Autoregressive Moving Models in anomaly and threat detection

Thesis for the Statistics course
Sapienza - University of Rome
Cybersecurity

Paolo Lucchesi · 1765134

December 25, 2025

Contents

1 Introduction

The world of cyber threats has evolved massively. The complexity, sophistication and diversification of modern cyberattacks called for the development of novel, technique-agnostic detection methods in the field of cybersecurity.

In this context, statistics plays a crucial role. Indeed, in many scenarios it allows to build detection systems capable of identifying threats regardless of their technical nature, and in the presence of usual, legit activity.

In particular, statistical *online algorithms* allow building real-time, numerically stable, noise-resistant, computationally efficient models for fast and reliable threat detection.

Statistical *Autoregressive Moving Average* models can be very effective in achieving such goal. They are often relatively easy to build and deploy, customizable via parameterization, and can take into account contextual stochastic processes (e.g. noise or errors). Moreover, they can be built to be self-adjusting to some extent, especially when they are needed to evolve in relation to time.

1.1 Solutions for anomaly and threat detection

Threat detection is a key field in cybersecurity. Many solutions have been developed to tackle the necessity of early detection in order to preemptively deal with ongoing threats. Many of such solutions make extensive use of statistical algorithms.

Intrusion Detection Systems are a perfect example in this context. An IDS is a piece of software that monitors a system or network in order to detect suspicious activity or policy violations. Such solutions often work with high volumes of data, so the algorithms used must be efficient and scalable.

Examples of widely used Intrusion Detection Systems are Wazuh (OSSEC), Zeek (formerly Bro IDS) and Suricata.

2 Theoretical background

In this section we will give a brief theoretical background necessary to properly understand the presented research works.

2.1 Online algorithms

An online algorithm is a statistical algorithm defined in terms of a recurrence relation. In practice, this means an online algorithm can compute a metric over a huge number of samples by performing small, incremental updates. In the context of computer science, this brings a number of advantages.

First, online algorithm offer far stronger numerical stability compared to batch algorithms. There is a much higher risk of computational errors in floating-point operations if big and small numbers are used together. Using incremental updates hugely mitigates such risk.

Moreover, a distinctive characteristic of online algorithms is that updating the metric is usually computationally cheap. This enables the construction of fast and efficient real-time detection systems that can process events incrementally.

Lastly, online algorithms are likely to not keep explicit memory of past samples, often even being *inline* by nature. For example, an online algorithm to compute the *mean* on a dataset numerical attribute does not need to keep memory of all the values of such attribute, and just needs a fixed amount of memory to work with any number of values.

2.2 Autoregressive Moving Average

The *Autoregressive Moving Average* (*ARMA* from now on) is a statistical metric used to identify trends in time-series data. The key idea is that, in contrast to simple average metrics, the influence of past observed data gradually fades, so newer data weights more on the metric value. ARMA models are built on top of *Autoregressive Models* (i.e. *AR* models) and *Moving Average* models (i.e. *MA* models).

A very clear and high-quality introduction to ARMA models is given in a distinguished lecture by the Berkeley University.

2.2.1 Autoregressive models

An AR model represents a stochastic differential equation in which the output variable is linearly dependant (in time) on the previous deterministic and random input.

The formal definition for an AR model of order p is given below:

Where $\phi_1, \dots, \phi_p$ are fixed parameters, $p \in \mathbb{N}$ and ω_* are stochastic parameters.

2.2.2 Moving Average models

MA models can be seen as a complement of AR models. Unlike their counterpart, MA models evolve linearly with the *error* (i.e. the stochastic process)

instead of a deterministic input.

The formal definition for an MA model of order q is given below:

Where $\theta_1, \dots, \theta_p$ are fixed parameters, $q \in \mathbb{N}$ and ω_* are stochastic parameters.

2.2.3 Autoregressive Moving Average models

On top of the AR(p) and MA(q) definitions, the *ARMA model* can be formalized:

As by definition ARMA models take into account the influence of stochastic processes (e.g. noise), they are often suitable to describe realistic scenarios.

