 and 70 seconds. For the Holt-Winters and the Kalman-Filter method, we recognized no greater change in the speed. Table 5 shows the results for different sizes of the training sets. We observe also that the SARIMA method is better than Holt-Winters and Kalman-Filter in two out of four datasets, Kalman-Filter yields error rates that are quite close to those of SARIMA. And finally, Holt-Winters performs the worst in all but one case (Firewall), in which all three deliver a comparable result. Given the difference in execution time and their result, Kalman-Filter is once again a preferable choice. SARIMA delivered a completely wrong prediction in the case of five weeks Email. We investigated this issue and will discuss the finding shortly.

Considering all results from the tables 4 and 5 we can answer $\mathbf{RQ}_1$:

> *Overall Kalman-Filter yields the most acceptable results in most cases with an error rate of approximately 10% within a short time of a few seconds.*

Looking at the different length of the training datasets, we can conclude that we get generally the best results if the length is in between three to five weeks (note that SARIMA might require at least four weeks). Shorter time periods do not provide enough historical data to make a meaningful prediction. If the training period

Table 4: Results for interval length equal to 1h. One needs at least 1 week as training set to use the Holt-Winters and the Kalman-Filter method. Size(w) stands for the training size in weeks.

Dataset	Size(w)	Method	Time(s)	MAE	MAPE(%)
DNS	10	HW	2	12,348	14.13
	5	HW	2	**11,426**	**12.95**
	3	HW	1	13,493	17.28
	1	HW	1	23,193	31.35
DNS	10	KF	1	10,315	12.40
	5	KF	1	12,519	15.38
	3	KF	1	10,402	11.59
	1	KF	1	**9,819**	**10.77**
Diameter	13	HW (addi.)	2	58,487	64.17
	10	HW (addi.)	2	68,788	78.08
	5	HW (addi.)	1	**42,087**	**50.76**
	3	HW (addi.)	1	46,934	56.60
	1	HW (addi.)	1	60,453	64.27
Diameter	13	KF	1	10,961	12.15
	10	KF	1	14,346	14.48
	5	KF	1	**9,070**	**10.63**
	3	KF	1	12,152	12.60
	1	KF	1	12,513	12.98
Firewall	13	HW	6	13,576	11.96
	10	HW	2	13,567	12.09
	5	HW	2	13,138	12.78
	3	HW	2	**8,823**	**8.97**
	1	HW	1	21,249	20.39
Firewall	13	KF	1	15,676	13.45
	10	KF	1	15,640	13.40
	5	KF	1	15,029	12.46
	3	KF	1	**13,966**	**11.75**
	1	KF	1	15,490	13.13
Email	10	HW	4	**861**	**27.75**
	5	HW	2	1,008	34.30
	3	HW	1	1,079	33.19
	1	HW	1	1,345	39.79
Email	10	KF	1	794	22.65
	5	KF	1	**736**	**20.57**
	3	KF	1	780	21.90
	1	KF	1	808	21.21

is bigger we have too many historical data and so the prediction will be less reactive to recent changes.

To answer $\mathbf{RQ}_2$:

Different training sizes yield different results, and the recommended size is in between three to five weeks for hourly data.

We studied the influence of wrong, missing data, and the presence of outliers in training data to the performance of the prediction methods. We propose to replace such data points with a corresponding seasonal mean. For instance, a missing data on Thu at 2PM is filled up with the average of other data points available on the same week day and time from other weeks. We used the Boxplot method [22] to detect wrong data and outliers in training data.

Figures 2 and 3 show the MAPE error of all the three methods on our Diameter dataset, without and with the treatment of training data. We pick Diameter as it has plenty missing data due to a maintenance during data collection. Looking at the MAPE barplots we observe that the treating of the training data has an influence on the result. It has the biggest influence on the Holt-Winters method. As a result, it is more vulnerable to wrong data than the other two methods. The SARIMA and the Kalman-Filter methods show a very slight amelioration. We repeated the same evaluation on other datasets that contain fewer missing or outliers. We also observed the similar effect. Thus we can provide a clear answer to $\mathbf{RQ}_3$:

Table 5: Results for interval length equal to 4 h. SARIMA reported an error when running with one and three-weeks training sets. Size(w) stands for the training size in weeks.

