Predictive Anomaly Detection

Wassim Berriche, Françoise Sailhan

Predictive Anomaly Detection

Wassim Berriche[1] and Francoise Sailhan[2]

[1] SQUAD & Cedric Laboratory, CNAM Paris, France
[2] IMT Atlantique, LAB-STICC laboratory, Brest, France

Abstract. Cyber attacks are a significant risk for cloud service providers and to mitigate this risk, near real-time anomaly detection and mitigation plays a critical role. To this end, we introduce a statistical anomaly detection system that includes several auto-regressive models tuned to detect complex patterns (e.g. seasonal and multi-dimensional patterns) based on the gathered observations to deal with an evolving spectrum of attacks and a different behaviours of the monitored cloud. In addition, our system adapts the observation period and makes predictions based on a controlled set of observations, i.e. over several expanding time windows that capture some complex patterns, which span different time scales (e.g. long term versus short terms patterns). We evaluate the proposed solution using a public dataset and we show that our anomaly detection system increases the accuracy of the detection while reducing the overall resource usage.

Keywords: Anomaly detection, ARIMA, Time series, Forecasting

1 Introduction

In the midst of the recent cloudification, cloud providers remain ill-equipped to cope with security and cloud is thereby highly vulnerable to anomalies and misbehaviours. It hence becomes critical to monitor today's softwarised cloud, looking for unusual states, potential signs of faults or security breaches. Currently, the vast majority of anomaly detectors are based on supervised techniques and thereby require significant human involvement to manually interpret, label the observed data and then train the model. Meanwhile, very few labelled datasets are publicly available for training and the results obtained on a particular controlled cloud (e.g. based on a labelled dataset) do not always translate well to another setting. In the following, we thus introduce an automated and unsupervised solution that detects anomalies occurring in the cloud environment, using statistical techniques. In particular, our anomaly detector relies on a family of statistical models referring to AutoRegressive Integrated Moving Average (ARIMA) and its variants [1], that model and predict the behaviour of the softwarised networking system. This approach consists in building a predictive model to provide an explainable anomaly detection. Any observation that is not following the collective trend of the time series is refereed as an anomaly. Still, building a predictive model based on the historical data with the aims of forecasting future values and further detect anomalies, remains a resource-intensive

process that entails analysing the cloud behaviour as a whole and typically over a long period of time, on the basis of multiple indicators, such as CPU load, network memory usage, packet loss collected as time series to name a few. It is therefore impractical to study all the historical data, covering all parameters and possible patterns over time, as this approach hardly scales. Furthermore, the performance of such approach tends to deteriorate when the statistical properties of the underlying dataset (a.k.a. cloud behaviour) changes/evolves over time. To tackle this issue, some research studies, e.g., [2], determine a small set of features that accurately capture the cloud behavior so as to provide a light detection. An orthogonal direction of research [3] devises sophisticated metrics (e.g., novel window-based or range-based metrics) that operate over local region. Differently, we propose an adaptive forecasting approach that addresses these issues by leveraging expanding window: once started, an expanding window is made of consecutive observations that grow with time, counting backwards from the most recent observations. The key design rational is to make predictions based on a controlled set of observations, i.e. over several expanding time windows, to capture some complex patterns that may span different time scales and to deal with changes in the cloud behaviour. Overall, our contributions includes:

- an unsupervised anomaly detection system that incorporates a family of autoregressive models (§3.2) supporting both univariate, seasonal and multivariate time series forecasting. Using several models – in opposition to a single model that is not necessarily the best for any future uses – increases the chance to capture seasonal patterns and complex patterns. The system decomposes the observations (i.e., time series) and attempts to forecast the subsequent behaviour. Then, any deviation from the model-driven forecast is defined as an anomaly that is ultimately reported.
- Our system uses expanding windows and therefore avoids the tendency of the model to deteriorate over time when the statistical properties of the observations change at some points. When a significant behavioural change is observed, a new expanding window is started. This way, the observations depicting this novel behaviour are processed separately. Thus, the resulting forecast better fits and the anomaly detection is robust to behaviour changes.
- Following, we assess the performances associated with our anomaly detector (§4) considering a cloud native streaming service.