2.3 Exponential Weighted Moving Average

The *Exponential Weighted Moving Average* (*EWMA* from now on) is a statistical metric which can be defined as a special case of the ARMA. Specifically, it is a completely deterministic ARMA metric that does not take into account the potential influence of stochastic processes.

An extensive theoretical explaination on how EWMA works and can be implemented in moving models is given in an awesome article from the Stanford University.

Let $x = x_1, x_2, \dots, x_n$ be a vector (i.e. our data). Let $\beta \in (0, 1)$ be the *forgetting factor*. The *Exponential Weighted Moving Average* is defined as follows, as a recurrent relation:

where the *normalization constant* α_t is defined as:

This EWMA definition allows for efficient, online implementations.

The EWMA can also be defined in a simplified manner with a fixed $alpha$ parameter $\forall t$:

As already stated, the EWMA can be defined in terms of an ARMA$(1, 0)$:

In order to eliminate the stochastic process ω_t and obtain the EWMA definition given before, let:

With those identifications, we can therefore obtain the exact definition of an EWMA model. Such obtained definition can be turned into the recurrent relation defined at the beginning of this section by induction.

3 Implementations and practical use

In this section we will explore how ARMA models can be used in practice. We will give some example implementations, and we will also see how modern and widely used cybersecurity solution can make use of them. A particular focus will be dedicated to the EWMA-based models, which are the simplest yet powerful ARMA model to use.

3.1 EWMA example implementation

In order to understand how EWMA can work in a practical application, we give a basic JavaScript implementation below:

```
export class Ewma {
    constructor({ beta, current = 0, t = 0, increment = 1 }) {
        this.beta = beta
        this.current = current
        this.t = t
        this.increment = increment
    }

    update(value, tau = this.t + this.increment) {
        const alphaT = this.alpha()
        const alphaTau = this.alpha(tau)
        this.current =
            alphaTau / alphaT * this.beta * this.current + alphaTau * value
        return this.current
    }

    alpha(t = this.t) {
        return (1 - this.beta) / (1 - Math.pow(beta, t))
    }

    batch(data = []) {
        return data.map((x, i) =>
            this.update(value, i + this.t + this.increment))
    }
}
```

The class defined above can be used like so:

```
const ewma = new Ewma({ beta: 0.5 })
const data = [1, 1.1, 1.2, 2, 1.5, 2]

// Batch use: Perform EWMA computation over a given dataset
let ewmaValues = ewma.batch(beta, data)

// Incremental use: Update the EWMA value online with a new datum
const updated = ewma.update(1)
ewmaValues.push(updated) // Do something with the updated EWMA value
```

3.2 EWMA usage in threat detection

In the context of cybersecurity, the EWMA is a useful metric to early detect anomalies, especially for network traffic and CPU usage. More specifically, it can be used to detect strong and deviations from an expected trend.

EWMA and EWMA-like algorithms can be easily implemented in widely used IDS solutions. In Zeek, we can compute real EWMA metrics using its own scripting language. For example, to detect network traffic anomaly per-host:

```
global ewma: table[addr] of double = table();
global alpha = 0.2;
global threshold_factor = 5;

event connection_state_remove(c: connection) {
    local host = c$id$orig_h;
    local x = c$orig$size;

    if (host !in ewma)
        ewma[host] = x;
    else
        ewma[host] = alpha * x + (1 - alpha) * ewma[host];

    if (x > threshold_factor * ewma[h])
        NOTICE([$note=Traffic_Anomaly,
                $msg=fmt("Traffic anomaly detected for host: %s", host)]);
}
```

We can notice that in this implementation we used a slightly, simplified EWMA definition with a fixed value α. This works perfectly fine in practical applications.