Dataset	Size(w)	Method	Time(s)	MAE	MAPE(%)
DNS	10	HW	2	45,479	12.42
	5	HW	1	**39.873**	**10.50**
	3	HW	1	40,063	11.64
	1	HW	1	112,580	35.20
DNS	10	KF	1	36,662	10.15
	5	KF	1	47,642	13.88
	3	KF	1	37,193	9.73
	1	KF	1	**32,591**	**8.34**
DNS	10	SARIMA	28	34,075	9.17
	5	SARIMA	68	**31,962**	**8.19**
Diameter	13	HW (addi.)	1	63,702	18.07
	10	HW (addi.)	1	69,734	20.84
	5	HW (addi.)	1	**56,146**	**16.74**
	3	HW (addi.)	1	109,999	31.51
	1	HW (addi.)	1	119,429	31.23
Diameter	13	KF	1	41,654	11.76
	10	KF	1	53,246	14.19
	5	KF	1	**34,689**	**10.77**
	3	KF	1	47,680	12.69
	1	KF	1	49,438	13.36
Diameter	13	SARIMA	33	**40,559**	**11.67**
	10	SARIMA	25	41,194	12.28
	5	SARIMA	68	47,349	12.76
Firewall	13	HW	1	52,301	11.96
	10	HW	1	52,214	11.97
	5	HW	1	48,436	11.71
	3	HW	1	**31,230**	**8.36**
	1	HW	1	188,178	41.70
Firewall	13	KF	1	62,936	14.06
	10	KF	1	63,582	14.14
	5	KF	1	58,138	12.83
	3	KF	1	**54,242**	**12.14**
	1	KF	1	58,021	12.98
Firewall	13	SARIMA	28	48,073	10.93
	10	SARIMA	28	**48,070**	**10.93**
	5	SARIMA	5	51,705	12.05
Email	10	HW	2	**3,106**	**25.30**
	5	HW	1	3,583	29.74
	3	HW	1	3,703	30.40
	1	HW	1	4,767	31.96
Email	10	KF	1	2,967	20.97
	5	KF	1	**2,734**	**18.63**
	3	KF	1	2,928	20.47
	1	KF	1	3,093	19.86
Email	10	SARIMA	22	**2,268**	**15.09**
	5	SARIMA	60	257,119	2,099.73

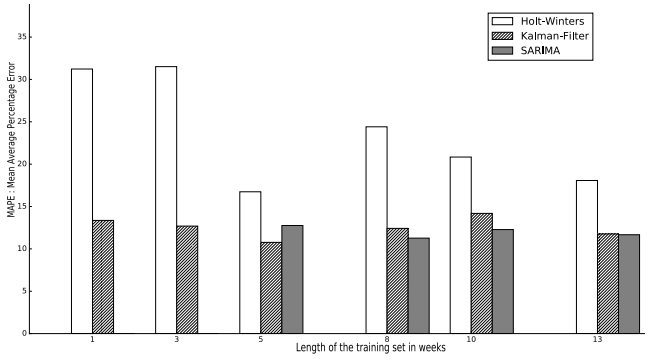

Figure 2: Barplot showing the MAPE of the Diameter dataset with 4 h interval for different lengths of the training data.

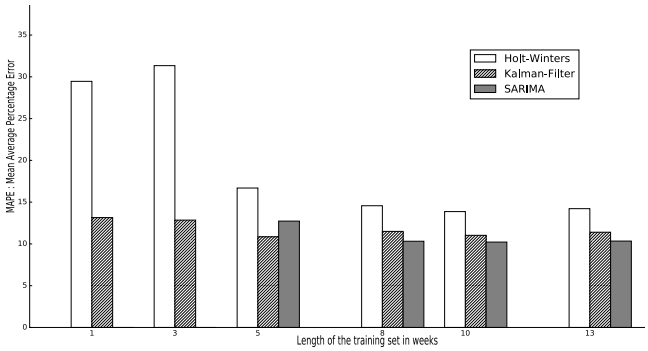

Figure 3: Barplot showing the MAPE of the treated Diameter dataset with 4 h interval for different lengths of the training data.

Wrong, missing values or outliers in training data do have a noticeable impact on Holt-Winters only. Replacing them with seasonal means helps improve prediction outcomes. Kalman-Filter and SARIMA are immune to those problems to a large extent.

The results above show a very strange and bad result when using the SARIMA method on the Email dataset with a training set size of five weeks. For all the other lengths of training sets, we get valuable results using the SARIMA(1,1,1)(1,1,1,42) configuration. There is no obvious reason why in that specific case we get a totally bad result. It seems that some points in this dataset with that length disturb the method.

Following, we investigate the results of different parameter configurations to examine if the results outperform the previous results. They are summarized in Table 6.