2 Adaptive Anomaly Detection

Auto-regressive algorithms are commonly used to predict the behaviour of a system. As illustration, network operator attempt to predict the future bandwidth/application needs [4] so as to provision in advance sufficient resources. In the same way, we propose to monitor and predict the behaviour of the softwarised network. Then, anomalies are detected by comparing the expected/predicted behaviour with the actual behaviour observed in the network ; the more deviant this behaviour is, the greater the chance that an attack is underway. In practice, the problem is that detection accuracy tends to degrade when there is

(even a small) change in behaviour. We thus introduce an anomaly detection system that relies on several auto-regressive models capable of capturing seasonal and correlated patterns for which traditional methods, including the small body of works ([5] leveraging univariate methods, fail on this aspect. In addition, our anomaly detection system uses several expanding windows to deal with a wider range of behavioural patterns that span different time scales and may change over time. From the moment a noticeable change of behaviour is observed, a new windows that runs over an underlying collection is triggered. Our anomaly detector consists in studying past behaviour based on some key indicators (e.g. CPU usage, amount of disk read) that are expressed as a set of K time series $\mathcal{Y} = \{y_t^1\}, \{y_t^2\}, \cdots, \{y_t^K\}$, with $K \geq 1$ and t is the time, with $t \geq T_0$ where T_0 denotes the start time. The behaviour forecasting is performed at equally spaced points in time, denoted $T_0, T_1, \cdots T_i, \cdots$. At time T_i, the resulting forecast model $\hat{\mathcal{M}}_i$ is established accordingly for the next period of time $\Delta T = [T_i, T_{i+1}]$. In particular, we rely on 3 regressive models (as detailed in §3.1) so as to establish in advance the expected behaviour of the softwarised network and compare them with that observed, at any time $t \geq T_1$. Rather than exploiting the whole historical dataset $\mathcal{Y}$, the analysis is focused on several time windows (i.e. time frames) to achieve some accurate predictions. Time window has the advantage of not having to conveniently deal with the never-ending stream of historical data that are collected. Small window typically accommodates short-term behaviour whilst allowing real-time anomaly detection at low cost. As a complement, larger window covers a wider variety of behaviours and ensures that long term behaviour are considered. Any expanding window W_j (with $1 \leq j \leq J$) is populated with the most recent data points and moves step-wise along the time axis as new observations are received: as time goes, the window grows. Let $\{1, \cdots, w_j\}$ denote the time stamps sequence of observations that are collected during any given time window W_j. This rolling strategy implies that observations are considered for further data analysis as long as they are located in the current window W_j. At time T_i (with $T_i \geq T_1$), all the windowed times series $\{\mathcal{Y}\}_{t=T_i-W_1}^{T_i}, \{\mathcal{Y}\}_{t=T_i-W_2}^{T_i}, \cdots, \{\mathcal{Y}\}_{t=T_i-W_j}^{T_i}, \cdots$ are analysed by a data processing unit that performs the forecasting and produces the predictive model $\hat{\mathcal{M}}^i = \hat{\mathcal{M}}^i(\{\mathcal{Y}\}_{t=T_i-W_1}^{T_i}), \cdots, \hat{\mathcal{M}}^i(\{\mathcal{Y}\}_{t=T_i-W_n}^{T_i})$. For this purpose, a family of predictive models denoted $\hat{\mathcal{M}}^i = \hat{\mathcal{M}}^i_{ARIMA}, \hat{\mathcal{M}}^i_{SARIMA}, \hat{\mathcal{M}}^i_{VARMA}$ is used. Based on $\hat{\mathcal{M}}_i$, the aim is to detect some anomalies $\mathcal{A}_i$, with $\mathcal{A}_i \subset \{\mathcal{Y}\}_{T_i}^{T_{i+1}}$.

3 Anomaly Detection based on Time Series Forecasting

We introduce an anomaly detection system that continuously detects anomalies and supports time series forecasting, which corresponds to the action of predicting the next values of the time series, leveraging the family of predictive models (§3.1) and making use of expanding windows (§3.2) to detect anomalies (§3.3).

3.1 Time Series Forecasting

Time series forecasting is performed by a general class of extrapolating models based on the frequently used AutoRegressive Integrated Moving Average (ARIMA) whose popularity is mainly due to its ability to represent a time series with simplicity. Advanced variants, including Seasonal ARIMA and Vector ARIMA are further considered to deal with the seasonality in the time series and multidimensional (a.k.a multivariate) time series.