In other solutions not supporting real EWMA estimation, we can mimic an EWMA-like metric using frequency-based analysis. For example, in Wazuh we can monitor the login failure rate of a given service and signal an anomaly if the actual value exceeds the expected trend:

```
<rule id="123456" level="7">
  <if_matched_sid>1234</if_matched_sid>
  <frequency>10</frequency>
  <timeframe>60</timeframe>
  <description>Possible brute-force (rate anomaly)</description>
</rule>
```

4 Research

Lately, sophisticated detection models making use of ARMA models, EWMA and EWMA-like metrics have been developed.

4.1 ARMA for anomaly detection in HTTP applications

The article *Hourly Network Anomaly Detection on HTTP Using Exponential Random Graph Models and Autoregressive Moving Average* (R. Li, M. Tsikerdekis, A. Emanuelson - 2022) formalizes a model for anomaly detection in structured network infrastructures. The model has been used to detect suspicious behaviors in the context of HTTP applications. Citing the article itself:

> We use exponential random graph models (ERGMs) in order to flatten hourly network topological characteristics into a time series, and Autoregressive Moving Average (ARMA) to analyze that time series and to detect potential attacks. In particular, we extend our previous method in not only demonstrating detection over hourly data but also through labeling of nodes and over the HTTP protocol. We demonstrate the effectiveness of our method using real-world data for creating exfiltration scenarios.

Exponential Random Graph Models (*ERGM*), in this context, can be used to produce so-called *log-odds coefficients*. Such coefficients are useful to estimate if the state of a network is usual or not.

An ARMA model analogous to the one defined in the theoretical introduction was used to produce predictions that enable to detect suspicious traffic volumes:

> In our approach, the ARMA is trained first; then, it calculates variable-sized prediction windows for a few points into the future; and if the observations fall outside the range, an alert is raised for an anomalous event.

In the article, the authors point out that the formulated model showed quite promising results against exfiltration techniques.

4.2 Enhanced EWMA for false positives reduction

The article An Enhanced EWMA for Alert Reduction and Situation Awareness in Industrial Control Networks (B. Jiang, Y. Liu et al. - 2022) shows how an enhanced version of EWMA can be used to drastically reduce the number of alerts raised by IDSs. Citing the article itself:

> IDSs typically generate a huge number of alerts, which are time-consuming for system operators to process. Most of the alerts are individually insignificant false alarms. However, it is not the best solution to discard these alerts, as they can still provide useful information about network situation. Based on the study of characteristics of alerts in the industrial control systems, we adopt an enhanced method of exponentially weighted moving average (EWMA) control charts to help operators in processing alerts.

The work is developed in the context of *Industrial Control Networks* (*ICNs*), in which a huge number of alerts is triggered, most of them being false positives. The key idea is that an EWMA model is used on the number of alerts itself to determine if, at a given time, the volume and type of them is usual (and therefore not representing suspicious activity) or not.

4.2.1 Enhanced EWMA

The *Enhanced EWMA* sets upper and lower control limits (i.e. *UCL* and *LCL*) to mitigate EWMA over-adjusting. Such values are used as thresholds in outlier detection. Formally, they are defined like below:

First, U is the *control limit factor*, i.e. a parameter scaling acceptable ranges. $e_p(i)$ is the *estimate prediction error* for the actual prediction error $e(i+1)$ (i.e. the *one-step-ahead prediction error*) and is defined as follows:

Where σ_e^2 is the variance of the prediction error. This definition of e_p actually makes the EWMA model *residual-based*.

4.2.2 Experiments and results

Experiments and performance measurements were conducted over a real, big, reliable dataset:

> We obtained nearly 600,000 alerts generated by the IDS of a power grid company's automation control system in June, 2021. This is a typical communication network in the ICS scenario where encryption and isolation measures are adopted to the communication between hosts. The network behaviors of the hosts are strictly restricted, and rigorous signatures are applied to the IDS for the sake of security.

Regarding the results and evaluations, the authors immediately pointed out the intrinsic evaluation difficulties:

> As the real-world data set lacks labels for malicious network attacks and other security events, it is difficult to evaluate our method using metrics like accuracy, precision and recall.

