IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received June 1, 2021, accepted July 5, 2021, date of publication July 16, 2021, date of current version July 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3097834

On Aggregation and Prediction of Cybersecurity Incident Reports

MIGUEL V. CARRIEGOS[1], **ÁNGEL L. MUÑOZ CASTAÑEDA**[1], **M. T. TROBAJO**[1], **AND DIEGO ASTERIO DE ZABALLA**[2]

[1]Departamento de Matemáticas, Universidad de León, 24007 León, Spain
[2]Instituto de Ciencias Aplicadas a Ciberseguridad, Universidad de León, 24007 León, Spain

Corresponding author: Miguel V. Carriegos (miguel.carriegos@unileon.es)

This work was supported in part by the Spanish Institute for Cybersecurity (INCIBE) under Contract Adenda 3, and in part by the Ciberseguridad, Aplicaciones, Fundamentos y Educación (CAFE) and NARVIC research groups.

ABSTRACT The study of cybersecurity incidents is an active research field. The purpose of this work is to determine accurate measures of cybersecurity incidents. An effective method to aggregate cybersecurity incident reports is defined to set these measures. As a result we are able to make predictions and, therefore, to deploy security policies. Forecasting time-series of those cybersecurity aggregates is performed based on Koopman's method and Dynamic Mode Decomposition algorithm. Both techniques have shown to be accurate for a wide variety of dynamical systems ranging from fluid dynamics to social sciences. We have performed some experiments on public databases. We show that the measure of the risk trend can be effectively forecasted.

INDEX TERMS Cybersecurity, extended dynamic mode decomposition, Koopman operator, time series forecasting, threat prediction.
MSC[2010]: 00A72 (General methods of simulation), 93A30 (Mathematical modeling), 65F20 (Overdetermined systems and pseudo-inverses), 68T10 (Pattern recognition), 37N40 (Dynamical systems in optimization and economics), 62-07 (Data analysis).

I. INTRODUCTION

Incidents of cybersecurity are ever-present threats compromising cybersystems. These events provoke social and economic losses by outwitting legitimate security systems. Hence, it is needed to detect, classify, and even predict these kinds of events to protect ourselves from damages they cause.

Several works have studied cybersecurity incidents and threats from different perspectives; see [33], [40], and references therein. They focus on analysis, detection, and prevention, but no prediction schemes have been provided to set up proactive measures to avoid the damage in advance.

The first problem one faces performing a predictive cybersecurity model is to define a cybersecurity threat. Several taxonomies have been proposed in the literature [16], [35], but there is no consensus on the topic. Hence, one needs to fix a concrete classification before setting up the model.

How can we decide whether a cybersecurity event is currently taking place or not in a particular system? Suppose we are interested in events concerning only our system connected to the Internet or another communication network. In that case, the usual scenario is that we have a firewall between our system and the Internet; thence, a threat would happen as an incoming message (a piece of code, an access request, etcetera) which is, in the end, an incoming chain of words in some formal language. We only need to learn what chains of words are likely to appear in malicious traffic but not regular traffic. Once malicious chains are known, one needs to design countermeasures to prevent new attacks or malfunctions. Often, we classify malicious traffic into several types of threats and design a countermeasure for each class. Note that the definition of an incident and taxonomies to classify varies from different bibliography sources [16].

Several techniques dealing with the classification of cybersecurity threats deal with the study of reports and often applying expert knowledge or some automatization or even machine learning algorithms [24], [33], [40].

Ideally, we demand a tool capable of giving reports of the level of risk and predict a probability of being attacked in the near future. This tool may be based on our records of

The associate editor coordinating the review of this manuscript and approving it for publication was Grigore Stamatescu.

activity and expert knowledge [24]. However, that type of tool is too local in its design though somewhat useful. It is needed to improve capabilities to anticipate threats and refine procedures to extend the horizon of valuable predictions.

Remark 1: Predicting the future behavior of some *a priori* selected measures of cybersecurity threats yields the ability to design alerts and countermeasures within systems. However, trying to predict when and where a concrete cybersecurity threat is going to happen is nonrealistic, even in the short term, because of deploying, disseminate, or exploding a threat is, in the end, a human decision. A more realistic task is to predict some aggregate measures like the approximate amount of threats that will occur in a specific (near future) time window or to predict whether the amount of threats is going to grow up or not. A solution to this last problem would lead to a predictive model of the risk trend instead of the risk itself. This is a coarser approach to the problem, though still very useful.

In this work, we approach the construction of a mathematical model to predict the risk trend for a cybersecurity incident to take place. This is a data-driven approach [7]; thus, we suppose we have some cybersecurity data. We deal with public malware databases [19] in our experiments.

There are many approaches to the term ''cybersecurity incident'' in the literature [16], [35], or [40]; thus, the prediction is sensitive to the definition. However, this work deals with forecasting reports of cybersecurity threats registered by outside sources, whatever the definition of ''cybersecurity incident'' is used by these sources.

The paper is organized as follows:

- In Section II, we give an introduction to the problem of forecasting cybersecurity aggregates. After stating the main formulations of the problem, a natural way of integrating time-stamped cybersecurity databases into a time series is presented.
- Section III deals with technical issues about time series forecasting. Precisely, Koopman's operator is presented in combination with its data-driven approximation.
- In Section IV, we discuss a Rolling Cross-Validation Scheme and define directional measures for the validation.
- Section V is devoted to running experiments on an actual Cyber Security database. We perform the Cross-Validation scheme presented in the previous section, and we describe the obtained results.
- In Section VI, we describe experiments to improve the dictionary of observables associated with Koopman's operator. Metaheuristic techniques are applied to search inside the dictionary space.
- Finally, in section VIII, we conclude by briefly synthesizing our findings.

It is worth mentioning the work [33] by Pokhrel *et al.* In this paper, the authors perform a Machine Learning approach to forecasting vulnerabilities on desktop operating systems. Our results and those of [33] are related because they also use reports (of vulnerabilities) as data input.

II. AGGREGATES, AGGREGABLE TABLES, AND TIME SERIES OF AGGREGATES

A. CYBERSECURITY THREATS AND AGGREGATES

There are many actors collecting information of users reporting cybersecurity incidents [33], [35], [40]. Typically users make alerts of incidents. These alerts are processed and classified; some are stacked and labeled as some kind of incident. This provides alphanumeric datasets gathering cybersecurity threat activity in some environments. If these datasets contain (and often do) time-stamps, one can forecast the behavior of the dataset and make predictions of near-future behavior of measures of cybersecurity reports.

The problem of forecasting time series of cybersecurity threats data can be stated in at least two ways, which we call ''strong formulation'' and ''weak formulation''.

1) THE STRONG FORMULATION

The strong formulation of the problem of forecasting cybersecurity threats consists in: collecting data concerning cybersecurity activity and deriving a model whose inputs are collected data and whose outputs are measures concerning cybersecurity threats that can be projected to the future and, hence, capable of making predictions about future measures.

This formulation requires integrating several procedures due to cybersecurity data being often obtained in an unstructured format. First of all, data should be in a format that can be parsed. Secondly, data should be classified into at least two classes: legitimate and threat activities. These risky activities could be classified as well into different kinds of threats according to an *a priori* established taxonomy. Finally, data should be transformed into numerical in order to be processed in a mathematical model.

A strong formulation of the problem is given in [40] in their so-called *Data-driven research methodology*. However, no explicit projection to future predictions is considered there.