Autoregressive Integrated Moving Average (ARIMA) process for univariate time series combines Auto Regressive (AR) process and Moving Average (MA) process to build a composite model of the time series. During the auto regressive process that periodically takes place at time $T_i = t_O + i\Delta T$ for any expanding windows W_j (with $1 \leq j \leq J$), the variable of interest y_t^k (with $T_i \leq t \leq T_{i+1}$ and $1 \leq k \leq K$) is predicted using a linear combination of past values of the variable $y_{t-1}^k, y_{t-2}^k, \cdots, y_{t-w_j}^k$ that have been collected during w_j:

$$y_t^k = \mu^k + \sum_{i=1}^{w_j} \phi_i^k y_{t-i}^k + \varepsilon_t^k \tag{1}$$

where μ^k is a constant, ϕ_i^k is a model parameter and y_{t-i}^k (with i= 1, $\cdots$, w_i) is a lagged value of y_t^k. ε_t^k is the white noise a time t, i.e., a variable assumed to be independently and identically distributed, with a zero mean and a constant variance. Then, the Moving Average (MA) term y_t^k is expressed based on the past forecast errors:

$$y_t^k = c^k + \sum_{j=1}^{q} \theta_j^k \varepsilon_{t-j}^k + \varepsilon_t^k = \theta(B)\ \varepsilon_t^k \tag{2}$$

where θ_i^k and respectively ε_{t-i}^k (with i= 1, $\cdots$, ΔT) are the model parameter and respectively the random shock at time $t-j$. ε_t^k is the white noise at time t, B stands for backshift operator and $\theta(B) = 1 + \sum_{j=1}^{w_j} \theta_j^k B^j$. Overall, the effective combination of Auto Regressive (AR) and moving average (MA) processes forms a class of time series model, called ARIMA, whose differentiated time series $y't$ is expressed as: $\phi^k(B)(1-B)^d y_t^k = \mu^k + \theta^k(B)$ with $\phi^k(B) = 1 - \sum_{i=1}^{p} \phi_i^k B^i$ and $y_t'^k = (1-B)^d y_t^k$ and d represent the number of differentiation. When seasonality is present in a time series, the Seasonal ARIMA model is of interest.

Seasonal ARIMA (SARIMA) process deals with the effect of seasonality in univariate time series, leveraging the non seasonal component, and also an extra set of parameters P, Q, D, π to account for time series seasonality: P is the order of the seasonal AR term, D the order of the seasonal Integration term, Q the order of the seasonal MA term and π the time span of the seasonal term. Overall, the SARIMA model, denoted SARIMA(p,d,q)(P,D,Q)π, has the following form:

$$\phi_p^k(B)\Phi(B^S)(1-B)^d(1-B^\pi)^D y_t^k = \theta_q(B)\Theta_Q(B^\pi)\varepsilon_t^k \tag{3}$$

where B is the backward shift operator, π is the season length, ε_t^k is the estimated residual at time t and with:

$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$
$\Phi(B^\pi) = 1 - \Phi_s B^\pi - \Phi_{2\pi} B^{2\pi} - \cdots - \Phi_{P\pi} B^{P\pi}$
$\theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$
$\Theta_Q(B^\pi) = 1 + \Theta_s B^\pi + \Theta_{2s} B^{2\pi} + \cdots + \Theta_{Q\pi} B^{Q\pi}$

Vector ARIMA (VARMA) process - Contrary to the (S)ARIMA model, which is fitted for univariate time series, VARMA deals with multiple time series that may influence each other. For each time series, we regress a variable on w_i lags of itself and all the other variables and so on for the q parameter. Given k time series $y_{1t}, y_{2t}, ..., y_{kt}$ expressed as a vector $Vt = [y_{1t}, y_{2t}, ..., y_{kt}]^T$, VARMA(p,q) models is defined by the following Var and Ma models:

$$\begin{bmatrix} y_{1t} \\ y_{2t} \\ . \\ . \\ . \\ y_{Kt} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ . \\ . \\ . \\ \mu_k \end{bmatrix} + \begin{bmatrix} \phi^1_{1,1} & \cdots & \phi^1_{1,k} \\ \phi^1_{2,1} & \cdots & \phi^1_{2,k} \\ . & . & . \\ . & . & . \\ . & . & . \\ \phi^1_{k,1} & \cdots & \phi^1_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ . \\ . \\ . \\ y_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} \phi^{w_i}_{1,1} & \cdots & \phi^{w_i}_{1,k} \\ \phi^{w_i}_{2,1} & \cdots & \phi^{w_i}_{2,k} \\ . & . & . \\ . & . & . \\ . & . & . \\ \phi^{w_i}_{k,1} & \cdots & \phi^{w_i}_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-w_i} \\ y_{2,t-w_i} \\ . \\ . \\ . \\ y_{k,t-w_i} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \\ . \\ . \\ . \\ \varepsilon_{kt} \end{bmatrix}$$