We see that nearly all the configurations give a good result better than the Holt-Winters method (MAPE = 29.74 %). The configuration SARIMA(1,1,1)(1,0,1,42) is also better than the Kalman-Filter method (MAPE = 18.63 %). However, we can also observe that the parameters of SARIMA have a significant impact on its results. Wrongly chosen parameters can lead to extreme deviating predictions, getting MAPEs of 2, 099.73 and 10, 000.27% for instance. We can conclude that SARIMA is sensitive to its configurations and some might render the method completely useless.

Table 6: Different parameter configurations for SARIMA for the 4 h internal Email log dataset with a training size length of five weeks

SARIMA Parameters	MAPE (%)
(1,1,1)(1,1,1,42)	2,099.73
(0,1,1)(1,1,1,42)	23.14
(1,0,1)(1,1,1,42)	28.44
(1,1,0)(1,1,1,42)	23.17
(1,1,1)(0,1,1,42)	10,000.27
(1,1,1)(1,0,1,42)	17.44
(1,1,1)(1,1,0,42)	22.62

8 IMPLEMENTATION AND APPLICATION

The present research work has been carried out in an industrial context with real-world data. Apart from assessing the considered methods scientifically, we also aim at their utilization in practice. Therefore, we have developed two add-ons for Splunk, one of the most powerful industrial tools for security and business analytics that can handle big data and enable real-time analyses. The Kalman-Filter method has been built into Splunk by its vendor. Our add-ons implement the other two methods, Holt-Winters and SARIMA. They are currently available upon request and we plan to make them open-source in the near future.

Figure 4 shows a visual comparison of the prediction of the methods versus actual data of a dataset during its last week. It is clearly visible that the predicted data are close to actual ones across all methods.

Moreover, we have used these methods to successfully detect real-world security incidents. For instance, we have been able to detect many *call frauds* in which a malicious attacker generated a massive number of dropped calls to trick recipients to call back through a premium number. The recipients have to pay a high fee for such a return call. Also, we identified an anomaly in DNS traffic due to a misconfiguration that led to network degradation. Finally, we discovered an email spam campaign that misused IoT devices that generated a huge amount of emails.

9 CONCLUSION

Knowing about the importance of real-time and near-real-time incident detection (and mitigation), statistical-based anomaly detection methods play a crucial role, as they are fast, do not require extensive computing power, and more importantly, not require prior knowledge of attacks. As they take just a few seconds for a full cycle from training to detection, we can retrain our models frequently to cope with the fast-changing pace of our services so as to cope with the ever-evolving attack landscape.

In this paper, we compared three prediction methods, Holt-Winters, SARIMA, and Kalman-Filter, that are popular for time series data. Our goal was to study the characteristics of the methods and how they perform on the industrial security datasets. We evaluated the methods with different training sizes to find out the

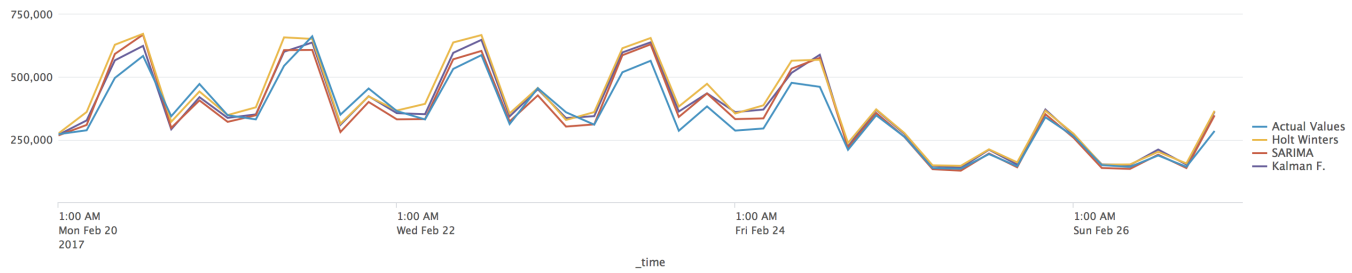

Figure 4: A screenshot of the implementation of the prediction methods in Splunk. The figure shows a visual comparison of the methods' prediction versus actual data.

effect of training size and the recommended size that should be considered in practice. Furthermore, we also treated problems in training data, including *missing* data points, *wrong* data, the presence of *outliers*, and high *seasonality*.

The obtained results indicate that the Holt-Winters and the Kalman-Filter methods are fast even for high seasonality, SARIMA is much slower in comparison with the others. Also, SARIMA requires time and expertise to find good parameters. In terms of prediction, SARIMA is slightly better than Kalman-Filter, but both of them are much better than Holt-Winters. Moreover, Holt-Winters is prone to wrong, missing data, and outliers, while Kalman-Filter and SARIMA resist well to them. Overall, Kalman-Filter is the recommended method.