2) THE WEAK FORMULATION

It is assumed that a database of numerical measures of some kind of cybersecurity threat (malware, IPBoot, or other threats) is given. These measures come labeled with some time-stamp, which is often the case in real scenarios because most of the standards in cybersecurity reporting include some kind of time-stamp into their mandatory information fields [48]–[52]. Then, time-stamps are used to build up a time series of the number of cybersecurity events of some particular class reported to the specific source we are researching. This time series will be called a time series of aggregates (see definition below).

The problem is thus reduced to a weaker form, and it is ready to be processed with some mathematical model.

B. INTEGRATING A TIME SERIES OF AGGREGATES

Definition 1 (Aggregable Table): An aggregable table is an n-row m-column table, $\mathbf{T}$, of data where there exists a distinguished column containing elements of a totally ordered semigroup, usually a time set. This last column is called time-stamp.

Remark 2 (On Databases of Raw Reports): First, we need to settle the question, which systems are of our interest? It could happen that the case of interest is not the system itself but a part of it; the clients' systems; or some systems out of our control but of interest. All of these systems usually report their cybersecurity events.

Then, we establish our sources of cybersecurity incidents. The incidents are reported in some format, usually, a formal language containing some information about the event: type of activity, time-stamp, IP-address, etcetera. All these reports are then stacked in a database of raw reports (for instance, see [48]).

Remark 3 (Gathering an Aggregable Table of Cybersecurity Reports): The raw database needs to be refined in order to get an aggregable table. It is also needed to locate a helpful time-stamp along with the reports. The step where we refine the raw database is crucial to building up an aggregable table. However, it is not straightforward in most cases. The reports might be presented in a different format depending on the source. Time-stamps might be non-congruent among different sources. Even a concrete source might report non-congruent time-stamps because reports could come up with the time when the incident happened, the time when the incident was reported, or even when the report was stacked. In our weak formulation setting, we assume this refining step has already been fulfilled. Hence, we will deal with already refined data, so we only need to fix target features, the time-stamp feature, and construct our table.

$$\mathbf{T} = \text{Ag.Table (Info.Sour., Target.Feat., Stack.Protocol)}$$

Definition 2 (Aggregate Measure on Aggregable Table): Consider an aggregable n-rows $m+1$-columns table $\mathbf{T}$ where rows are refined reports$\{r_1, \dots r_n\}$ and columns are features $\{\tau, f_1, \dots, f_m\}$.[1] The aggregate of table $\mathbf{T}$ for time-lapse $d\tau > 0$ over feature value $f_k = \xi$ is given by:

$$\mathfrak{X}^{d\tau}_{(f_k=\xi)}(t) = \int_{\mathbf{T}} d\tau[f_k = \xi] = \\ := \# \left\{ i \,\middle|\, \begin{array}{c} \pi_k(r_i) = \xi \\ \\ t \cdot d\tau \leq \pi_0(r_i) < (t+1)d\tau \end{array} \right\}.$$

This is a scalar time series depending on $t \in \mathbb{Z}$, and will be called time series of aggregates $\int_{\mathbf{T}} d\tau[f_k = \xi]$. If there is no confusion, we will denote it just by $\mathfrak{X}_\xi(t)$.

Remark 4: The time-lapse $d\tau$ should be (ideally) infinitesimal in the sense of non-standard calculus, see [21], but it is sufficient to assure that $d\tau$ is small enough compared with the whole time-window of the table

$$|\max(\pi_0(r_i)) - \min(\pi_0(r_i))|$$

[1] It is assumed that time-stamp τ is located on the 0-th column and it is an additive semigroup of $\mathbb{R}$

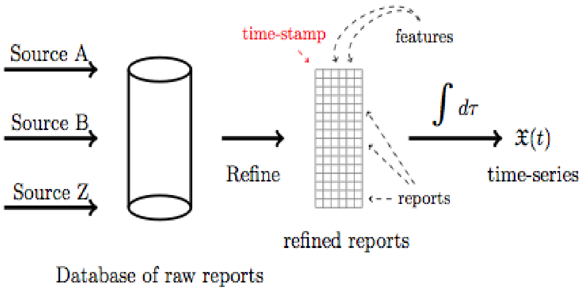


FIGURE 1. Integrating a time series of aggregates from a database of reports (rows). One feature (column) is a time-stamp.

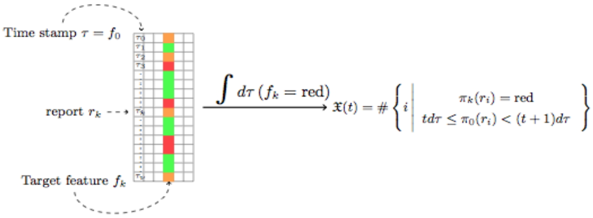

Time stamp $\tau = f_0$
τ_0
τ_1
τ_2
τ_3
report r_k
τ_k
τ_N
Target feature f_k
$\int d\tau\,(f_k = \text{red})$
$\mathfrak{X}(t) = \# \left\{ i \middle| \begin{array}{c} \pi_k(r_i) = \text{red} \\ td\tau \leq \pi_0(r_i) < (t+1)d\tau \end{array} \right\}$

FIGURE 2. Integrating a time-stamped table to a time series of aggregates.

Remark 5: There are other possible aggregate measures on an aggregable table yielding time series of aggregates. For instance, one could focus on some concrete values for several features $f_{k_1} = \xi_1, \dots f_{k_s} = \xi_s$ yielding the time series

$$\int_{\mathbf{T}} d\tau \left[\bigwedge_{j=1}^{s} (f_{k_j} = \xi_j) \right] = \\ = \# \left\{ i \,\middle|\, \begin{array}{c} \bigwedge_{j=1}^{s} (\pi_{k_j}(r_i) = \xi_j) \\ \\ t \cdot d\tau \leq \pi_0(r_i) < (t+1)d\tau \end{array} \right\}$$

In general, if P is any first-order sentence with non-time-related features of the table as arguments, and $d\tau$ is a time-lapse, then one might build up an aggregate on the table as

$$\int_{\mathbf{T}} d\tau P = \# \left\{ i \,\middle|\, \begin{array}{c} P(r_i) = \text{true} \\ \\ t \cdot d\tau \leq \pi_0(r_i) < (t+1)d\tau \end{array} \right\}$$

C. FORECASTING TIME SERIES OF AGGREGATES

Let us assume that we have computed our target time series

$$x(t) = \int_{\mathbf{T}} d\tau P.$$

Our next step is to perform forecasting of such a time series to obtain valuable information to describe the threat environment or even to fed back our security measures or policies.

Remark 6 (Time-Series Forecasting): This is a classical and still very active research field. There are many techniques for forecasting and predicting time series. Hence, there are multiple options to forecasting time series of aggregates of cybersecurity reports. Several recent studies face some points of the problem we state here. Some examples range from the description of advanced persistent threats [24] to the optimization of provisioning in cloud computing [14], and hyper-parameter optimization of machine learning algorithms on vectorial databases [10], and to the detection and removal of redundancies [9].

Remark 7 (Data-Driven Methods): Data analysis is used to set up a linear model to describe the behavior of a target time series $x(t)$. We emphasize the term "data-driven" in order to highlight that no expert knowledge is assumed.

Remark 8 (Artificial Intelligence): The 2017 paper [33] by Pokhrel *et al.* performs a Machine Learning approach to forecasting vulnerabilities on desktop operating systems. Our results and Pokhrel *et al.* results are related because they also use reports (of vulnerabilities) as data input. However, the tools are quite different because our approach is entirely data-driven based. We only use Machine Learning techniques in the last step to develop parameters and optimize some kind of directional measures.