$$\begin{bmatrix} y_{1t} \\ y_{2t} \\ . \\ . \\ . \\ y_{Kt} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ . \\ . \\ . \\ \mu_k \end{bmatrix} + \begin{bmatrix} \theta^1_{1,1} & \cdots & \theta^1_{1,k} \\ \theta^1_{2,1} & \cdots & \theta^1_{2,k} \\ . & . & . \\ . & . & . \\ . & . & . \\ \theta^1_{k,1} & \cdots & \theta^1_{k,k} \end{bmatrix} \begin{bmatrix} \varepsilon_{1,t-1} \\ \varepsilon_{2,t-1} \\ . \\ . \\ . \\ \varepsilon_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} \theta^q_{1,1} & \cdots & \theta^q_{1,k} \\ \theta^q_{2,1} & \cdots & \theta^q_{2,k} \\ . & . & . \\ . & . & . \\ . & . & . \\ \theta^q_{k,1} & \cdots & \theta^q_{k,k} \end{bmatrix} \begin{bmatrix} \varepsilon_{1,t-q} \\ \varepsilon_{2,t-q} \\ . \\ . \\ . \\ \varepsilon_{k,t-q} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \\ . \\ . \\ . \\ \varepsilon_{kt} \end{bmatrix} \tag{4}$$

where μ_i is a constant vector, the $k \times k$ matrices, denoted $\phi^r_{i,j}$ and respectively $\theta^r_{i,j}$ (with $i, j = 1, \cdots, k$ and $r = 1, \cdots, p$) are the model parameters, the vector $y_{k,t-i}$ (with $i = 1, \cdots, p$) correspond to the lagged values, vector $\varepsilon_{i,t-u}$ (with $i = 1, \cdots, k$ and $u = 1, \cdots, q$) represents random shocks and ε_{it} (with $i = 1, 2, ..., k$) is the white noise vector.

In summary, the proposed anomaly detection system relies on ARIMA, SARIMA and VARIMA that predict the future behaviour on a regular basis, i.e., during the consecutive time periods $[T_1, T_2], \cdots [T_i, T_{i+1}], \cdots$. In particular, the prediction method further utilises several expanding windows to support anomaly detection at different resolutions. At T_i (with $i > 0$), the resulting predictive models $\hat{\mathcal{M}}^i = \hat{\mathcal{M}}^i(\{\mathcal{Y}\}^{T_i}_{t=T_i-W_1}), \cdots, \hat{\mathcal{M}}^i(\{\mathcal{Y}\}^{T_i}_{t=T_i-W_j}), \cdots$. makes a prediction of the behaviour over the next period of time $[T_i, T_{i+1}]$. For each iteration step T_i (with $i \geq 1$), the complexity[3] associated with forecasting the values with ARIMA, SARIMA and VARIMA for all the spanning windows corresponds to:

$$\sum_{j=1}^{J} (w_j + q_j)^2 (1 + K^2) + (w_j + q_j + P_j + Q_j)^2 \tag{5}$$

[3] Complexity can be reduced by distributing and paralleling [6].

As a forecast is performed for each window, this implies that the more windows there are, the more expensive the forecast becomes.

3.2 Expanding Windows

In order to control the forecasting cost associated with handling several expanding windows, the windows management problem then amounts to (i) determine when a new expanding windows needs to be added and (ii) suppress an existing expanding window if needed. The design of the expanding windows management is such that it favours the forecasting with expanding windows that produce the fewest forecast errors while privileging the less computationally demanding ones in the case of an error tie. A novel expanding window starts if an existing expanding windows provides erroneous predictions (i.e. the prediction error is greater than a given threshold). If required (i.e. the number of windows is too large and reaches the desired limit), this addition leads to the deletion of another window.