The second big question was whether the length of the training data has an influence on the prediction and if yes to find the optimal length. We can conclude that one gets the best results when the length of the training data is in between three to five weeks of hourly data. Smaller or bigger training data might lead to undesired results. Note that this finding applies to the data used in our experiments and probably for datasets that share similar characteristics (see Section 6.2).

As an additional contribution of the work, we have implemented Holt-Winters and SARIMA and made them available as add-ons for Splunk, an industry tool for big data analytics. We applied Kalman-Filter in real-world use cases and have detected various security incidents, e.g, email spamming or call frauds.

ACKNOWLEDGEMENTS

This research has been performed with the support of the Cyber Security Department at POST Luxembourg.

REFERENCES

[1] Peter J. Brockwell and Richard A. Davis. 1990. *Time Series: Theory and Methods.* Springer.
[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *ACM Comput. Surv.* 41, 3, Article 15 (July 2009), 58 pages. https://doi.org/10.1145/1541880.1541882
[3] Deepthi Cheboli. 2010. *Anomaly Detection of Time Series.* Master's thesis. University of Minesota.
[4] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers Security* 28, 1 (2009), 18 – 28. https://doi.org/10.1016/j.cose.2008.08.003
[5] Rob J. Hyndman and George Athanasopoulos. 2014. *Forecasting: Principles and Practice.* OTexts.com.
[6] H. Zare Moayedi and M. A. Masnadi-Shirazi. 2008. Arima model for network traffic prediction and anomaly detection. In *2008 International Symposium on Information Technology*, Vol. 4. 1–6. https://doi.org/10.1109/ITSIM.2008.4631947
[7] Animesh Patcha and Jung-Min Park. 2007. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks* 51, 12 (2007), 3448 – 3470. https://doi.org/10.1016/j.comnet.2007.02.001
[8] P. Shirani, M. A. Azgomi, and S. Alrabaee. 2015. A method for intrusion detection in web services based on time series. In *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*. 836–841. https://doi.org/10.1109/CCECE.2015.7129383
[9] Dave Albaugh. 2017 (accessed 11.07.2017). *Biggest data breaches in history.* https://www.comparitech.com/blog/information-security/biggest-data-breaches-in-history/.
[10] Peter Blomgren. 2016. Numerical Analysis Lecture Notes # 18: Quasi-Newton Methods - The BFGS method.
[11] Chris Chatfield. 1978. Holt-Winters forecasting Procedure. *Journal of the Royal Statistical Society, Applied Statistics* 27, 3 (1978), 264–279.
[12] Chris Chatfield and Mohammad Yar. 1985. Holt-Winters forecasting: some practical issues. *Journal of Forecasting* 4, 2 (1985), 129–140.
[13] Ben Dickson. 2016 (accessed 11.07.2017). *How predictive analytics discovers a data breach before it happens.* https://techcrunch.com/2016/07/25/how-predictive-analytics-discovers-a-data-breach-before-it-happens/.
[14] Ramsey Faragher. 2012. Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation. *IEEE SIGNAL PROCESSING MAGAZINE* (2012), 128–132.
[15] Georgios Galvas. 2016. Time series forecasting used for real-time anomaly detection on websites. Master Thesis.
[16] Everett S. Gardner Jr. 1978. Exponential Smoothing: The State of the Art. *Journal of the Royal Statistical Society, Applied Statistics* 27 (1978), 1–38.
[17] Everett S. Gardner Jr. 2005. Exponential Smoothing: The State of the Art - Part 2.
[18] Rob J. Hyndman. 2010 (accessed 11.07.2017). *Forecasting with long seasonal periods.* https://robjhyndman.com/hyndsight/longseasonality/.
[19] R. E. Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering* 82, Series D (1960), 35–45.
[20] Dong Hui Li and Masao Fukushima. 2001. On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. *Society for Industrial and Applied Mathematics* 11, 4 (2001), 1054–1064.
[21] Greg Welch and Gary Bishop. 2006. An Introduction to the Kalman Filter.
[22] John W Tukey. [n. d.]. Exploratory Data Analysis. ([n. d.]).
[23] Guilan Wang, Zhenqi Wang, and Xianjin Luo. 2009. Research of Anomaly Detection Based on Time Series. In *World Congress on Software Engineering*, Vol. 1. 444–448.
[24] William W. S. Wei. 2006. *Time Series Analysis: Univariate and multivariate methods.* PEARSON Addison Wesley.