III. DATA-DRIVEN FORECASTING OF TIME SERIES OF AGGREGATES

A. ON KOOPMAN'S OPERATOR

Koopman spectral analysis [6] is an operator-theoretic perspective of dynamical systems, first introduced in the 1930s by B.O. Koopman [25], [26] to study nonlinear dynamics associated with Hamiltonian flows. The Koopman operator is an operator on the infinite-dimensional Hilbert space of observable functions. Recently, Koopman's analysis has made possible interesting breakthroughs in the study of asymptotic dynamics of complex systems [29], [30]. There is a growing interest in operator-theoretic approaches for analyzing dynamical systems based on the Koopman operator [25], [30]. The Koopman operator allows studying the evolution of observables, which are *good* functions of state vectors, in a function space.

Suppose that we have a dataset of cybersecurity events of some kind (via cybersecurity reports, network traffic). We consider the alphanumeric aggregable table **T** of data

$$R = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(m) \end{bmatrix}.$$

Assume also that one has decided the first-order logic sentence P involving the features, and integrated a time series of aggregates

$$x(t) = \int_{\mathbf{T}} d\tau P$$

running on the time-stamp. Then, we face the problem of finding out a dynamic model,

$$x(t+1) = f(x(t)), \tag{1}$$

describing the behavior. The above dynamic model can be obtained through the Koopman operator $\mathcal{K}$, which is defined [41] as an infinite-dimensional linear operator acting on Lebesgue square-integrable observables $g : \mathcal{M} \subseteq \mathbb{R}^{1\times n} \to \mathbb{C}$ such that

$$\mathcal{K}[g](x) = g(f(x)).$$

Hence, the analysis of the above nonlinear dynamics can be lifted to a linear (but infinite-dimensional) regime because the associated Koopman system is set up as follows.

Definition 3 (Koopman System): Given a time series

$$\mathcal{S} = \left(\mathcal{M} = \mathbb{R}^{1\times n}, f, \mathbb{N}\right),$$

consider the $\mathbb{C}$-algebra of Lebesgue square-integrable functions (observables) $\mathcal{F}(\mathcal{M}) = \mathscr{L}^2(\mathcal{M}, \mathbb{C})$ and the Koopman linear system over $\mathcal{F}(\mathcal{M})$ given by

$$\mathrm{Koopman}(\mathcal{S}) = (\mathcal{F}(\mathcal{M}), \mathcal{K}(f), \mathbb{N})$$

where $\mathcal{K}(f) = \mathcal{K}$ is the operator

$$\begin{array}{rccc} \mathcal{K}: & \mathcal{F}(\mathcal{M}) & \longrightarrow & \mathcal{F}(\mathcal{M}) \\ & (\phi : \mathcal{M} \to \mathbb{C}) & \mapsto & \mathcal{K}[\phi] = (\phi \circ f : \mathcal{M} \to \mathbb{C}) \end{array}$$

In terms of dynamic equations, Koopman system $\mathrm{Koopman}(\mathcal{S})$ is

$$\phi_{t+1} = \mathcal{K}[\phi_t]$$

Remark 9: Koopman system associated with a time series is linear due to the linearity of composition. Thus Koopman system is a $\mathbb{C}$-linear dynamical system, but the underlying state-space is infinite-dimensional. Therefore, analyzing a nonlinear dynamical system with finite-dimensional state space is traded by analyzing a linear infinite-dimensional dynamical system through the Koopman operator. This approach has been considered to study complex phenomena employing different implementations ranging from robotic control, image processing, and nonlinear system identification; see [41] and references therein.

B. DATA-DRIVEN APPROXIMATION OF KOOPMAN'S OPERATOR: DYNAMIC MODE DECOMPOSITION

Dealing with concrete experiments and calculations, one needs to obtain a good finite-rank approximation to the linear Koopman operator that estimates the original dynamics.

The Extended Dynamic Mode Decomposition (EDMD) yields a finite-dimensional operator (and hence a matrix K), which approximates Koopman operator $\mathcal{K}$.

Remark 10: Dynamic Mode Decomposition (DMD) was introduced in [38]. It currently has a wide application range in computational fluids and in other research fields where nonlinear behaviors do appear. It may also be applied when the dynamic is unknown or there is a lack of natural laws governing the dynamics. An extended non-algorithmic definition of EDMD was set up in [42]. Suppose that some sampling of an unknown dynamic model (1) is given,

$\{(x_1, y_1), \ldots, (x_m, y_m)\}$. Then the EDMD is performed [42] in terms of the data matrices

$$X = [x_1, \ldots, x_m]\ ,\ Y = [y_1, \ldots, y_m]$$

and it provides the best linear approximation of $\mathcal{K}(f)$ based on our data pairs $\{(x_1, y_1), \ldots, (x_m, y_m)\}$.

Definition 4: Let $\mathcal{M} = \mathbb{R}^{1\times n}$ be the state-space and $\left\{\underline{x}(0), \ldots, \underline{x}(m)\right\} \subseteq \mathcal{M}$ a dataset. Consider the problem of approximating a time series provided by the dataset such that $\underline{x}(t+1) = \underline{f}(\underline{x}(t))$ by using a Koopman operator as will be described in the following. The $\mathbb{C}$-free set of observables $\mathcal{D} = \{g_1, \ldots, g_s\} \subseteq \mathcal{F}$ on which the Koopman operator acts is known as the *Dictionary* of observables. Moreover, the vector

$$\vec{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_s \end{bmatrix} : \mathcal{M} \to \mathbb{C}^s$$

is known as *the vector of observations*

Remark 11: Let n be the number of target variables, s the cardinal of the free set of observables, and m the number of samples or rows of the dataset. In practice, the order relation $n < s < m$ is usually satisfied.

We state below the main theoretical result we will need in what follows and in our experiments. This result was already noted in [42], and [45, §2.2, §2.3]; but the proof of the result is given here in detail.

Theorem 1: Let

$$x(t+1) = f(x(t))$$

be a one-dimensional dynamical system (i.e. $n = 1$). Let $\{g_1, \ldots, g_s\}$ be a dictionary of observables, and assume $g_1(x) = x$. Define

$$K = \begin{bmatrix} 1 \ 0 \ldots 0 \end{bmatrix} \cdot Y_1 \cdot Y_0^\dagger,$$

where

$$\begin{aligned} Y_0 &= \begin{bmatrix} \vec{g}(x(0)) \ldots \vec{g}(x(m-1)) \end{bmatrix} \in \mathbb{C}^{s\times m} \\ Y_1 &= \begin{bmatrix} \vec{g}(f(x(0))) \ldots \vec{g}(f(x(m-1))) \end{bmatrix} \in \mathbb{C}^{s\times m} \end{aligned}$$

and $Y_0^\dagger$ denotes the pseudo-inverse of the matrix Y_0. Then, K is the least squares/minimum norm solution to the Koopman operator $\mathcal{K}(f)$ restricted to the vector subspace $\langle g_1, \ldots, g_s\rangle \subset \mathcal{F}(\mathbb{R})$.