3.3 Threshold-based Anomaly Detection

The anomaly detection process is periodically triggered at time T_i (with $T_i \geq T_1$) considering the three predictive models $\hat{\mathcal{M}}^i = \hat{\mathcal{M}}^i_{ARIMA}, \hat{\mathcal{M}}^i_{SARIMA}, \hat{\mathcal{M}}^i_{VARMA}$. In particular, a subset of values $\mathcal{A}_i \subset y_{t_{t=T_i}}^{T_{i+1}}$ is defined as anomalous if there exists a noticeable difference between the observed value y^k_t and one of the forecast values at time t in $\hat{\mathcal{M}}^i = \hat{\mathcal{M}}^i(\{\mathcal{Y}\}^{T_i}_{t=T_i-W_1}), \cdots, \hat{\mathcal{M}}^i(\{\mathcal{Y}\}^{T_i}_{t=T_i-W_J})$. In one of the given models, a noticeable difference between the observation value y^k_t and the forecasted values $\hat{y^k}_t$ at time t (with $T_i \leq t \leq T_{i+1}$) is greater than a threshold. The threshold is calculated using to the so-called three sigma rule [7], which is a simple and widely used heuristic that detects outlier [8]. Other metrics such as the one indicated in [3] could be easily exploited. Based on all the prediction errors $\{\epsilon^k_t\}^{T_i}_{t=T_i-w_i}$ observed during $[T_i - w_i, T_i]$ where $\epsilon^k_t = |y^k_t - \hat{y}^k_t|$, the threshold is defined as:

$$\delta_{w_i}(T_i) = \alpha\ \sigma(\epsilon^k_t) + \mu(\epsilon^k_t) \tag{6}$$

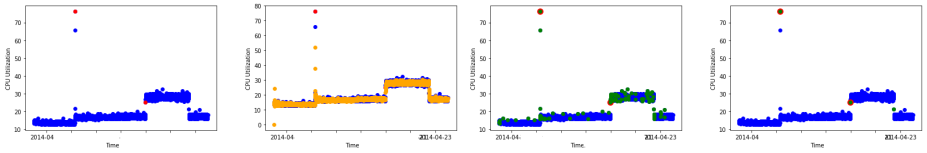
α is a coefficient that can be parameterised based on the rate of false positive/negative observed/expected and $\sigma(\epsilon^k_t)$ and resp. $\mu(\epsilon^k_t)$ correspond to the standard deviation and resp. mean of the prediction error.

4 Assessment

Our solution supports the forecasting along with anomaly detection, provided relevant measurements (a.k.a time series). The proposed solution is evaluated relying on a public dataset, which contains data provided by a monitored cloud-native streaming service. The Numenta Anomaly Benchmark (NAB) dataset[4] corresponds to a labelled dataset, i.e. the dataset contains anomalies for which

[4] https://www.kaggle.com/boltzmannbrain/nab

the causes are known. The dataset depicts the operation of some streaming and online applications running on Amazone cloud. The dataset reported by the Amazon Cloud watch service includes various metrics, e.g., CPU usage, incoming communication (Bytes), amount of disk read (Bytes), ect. Our prototype



(a) normal observations (blue), known anomalies (red)

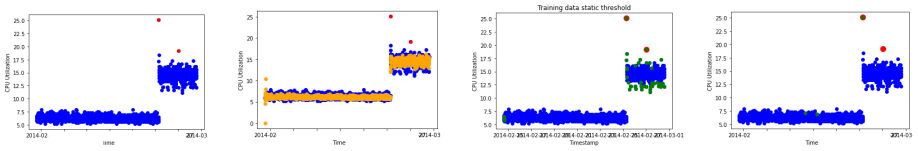
(b) forecast values (orange), known anomalies (red)

(c) detected anomalies (green) with static threshold

(d) detected anomalies (green) with dynamic threshold

Fig. 1: **Observations versus forecast measurement** - CPU utilisation of cloud native streaming service during 23 days.

implementation is focused on the preprocessing of the monitored data, forecasting and detection of anomalies. The prototype requires a Python environment as well panda[5], a third-party packages handling times series and data analytic. Our detector proceeds as follows. filtered and converted into an appropriate format. Then, measurements are properly scaled using Min-Max normalisation [9] of the features. As suggested by Box and Jenkins, the ARIMA model along with their respective (hyper)parameters are established. Finally, anomalies are detected. Relying on the dataset and our prototype, we evaluate the performances associ-



(a) Observations (blue), known anomalies (red)

(b) Forecasted values (orange), known anomalies (red)