However, the actual volume of alerts raised by the model was drastically reduced. Moreover, cases in the resulting data were spot; in such cases, it was quite clear that the level of alerts increased drastically, in a much more recognizable way compared to a bare IDS.

4.3 ϕ-entropy and EWMA for anomaly detection

The article Self-adaptive Threshold Traffic Anomaly Detection Based on ϕ-Entropy and the Improved EWMA Model (M. Deng, B. Wu - 2020) formalizes an enhanced EWMA model featuring self-adaptive threshold based on ϕ-entropy.

The article is very technical, formal analysis of a novel EWMA model for anomaly detection. It delivers own proved theorems and is a wonderful piece

of research.

In the abstract, the authors clearly explain the rationale behind their work:

> Most of traffic anomaly detection algorithms use a fixed threshold for anomaly judgment, but these methods cannot keep a high detection accuracy in numerous cases. Aiming at this problem, this paper proposes a method to generate a self-adaptive threshold based on the improved Exponentially Weighted Moving Average (EWMA) model. The method predicts the value of phi-Entropy at the next moment and further generate the threshold. Results of simulation and experiment show that the algorithm can effectively detect abnormal network traffic

4.3.1 Model description

The idea behind the proposed model is quite clever and creative. Again, the article itself provide a succinct and crystal clear explanation:

> The phi-Entropy is used to describe the autocorrelation of network traffic. The improved EWMA model is used to predict the phi-Entropy at the next moment and further generate the adaptive threshold. The obtained threshold is used to determine whether the traffic at the next moment is anomaly traffic or not.

The improvement of the classic EWMA model is achieved with two modifications:

1. a slide-window model using just a few recent data is used to compute the next predicted value
2. the α parameter is recomputed at each prediction; the computation is done by interpolation of two intermediate values α_{high} and α_{low}, which are based on the speed of data change

The model thresholds can be adapted at every prediction as below:

Where \sigma is the floating range and $\overline{y}$ is the predicted value. The traffic can be evaluated in the following way:

4.3.2 Experiments and results

In order to evaluate the performance of the model, the following metrics have been used:

- *ACC*: Accuracy
- *DR*: Detection Rate
- *FAR*: False Alarm Rate

With various parameters, the model showed very good results:

- The estimated *ACC* was always over 90% but for one time
- The estimated *DR* was consistently around 98%, being at 97% just once
- The estimated *FAR* was always under 5%

Complete results are given below with the corresponding parameters.

w	β	α	ACC/%	DR/%	FAR/%
5	0.5	0.6	93.81	98.71	3.61
10	0.5	0.6	94.01	98.54	3.42
20	0.5	0.6	95.17	98.10	3.17
5	1.0	0.6	93.01	98.91	3.87
10	1.0	0.6	93.71	98.11	3.71
20	1.0	0.6	94.21	97.13	3.65
5	1.5	0.6	93.17	99.11	4.01
10	1.5	0.6	94.31	98.67	3.94
20	1.5	0.6	95.71	98.77	3.83
20	1.5	0.5	93.81	98.71	4.07
20	1.5	0.4	90.74	98.66	4.51
20	1.5	0.3	87.64	98.70	4.55

The authors claimed that their model actually outperformed existing alternatives (such as joint-entropy or Shannon entropy based models) in both *DR* and *FAR*.

Algorithm	DR/%	FAR/%
Based on joint-entropy	95.30	4.76
Based on entropy and DNN	93.78	6.21
Based on Shannon entropy	93.50	5.5
Proposed algorithm	98.71	4.07

5 Conclusions

We have shown how Autoregressive Moving Average models can play a key role in anomaly and threat detection. After giving a theoretical introduction, we have shown how such models can be used in both existing, widely adopted solutions and cutting-edge research applications.

For the latter, we have shown how ARMA models can be used to both improve the detection process and overall accuracy and enhance existing anomaly detection solutions. Of course, in this context we have shown just a small portion of existing works. The research ecosystem in this field is vast and continuously evolving.