Proof: Consider the data matrices

$$Y_0 = \begin{bmatrix} \vec{g}(x(0)) \ldots \vec{g}(x(m-1)) \end{bmatrix} \in \mathbb{C}^{s\times m} \tag{2}$$

and

$$\begin{aligned} Y_1 &= \begin{bmatrix} \vec{g}(f(x(0))) \ldots \vec{g}(f(x(m-1))) \end{bmatrix} = \\ &= \begin{bmatrix} \vec{g}(x(1)) \ldots \vec{g}(x(m)) \end{bmatrix} \in \mathbb{C}^{s\times m} \end{aligned} \tag{3}$$

From the Koopman operator properties, it follows that

$$\begin{aligned} \mathcal{K}Y_0 &= \mathcal{K}\begin{bmatrix} \vec{g}(x(0)) \ldots \vec{g}(x(m-1)) \end{bmatrix} = \\ &= \begin{bmatrix} \vec{g}(f(x(0))) \ldots \vec{g}(f(x(m-1))) \end{bmatrix} = Y_1 \end{aligned}$$

Hence, any approximation matrix K to infinite-dimensional operator $\mathcal{K}$ that operates onto the s-dimensional subspace generated by the observables

$$K : \operatorname{span}_{\mathbb{C}}\{\mathcal{D}\} \longrightarrow \operatorname{span}_{\mathbb{C}}\{\mathcal{D}\}$$

must approximate the linear equation

$$Y_1 = K \cdot Y_0$$

On the other hand, by the properties of Moore-Penrose pseudo inverse, we obtain the minimum Frobenius norm solution to the optimization problem

$$Y_1 \cdot Y_0^\dagger = \operatorname{argmin}\{\|AX - Y\|_F\},$$

where $\|A\|_F = +\sqrt{AA^*}$ denotes the Frobenius norm.

Now we recall that the first observable $g_1(x) = x$ is in fact the identity function, hence above equality now reads

$$K \cdot Y_0 = K \begin{bmatrix} x(t) \\ g_2(x(t)) \\ \vdots \\ g_s(x(t)) \end{bmatrix} = \begin{bmatrix} x(t+1) \\ g_2(x(t+1)) \\ \vdots \\ g_s(x(t+1)) \end{bmatrix}$$

and, consequently, the approximated time series is obtained as

$$x(t+1) = \begin{bmatrix} 1 \ 0 \ldots 0 \end{bmatrix} \cdot K \cdot \begin{bmatrix} x(t) \\ g_2(x(t)) \\ \vdots \\ g_s(x(t)) \end{bmatrix}$$

Remark 12: The above method is, in fact, a particular case of the exact EDMD method. The reader can see in [42] a complete formulation of exact DMD algorithms both in sequential and non-sequential data [42, Algorithms 1,2]. Note that both algorithms are near identical in their original setting [42, Remark 2] and that the key issue is to manage data pairs $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ such that $y_k = f(x_k)$.

Note that in the sequential case, one also has $y_k = x_{k+1}$. This property is observed in our case below since our data is obtained as a time series. However, $y_k = x_{k+1}$ is not needed in the general setting where it is only demanded that each y happens to be the successor of the corresponding x.

C. DICTIONARIES OF OBSERVABLES

The choice of the vector $\vec{g}$ of observables (equivalently, the dictionary $\mathcal{D}$ of observables) is crucial for the method. In [44], it is proved that the accuracy and rate of convergence of DMD depend on $\mathcal{D}$, whose elements span the subspace of observables $\mathcal{F}_{\mathcal{D}} \subset \mathcal{F}$. Possible choices of elements of $\mathcal{D}$ include: polynomials, Fourier modes, radial basis functions, and spectral elements; but the optimal choice of basis of functions likely depends on both the underlying dynamical system and the sampling strategy used to obtain the data. Unfortunately, there is no method to select a good dictionary, and, in general, the choice depends on the dynamic that is going to be approximate.

Since our data-driven method considers no *a priori* knowledge of any property of the dynamic, we need to make a blind choice of the dictionary. Due to the lack of prior knowledge about the underlying dynamical system, any of these sets would be, in principle, a reasonable choice for $\mathcal{D}$.

The first experiment in section V involves some very basic dictionaries that have been chosen to obtain some results.

In the further experiment in section VI, a dictionary optimization is developed for each database studied by performing several trials and feeding back the results using adequate metrics on the predictions.

IV. VALIDATION

A. VALIDATION OF THE MODEL

We evaluate the models by Cross-Validation (CV) with a rolling scheme. We compute three directional measures and report the average overall folds. All folds have the same size, which is a fixed constant along with the validation method.

Several error measures are commonly considered for evaluating forecasting accuracy. Measures based on the sum of squares of errors are usually computed in the case of parametric-based models. Nevertheless, these measures are biased because significant errors are over penalized; hence both trend and direction of data are not considered [4]. In this work, accuracy is computed by using directional measures as follows.

Definition 5 (Directional Measures of Forecasted Time Series): Assume

$$(x_1, x_2, \ldots, x_m)$$

are the test-data and $\hat{x}_i$ is the forecasted value of x_i. Let True$[P]$ be the true value of the sentence P; that is, True$[P] = 1$ if P does hold and True$[P] = 0$ if it does not. Then, accuracies are defined below based on whether predictions and the true values of the time series "have the same direction". To be precise, we define:

- Directional Accuracy is given by

$$DA(i) = \begin{cases} 1 & \text{if } (\hat{x}_{i+1} - \hat{x}_i) \cdot (x_{i+1} - x_i) > 0 \\ 1 & \text{if } \hat{x}_{i+1} = x_{i+1} = x_i \\ -1 & \text{otherwise} \end{cases}$$

- Directional forecast Value

$$DV(i) = |x_{i+1} - x_i| \cdot DA(i)$$

- Mean Directional Accuracy

$$MDA = \text{mean of } DA(i)$$

- Mean Directional forecast Value

$$MDV = \text{mean of } DV(i)$$

MDA and MDV are obtained for each time-window. Then, we compute the average along all time-window data. Now we propose a new directional accuracy measure:

Definition 6 (Normalized Forecast Values):

- Normalized Directional forecast Value

$$NDV = \frac{\sum DV(i)}{\sum |x_{i+1} - x_i|}$$

- Mean of Normalized Directional forecast Value

$$MNDV = \text{mean of } NDV \text{over all data windows}$$

MDA and MNDV range between -1 (which means that predictions do not match the directions of real time series), and $+1$ (in the case of all predictions do match).

B. DATA-DRIVEN IMPROVEMENT OF DICTIONARIES

Former algorithms MDA and MDV are stated in terms of a static dictionary $\mathcal{D}$ which contains an identity function. Hence $\vec{g}(x) = [x, g_1(x), \ldots, g_s(x)]^t$ is a constant vector of functions. However, a search of optimal observables might be performed through of Artificial Intelligence methods.

Once a validation method has been established, MDA and MDV for a cross-validation rolling scheme become objective functions to maximize. Hence, it makes sense to introduce dynamic dictionaries. In our subsequent experiments, we start by validating our method over some static dictionaries. Later on, we introduce dynamic dictionaries. These are dictionaries that depend on some parameters and show that the score function is improved.

V. AN EXPERIMENT

We will base our experiment on a Canadian corporation [19] public database of Adware events. An Adware captures the user's browser or other parts of the system to overflow it with unwanted ads. There are different types of Adware, each of them presenting its particular behavior. Some assail with advertisements; others download unsolicited plug-ins or applications; others track the user's Internet activity and inform their owner to sell information. Some adware even acts as a "man-in-the-middle" attack and redirects all traffic through the user's system. Adware can collect personal information by tracking the visited websites or logging the pressed keys at its most extreme level. This aspect of Adware is very similar to spyware. Adware aims to generate income for its owner, who earns money each time the user clicks on one of the displayed advertisements. They can also sell their browsing data to third parties.