(c) Detected anomalies (green) with static threshold

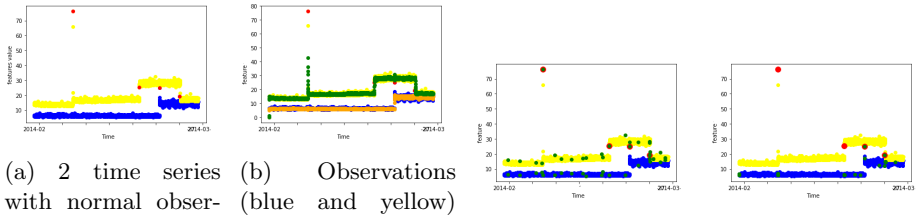
(d) Detected anomalies (green) with dynamic threshold

Fig. 2: **Observations versus forecast values** - CPU utilisation of cloud native streaming service during 1 month.

ated with the proposed anomaly detector. We consider two time frames lasting 23 days (Fig. 1 and 3) and one month (Fig. 2) during which labelled anomalies (red points) are detected (green points in Figures 1c, 1d, 2c, 2d, 3c and 3d) or not. As

[5] https://pandas.pydata.org

expected, forecast values (orange points in Figures 1b and 2b) are conveniently close to the normal observations (blue points). In both cases, anomalies are not always distant from both the normal values (blue points), which makes anomaly detection challenging even if in both cases they are adequately detected. With a dynamic threshold (Figures 1d and 2d), the number of false positives (green points not circled in red in Fig. 1d and 1c) is negligible comparing to a static threshold (Fig. 1c and 2c) that involves a very high false positive rate. When we



(a) 2 time series with normal observations (blue, yellow) and anomalies (red)

(b) Observations (blue and yellow) versus forecast observations (orange and green)

(c) Anomaly detection (green) with static threshold

(d) Anomaly detection (green) with dynamic threshold

Fig. 3: **Multivariate Forecast**

focus on a multivariate prediction and detection (Fig. 3), we see that the parameterization of the threshold plays a significant role in the detection accuracy and in the rate of false positives and false negatives. Comparing to a static threshold, a dynamic threshold constitutes a fair compromise between a accurate detection and an acceptable false positive rate.

5 Related work

Anomaly detection is an long-standing research area that has continuously attracted the attention of the research community in various fields. The resulting research on anomaly detection is primarily distinguished by the type of data processed for anomaly detection and the algorithms applied to the data to perform the detection. The majority of the works deals with temporal data, i.e., typically discrete sequential data, which are univariate rather than multivariate: in practice, several time series (concerning e.g. CPU usage, memory usage, traffic load, etc.) are considered and processed individually. Based on each time series, traditional approaches [10] typically apply supervised or unsupervised techniques to solve the classification problem related to anomaly detection. They construct a model using (un)supervised algorithms, e.g., random forests, Support Vector Machine (SVM), Recurrent Neural Networks (RNNs) and its variants including Long Short-Term Memory (LSTMs) [11–13] and deep neural network (DNN).

Recently, another line of research that have been helpful in several domains, is to analyse time-series to predict their respective behaviour. Then, an anomaly is

detected by comparing the predicted time series and the observed ones. To model non-linear time series, Recurrent Neural Networks (RNNs) and some variants, e.g. Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTMs) [14] have been studied. Filinov et. al [14] use a LSTM model to forecast values and detect anomalies with a threshold applied on the MSE. Candielieri [15] combines a clustering approach and support vector regression to forecast and detect anomalies. The forecasted data are clustered ; then anomaly is detected using Mean Absolute Percentage Error. In [5], Vector Auto Regression (VAR) is combined with RNNs to handle linear and non-linear problems with aviation and climate datasets. In addition, a hybrid methodology called MTAD-GAT [16] uses forecasting and reconstruction methods in a shared model. The anomaly detection is done by means of a Graph Attention Network. The works mentionned above rely on RNNs that are non-linear models capable of modelling long-term dependencies without the need to explicitly specify the exact lag/order. In counterpart, they may involve a significant learning curve for large and complex models. Furthermore, they are difficult to train well and may suffer from local minima problems [17] even after carefully tuning the backpropagation algorithm. The second issue is that RNNs might actually produce worse results than linear models if the data has a significant linear component [31]. Alternatively, autoregressive models, e.g. ARIMA, Vector Autoregression (VAR) [5] and latent state based models like Kalman Filters(KF) have been studied. Time series forecasting problems addressed in the literature, however, are often conceptually simpler than many tasks already solved by LSTM.