A. THE DATA

The CSV files were stacked and transformed into an aggregable table format so that each row, $r(i)$, is a cybersecurity report containing 85 alphanumeric attributes of the event. Our dataset consists of roughly 425.000 rows. Hence, table T has 425.000 rows and 85 columns and is an aggregable table in the sense of 1. Denote by $r(i)$ the i-th row of the dataset and $R_j(i)$ the j-th component of the i-th row. Here, the features that we will consider to form the aggregate time series are:

- The timestamp of i-th report, which is located at 7-th entry, $r_7(i)$.
- The threat label of i-th report, which is located at 85-th entry, $r_{85}(i)$.

B. THE TIME SERIES

We integrate two attributes along the table $\mathbf{T}$ and select a determined threat. We set up the time series of how many threats have been reported in some time-interval by using the time stamp. An interval of $d\tau = 10'$ has been selected. Hence, we obtain the time series

$$(\mathfrak{X}_{\text{Adware}}(t))_0^{264} = \int_{\mathbf{T}} d\tau [r_{85} = \text{Adware}]$$

of how many Adware events have been reported to the Canadian corporation every 10 minutes during 44 hours. This time series is our object of study in experiments.

Remark 13: Note that once we have forecasted the series, we do not have an estimation/prediction of how many adware events will occur in the (say) next 10 minutes intervals. We will estimate of how many adware events will be reported to the corporation in (say) the next 10 minutes intervals.

Remark 14: Note also that we have not considered most information in the reports. Of course, it is possible to consider more features to show correlations between them or even to give a more complex model than how-many-events-time series. We recall that dynamic mode decomposition below and, in a more general setting, the Koopman operator method allows to manage vectorial time series and even dynamical control systems containing external inputs.

Remark 15: Note that the data we are considering had been preprocessed in advance. However, when working with data from other sources, the preprocessing step is of vital importance. Concerning time series, this usually consists of filtering the signal with a low pass filter and replacing the time series with a rolling average calculated from it.

C. THE MODEL

A fully data-driven method is chosen: The EDMD algorithm for sequential data of a time series

```
Algorithm 1: Approximating $K$ as in Theorem 1
  Input: $\{x(t), x(t+1), \dots, x(t+m-1)\}$,  $\vec{g}$
        /* List of values in the time
        series & dictionary */
  Output: $K$         /* LSS approximation */
  Function Koopman ($\{x(t), \dots, x(t+m-1)\}, \vec{g}$):
      $Y_0 \leftarrow \left[\vec{g}(x(i)) \dots \vec{g}(x(i+m-1))\right]$;
      $Y_1 \leftarrow \left[\vec{g}(x(i+1)) \dots \vec{g}(x(i+m))\right]$;
      $K \leftarrow \left[1\ 0 \dots 0\right] \cdot Y_1 \cdot Y_0^{\dagger}$;
      return $K$;
  End Function
```

By using Algorithm 1, we calculate a predictor K from which we get the dynamic equation $\hat{x}(t+1) = K \cdot \vec{g}(x(t))$. This leads to the procedure, OneStepPred, we are proposing as technique: Obtain(s) 1 step prediction by means of Data-Driven Koopman method over s term of a time series. The idea of forecasting the one step trend of cybersecurity reports in the next temporal horizon.

In that sense, our approach consists in the following:

1) Whenever we receive a new sample $x(T)$ of the data we calculate K using algorithm 1 for input $\{x(T-m), \dots, x(T)\}$ and $\vec{g}$ some dictionary of observables,
2) then we calculate prediction $\hat{x}(T+1) = K \cdot \vec{g}(x(T))$,
3) we wait for next sample $x(T)$ and when received we go back to point 1

```
Procedure OneStepPred
  Input: $x(t), \vec{g}, s$
  while we keep receiving values $x(T)$ do
      $K \leftarrow Koopman(\{x(T-s), \dots, x(T)\}, \vec{g})$;
      $\hat{x}(T+1) \leftarrow K \cdot \vec{g}(x(T))$;
      wait for next $x(T)$;
  return $\hat{x}$;
```

D. STATIC DICTIONARIES OF OBSERVABLES

EDMD requires also a choice of a so-called *Dictionary of observables*, which are free (in the sense of Linear Algebra) families of functions. This choice is another hyper-parameter of the model also subject to further optimization. In our experiment we chose three static dictionaries:

$$\begin{aligned} \mathcal{D}_1 &= \{x, 1, \sin x, \cos x, \sin\ 2x, \cos 2x\}, \\ \mathcal{D}_2 &= \left\{x, 1, x^2, x^3, x^4\right\}, \\ \mathcal{D}_3 &= \{x, 1, \sin x, \cos x\}. \end{aligned} \tag{4}$$

E. VALIDATION RESULTS

Cross Validation (CVAL) and Out of Sample (OOS) are validation approaches in forecasting. OOS methods, like Rolling Cross Validation (RCV) are specially designed for time dependent data. In this paper, a RCV is performed by means of successive validations over data by using windows of size m. Each data window has one more point at the end and one less at the beginning ($N - m + 1$ windows). Each data window is divided into a training data ($p\%$) and a test data ($(100 - p)\%$ at the end).

Koopman matrix K is approximated with the training data at each step. Forecast values are computed over the corresponding test data sets. Validation was performed using *MATLAB*.

Above tables 1 and 2 show MDA, MDV and MNDV measures obtained by Rolling CV with $p = 60\%$ and two different window sizes: 24 points (4 hours), and 48 points (8 hours). These two time intervals are sufficiently significant for the 44 hours of collected data. Table 1 shows results for real Adware time series, while Table 2 shows results for standardized data. It is worth noticing that results show that our procedure fairly forecasts the directions of data. Best results are obtained for dictionary $\mathcal{D}_3$ with window size $= 48$ points.

Algorithm 2: Calculating Forecasted Values for the $(1-p)\%$ Testing Area of a Window

```
Input: $\{x(t), \ldots, x(t+m)\}$,  $\vec{g}$,  $p$ /* Window of
       the time series, dictionary &
       perc. */
Output: $\{\hat{x}(t+p*m+1), \ldots, \hat{x}(t+m+1)\}$
        /* forecasted values in the
        testing area */
Function PredictTest ($\{x(t), \ldots, x(t+m)\}, \vec{g}, p$):
    $K \leftarrow Koopman(\{x(t), \ldots, x(t+m*p)\}, \vec{g})$;
    $\hat{x} \leftarrow \{\ \ \}$
    for $i \leftarrow m*p$ to $m$ do
        $\hat{x}$.append($K \cdot \vec{g}(x(i))$)
    end
    return $\hat{x}$;
End Function
```

Algorithm 3: Rolling Cross Validation

```
Input: $x(t)$,  $\vec{g}$,  $m$
Output: MDA
$x \leftarrow \{\ \ \}$;
$\hat{x} \leftarrow \{\ \ \}$;
for each disjoint window $w \leftarrow \{x(t), \ldots, x(t+m)\}$
 do
    $\hat{x}$.concatenate($PredictTest(w, \vec{g}, p)$);
    $x$.concatenate($\{x(t+m*p), \ldots, x(t+m)\}$);
return MDA($x, \hat{x}$);;
```

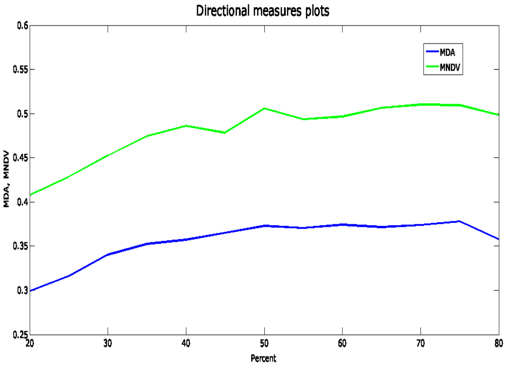


FIGURE 3. MDA and MNDV for rolling CV as function of *p*.

Figure 3 shows the MDA and MNDV for our rolling CV method taking into account 48 data and training percentages ranging from $p = 20\%$ to $p = 80\%$. We put this results in order to take into account how do those accuracy measures vary as a function of training percentage p. For this concrete experiment one can see that best option to train the method is a percentage of around $p = 70\%$.

Consider the three dictionaries in (4). Forecasted values within the validation are depicted in figures 4, 5, 6, 7, 8 and 9. For each graph, there are four elements that should strike our attention:

TABLE 1. Directional measurements obtained by Rolling CV procedure of adware time series data.

	Window Size 24		
	MDA	**MDV**	**MNDV**
Dictionary 1	0.3251	187.33	0.4746
Dictionary 2	0.2975	390.25	0.4121
Dictionary 3	0.3554	465.20	0.4977

TABLE 2. Directional measurements obtained by Rolling CV procedure of adware time series data.

	Window Size 48		
	MDA	**MDV**	**MNDV**
Dictionary 1	0.3721	193.68	0.5050
Dictionary 2	0.3308	452.44	0.4708
Dictionary 3	0.3741	475.62	0.4966

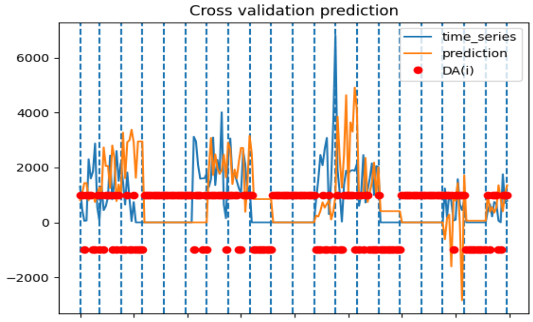


FIGURE 4. Execution of the algorithm with dictionary D1 and window size 24.

1) The blue line represents the real time series $x(t)$ of adware events for each time t.
2) The orange line represents the prediction $\hat{x}(t)$ done inside the validation process.
3) The red dots represent $DA(t)$ for each time t they are placed on top of $y = 0$ if the prediction had the same trend as the time series (that is when $DA(t) = 1$) and placed under $y = 0$ if not (that is when $DA(t) = -1$).
4) The vertical dashed lines represent a separation between predictions. As we are representing the cross validation method there exists a gap between each testing block. Precisely, this is the gap formed by the 70% of the window used for training.

VI. A SECOND EXPERIMENT: TOWARDS BETTER DICTIONARIES

In this section, we discuss how to correctly choose our dictionary $\mathcal{D}$; results could be improved by predicting the time series with appropriately chosen dictionaries. Presumably, the best dictionary to use depends on the data being handled. For example, it seems natural to use harmonics on rhythmical data or polynomials on data that is sufficiently stable (smoothly changing). However, big chunks of data do often appear in real-world applications. Thus, trends are difficult to infer just by looking at the time series. Choosing a dictionary

TABLE 3. Directional measurements obtained by Rolling CV procedure of standardized adware time series data.

	Window Size 24		
	MDA	**MDV**	**MNDV**
Dictionary 1	0.2929	0.3247	0.3939
Dictionary 2	0.2443	0.2849	0.3491
Dictionary 3	0.3221	0.3592	0.4597

TABLE 4. Directional measurements obtained by Rolling CV procedure of standardized adware time series data.

	Window Size 48		
	MDA	**MDV**	**MNDV**
Dictionary 1	0.3400	0.3927	0.4677
Dictionary 2	0.3201	0.3781	0.4600
Dictionary 3	0.3461	0.3952	0.4801

that maximizes the prediction score can be treated as an optimization problem. As we are facing a general problem, metaheuristics seem a good approach. We will be using two very known techniques to maximize an objective function O which we assume is differentiable. Namely:

1) Gradient ascent
2) Simulated annealing

A. OBJECTIVE FUNCTION

The objective function $O : \mathbb{R} \times \ldots \times \mathbb{R} \rightarrow \mathbb{R}$ is going to be defined in terms of the MDA score defined in IV-A. For every parametrized dictionary $\mathcal{D}(\theta_1, \ldots, \theta_l)$ we will be calculating the average MDA score for a cross validation scheme with window size 24 where the first 14 elements are taken as training data and the last 10 as test. That is MDA will be calculated for the training data of each window and doing the average over all the windows. The reason why we used a window of size 24 is that previous experiments showed little difference in the results as the window size varied. The main purpose of this experiment is establishing a link between good choices of dictionaries and better predictions.

B. EXPERIMENTING WITH PARAMETRIZED DICTIONARIES

When parametrizing the family of free dictionaries, it is important to notice how changes in the dictionary do not affect the Koopman operator if the spanned subspace is not extended. Hence, when introducing variability to a dictionary, it has to be done in a non-linear fashion.

We choose two families of free dictionaries. It is presumed that the goodness of their fit will depend on the nature of the data.

1) (PRESUMED) PERIODIC DATA

Let $\mathcal{D}_{harm} : \mathbb{N} \times \mathbb{R} \rightarrow \mathscr{P}(\mathcal{F})$ be a function that maps each pair (n, T) to a dictionary of harmonics:

$$\begin{aligned}(n, T) \mapsto \{x, 1\\ &\cos\left(\frac{2\pi x}{T}\right), \ldots, \cos\left(\frac{2\pi nx}{T}\right),\\ &\sin\left(\frac{2\pi x}{T}\right), \ldots, \sin\left(\frac{2\pi nx}{T}\right)\}\end{aligned}$$

In this experiment, we will be exploring the parameter space $\mathbb{N} \times \mathbb{R}$ to find a local maximum of the MDA function using the gradient ascent technique.

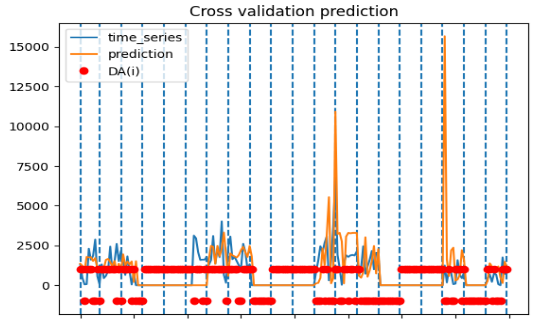


FIGURE 5. Execution of the algorithm with dictionary D2 and window size 24.

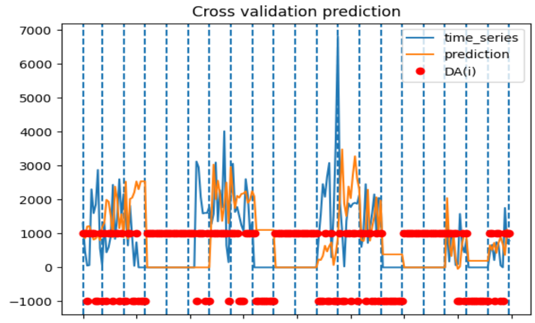


FIGURE 6. Execution of the algorithm with dictionary D3 and window size 24.