For multivariate time series, anomaly is detected by comparing the predicted time series and the observed ones. To predict the future values, several algorithms are employed. Filinov et. al [14] use a LSTM-based model to forecast values and detect anomalies with a threshold on the MSE. Candielieri [15] combines clustering and support vector regression to forecast and detect anomalies: forecasted values are mapped into clusters and anomalies are detected using Mean Absolute Percentage Error. R2N2 [5] combines the traditional Vector Auto Regression (VAR) and RNNs to deal with both linear and non linear problems in the aviation and climate datasets. An hybrid methodology [16] uses forecasting and reconstruction methods in a shared model while anomaly detection is done with Graph Attention Network.

6 Conclusion

Anomaly detection plays a crucial role on account of its ability to detect any inappropriate behaviour so as to protect every device in a cloud including equipment, hardware and software, by forming a digital perimeter that partially or fully guards a cloud. In this article, we have approached the problem of anomaly detection and introduced an unsupervised anomaly detection system that leverages a family of statistical models to predict the behaviour of the softwarised networking system and identify deviations from normal behaviour based on past observations. Existing solutions mostly exploit the whole set of historical data

for model training so as to cover all possible patterns spanning time. Nonetheless, such a detection approach may not scale and performance of these models tend to deteriorate as the statistical properties of the underlying data change across time. We address this challenge through the use of expanding windows with the aim of making predictions based on a controlled set of observations. In particular, several expanding time windows capture some complex patterns that may span different time scales (e.g. long term versus short terms patterns), and, deal with changes in the cloud behaviour. Following, we have implemented and experimented our solution. Our prototype contributes to enhancing the accuracy of the detection at a small computational cost.

References

1. G. E. Box and G. M. J. R. Reinsel, *Time series analysis: forecasting and control*. John Wiley, 2015.
2. B. Hammi, G. Doyen, and R. Khatoun, "Toward a source detection of botclouds: A pca-based approach," in *IFIP International Conference on Autonomous Infrastructure, Management and Security (AIMS)*, 2014.
3. A. Huet, J.-M. Navarro, and D. Rossi, "Local evaluation of time series anomaly detection algorithms," in *Conference on Knowledge Discovery & Data mining*, 2022.
4. W. Yoo and A. Sim, "Time-series forecast modeling on high-bandwidth network measurements," *Journal of Grid Computing*, vol. 14, 2016.
5. H. Goel, I. Melnyk, and A. Banerjee, "R2n2: residual recurrent neural networks for multivariate time series forecasting," https://arxiv.org/abs/1709.03159, 2017.
6. X. Wanga, Y. Kanga, R. Hyndmanb, and al., "Distributed arima models for ultra-long time series," *International Journal of Forecasting*, June 2022.
7. F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, 1994.
8. L. Rüdiger, "3sigma-rule for outlier detection from the viewpoint of geodetic adjustment," *Journal of Surveying Engineering*, pp. 157–165, 2013.
9. A. Zheng and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc., 2018.
10. D. Gümüsbas, T. Yildirim, A. Genovese, and F. Scotti, "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems," *IEEE Systems Journal*, vol. 15, no. 2, 2020.
11. T. Kim and S. Cho, "Web traffic anomaly detection using c-lstm neural networks," *Expert Systems with Applications*, vol. 106, pp. 66–76, 2018.
12. Y.Su, Y. Zhao, C. Niu, and al., "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *25th ACM International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2828–2837.
13. A. Diamanti, J. Vilchez, and S. Secci, "Lstm-based radiography for anomaly detection in softwarized infrastructures," in *International Teletraffic Congress*, 2020.
14. P. Filonov, A. Lavrentyev, and A. Vorontsov, "Multivariate industrial time series with cyber-attack simulation: fault detection using an LSTM-based predictive data model," https://arxiv.org/abs/1612.06676, 2016.
15. A. Candelieri, "Clustering and support vector regression for water demand forecasting and anomaly detection," *Water*, vol. 9, no. 3, 2017.
16. H. Zhao, Y. Wang, J. Duan, and al., "Multivariate time-series anomaly detection via graph attention network," in *International Conference on Data Mining*, 2020.
17. M. Y. S. Uddin, A. Benson, G. Wang, and al., "The scale2 multi-network architecture for iot-based resilient communities," in *IEEE SMARTCOMP*, 2016.