2) POLYNOMIAL DICTIONARY

Let $\mathcal{D}_{poly} : \{0, 1\}^7 \rightarrow \mathscr{P}(\mathcal{F})$ be a function that maps each septuple of zeros and ones to a polynomial dictionary:

$$(b_i)_{i=0,\ldots,6} \mapsto \left\{x, \delta_{b_0 1} 1, \delta_{b_1 1} x, \ldots, \delta_{b_6 1} x^6\right\}$$

We will be using simulated annealing to find the best septuple that maximizes the MDA function. The neighbour function in the simulated annealing technique is calculated with a random walk inside the Hasse diagram of the power set of the 7-element set $\left\{1, x, x^2, \ldots, x^6\right\}$ ordered by inclusion.

Just the first 7 elements of the infinite basis of the polynomial family have been chosen; the number of elements and their degree has an influence in the performance of the program thus should be tweaked according to that fact.

C. STEEPEST GRADIENT

We start by optimizing the objective function MDA $\circ$ D_{harm} for the dictionary of harmonics using the steepest gradient

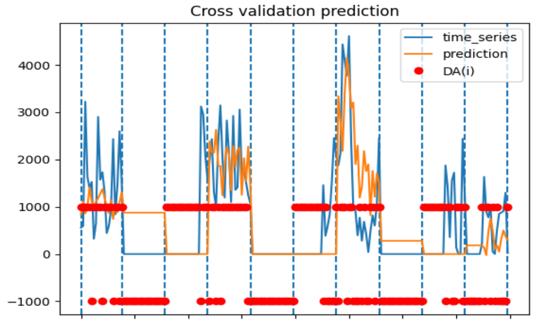


FIGURE 7. Execution of the algorithm with dictionary D1 and window size 48.

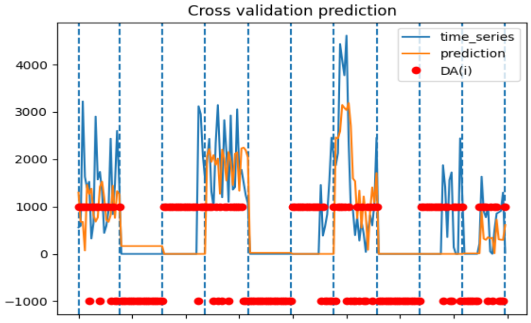


FIGURE 8. Execution of the algorithm with dictionary D2 and window size 48.

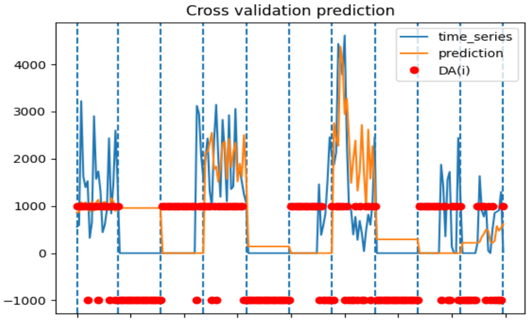


FIGURE 9. Execution of the algorithm with dictionary D3 and window size 48.

heuristic. We see how we start with parameters $N = 8$ and $T = 100$ which give a score $MDA = 0.279$ and after a gradient ascent technique we reach a local maximum at $N = 27$ and $T = 9$ where the objective function is $MDA = 0.29$. In 10 there is a graphical representation of the evolution of MDA through the steepest gradient execution and in 11 we see how this prediction adjusts to the data.

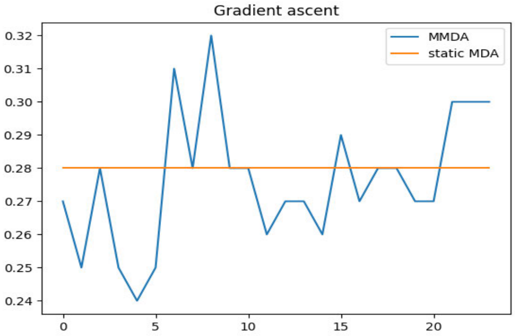


FIGURE 10. Gradient ascent for harmonics.

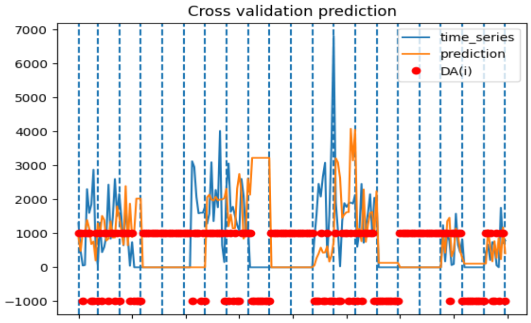


FIGURE 11. Predicting with tuned dictionary of harmonics.

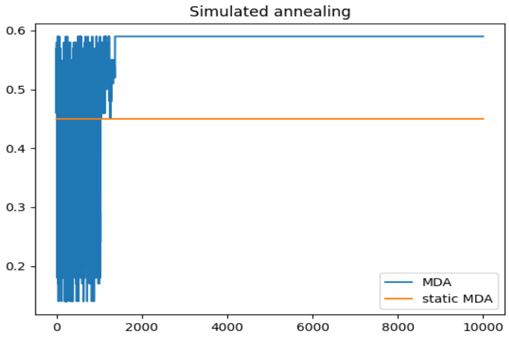


FIGURE 12. Simulated annealing technique on the dictionary of polynomials.

D. SIMULATED ANNEALING

The second optimization problem is discrete, it can be rephrased as choosing the best [2] combination of numbers within 5 numbers. As it is not continuous it is naturally approached with the simulated annealing technique. We want to maximize an original score given by dictionary $\mathcal{D} = \{x, x^2, x^4, x^5\}$ of $MDA = 0.45$. We get a score of $MDA = 0.59$ with the dictionary $\mathcal{D} = \{x, x^5\}$ after executing the simulated annealing procedure over our space of parameters. In 12 we see the evolution of the objective function on every update of the simulated annealing procedure and in 13 we see how the prediction adjusts the data.

E. FURTHER COMMENTS

1) It is worth noticing that correctly tuning the dictionary used to forecast values substantially improves the score obtained by an arbitrary dictionary. Hence, tweaking the dictionary is proposed as a key step when using our predictor. In addition, this should serve as encouragement to search for new techniques in order to find

[2] where best is defined in terms of maximizing MDA

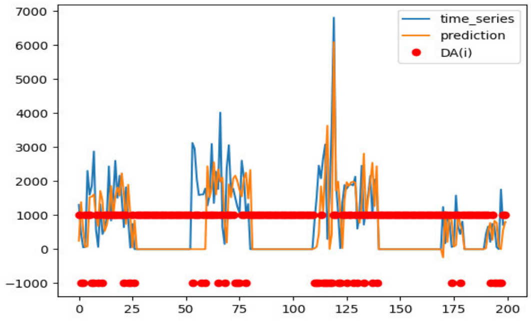


FIGURE 13. Predicting with tuned dictionary of polynomials.

better dictionaries. In that sense, we pursue a streaming procedure sensible to newly arrived data. The choice of dictionary deeply affects the quality of the prediction and, therefore, feedback from new samples should be employed to tune the dictionary. These samples being managed in a pay-as-you-go fashion.

2) The results provided in this article have been obtained considering the temporary data of Adware-type cyberattacks. In the case of having two different types of attacks, one could ignore this aspect and carry out the same procedure that we have exposed. However, when the nature of the attack types differs too much from one to another, the aggregate time series can also follow very different patterns. This suggests that ignoring this aspect may lead to less accurate than desired results. An alternative way of predicting the risk trend in a situation like the one described above would be to treat the two cases separately and take the sum of the two predictions as the final prediction. The multi-attack risk prediction problem should be addressed in the future.

3) The experimentation taken in this article deals with a particular case of the general formulation given in Section II-B, namely, the case in which r_i is a discrete (in fact, binary) random variable. There are many other situations in which our procedure can be applied. For instance, one can monitor social networks and predict the trend of risk of fake news appearing [13]. However, our procedure can also be used in the continuous case through a previous discretization process. Discretization processes use to be characterized by giving thresholds. In our situation, this can easily be formulated as follows: define the first order sentence $P(r_i) = \pi_k(r_i) > a$ and apply Remark 5. This might be particularly useful in intelligent environmental control systems [20]. Here, we can define a *risky situation* as one in which the temperature, humidity, or any other variable of interest, takes a value greater (or lower) than a given threshold. Our procedure can be understood, in this situation, as a way to predict the trend of risk of having an extreme environment.

VII. DISCUSSION

The previous experiments show how this prediction method can benefit substantially from a correct choice of parameters. On the one hand, in the first set of experiments, we noticed how the best results were obtained with the third dictionary $\mathcal{D}_3$ for window size = 48 points and a training percentage of around 70%. This is a choice that is highly dependent on the data we are treating, so we suggest a fine-tuning of these parameters when considering applications of the method described in this article. One way of doing this is using any of the grid-search implementations available. On the other hand, it is also important to notice how the choice of the dictionary involved in the prediction affects the prediction accuracy. The application of meta-heuristics to this particular problem is of great benefit. Precisely, the steepest-gradient technique has been applied to accurately choose a dictionary of harmonics. This technique increased the score from $MDA = 0.279$ to $MDA = 0.29$. Since the data used was not periodical, this increase is valuable. Moreover, applying the simulated annealing technique to a dictionary of polynomials had the greatest impact on precision, increasing the score from $MDA = 0.45$ to $MDA = 0.59$. This set the maximum value within all experiments.

VIII. CONCLUSION

A concrete procedure to obtain predictions about cybersecurity measures from cybersecurity reports is given. The procedure relies on a weak statement of the problem by assuming that a time-stamped database of numerical measures of some cybersecurity threat is given. In the stronger formulation of the problem, the data shall be collected and ulteriorly aggregated into a table.

Next, a time series is integrated from the table; this time series is forecasted using data-driven methods. Finally, predictions are validated employing Cross-Validation of directional measures.

First experiments of an EDMD model to forecast the number of cybersecurity reports in a concrete environment (which are datasets of reports of Android malware gathered by some providers) show acceptable forecasting for data direction. Hence, it is worth improving this kind of data-driven methods in the future.

Moreover, metaheuristics have proven themselves useful in the parameter tuning technique to select suitable dictionaries for building up the model. We leave for future work the development of a genetic algorithm to tune the dictionary of observables.

In this work, the dependency relation between the size of the prediction time window and the precision of the model's predictions has not been studied. Knowing how large the size of the prediction time window can be before the model loses too much precision is of great relevance. It is worth mentioning that the automatization of the procedure presented in this work would allow a statistical study to be carried out easily.

As a final remark, we would like to note that the proposed forecasting method would apply to any time series. Hence, if one is provided with any time-stamped dataset, the aggregation procedure produces a time series, no matter the original data properties. Thus any dynamic process can be forecasted following the steps: *1) integrate the time series from your time-stamped data, 2) forecast the time series using some Koopman-based method, 3) optimize dictionaries, hyper-parameters*. Hence, some interesting and data-intensive research fields like cybersecurity (for instance, web credibility [13], risky behaviors in private cyber-activity [11], models of cyberattack detection [2], and dynamic monitoring systems) or infectious models (like [1], [3], and [12]) are fields of application of the prediction tools presented in this paper.

REFERENCES

[1] M. O. Adeniyi, M. I. Ekum, C. Iluno, and S. I. Oke, "Dynamic model of COVID-19 disease with exploratory data analysis," *Sci. Afr.*, vol. 9, Sep. 2020, Art. no. e00477.

[2] H. I. Ahmed, A. Nasr, A. Abdel-Mageid, and H. K. Aslan, "DADEM: Distributed attack detection model based on big data analytics for the enhancement of the security of Internet of Things (IoT)," *Int. J. Ambient Comput. Intell.*, vol. 12, no. 1, pp. 114–139, 2021.

[3] M. Ala'raj, M. Majdalawieh, and N. Nizamuddin, "Modeling and forecasting of COVID-19 using a hybrid dynamic model based on SEIRD with ARIMA corrections," *Infectious Disease Model.*, vol. 6, pp. 98–111, Jan. 2021.

[4] C. Bergmeir, M. Constantini, and J. M. Benítez, "On the usefulness of cross-validation for directional forecast evaluation," *Comput. Statist. Data Anal.*, vol. 76, pp. 132–143, 2014.

[5] A. Bjerhammar, "Application of calculus of matrices to method of least squares; with special references to geodetic calculations," Trans. Roy. Inst. Technol., Stockholm Sweden, Tech. Rep. 49, Mar. 1951, pp. 1–86.

[6] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. Nathan Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," 2015, *arXiv:1510.03007*. [Online]. Available: http://arxiv.org/abs/1510.03007

[7] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering*. Cambridge, U.K.: Cambridge Univ. Press, 2019.

[8] M. V. Carriegos and R. A. Fernández-Díaz, "Towards forecasting time-series of cyber-security data aggregates," in *Proc. 13th Int. Conf. Comput. Intell. Secur. Inf. Syst. (CISIS)*, vol. 1267, A. Al. Herrero, Eds. Springer, 2021.

[9] N. DeCastro-García, Á. L. Muñoz Castañeda, M. Fernández Rodríguez, and M. V. Carriegos, "On detecting and removing superficial redundancy in vector databases," *Math. Problems Eng.*, vol. 2018, May 2018, Art. no. 3702808.

[10] N. DeCastro-García, Á. L. Muñoz Castañeda, D. Escudero García, and M. V. Carriegos, "Effect of the sampling of a dataset in the hyperparameter optimization phase over the efficiency of a machine learning algorithm," *Complexity*, vol. 2019, Feb. 2019, Art. no. 6278908.

[11] W. Chmielarz and O. Szumski, "Cyber security patterns students behavior and their participation in loyalty programs," *Int. J. Ambient Comput. Intell.*, vol. 9, no. 2, pp. 16–31, Apr. 2018.

[12] A. M. Elaiw, S. F. Alshehaiween, and A. D. Hobiny, "Global properties of virus dynamics with B-cell impairment," *Open Math.*, vol. 17, no. 1, pp. 1435–1449, Dec. 2019.

[13] M. Faraon, "Fake news and aggregated credibility: Conceptualizing a co-creative medium for evaluation of sources online," *Int. J. Ambient Comput. Intell.*, vol. 11, no. 4, pp. 93–117, 2020.

[14] M. Fliess, C. Join, M. Bekcheva, A. Moradi, and H. Mounier, "Easily implementable time series forecasting techniques for resource provisioning in cloud computing," 2019, *arXiv:1903.02352*. [Online]. Available: http://arxiv.org/abs/1903.02352

[15] K. Fujii and Y. Kawahara, "Supervised dynamic mode decomposition via multitask learning," *Pattern Recognit. Lett.*, vol. 122, pp. 7–13, May 2019.

[16] T. Grance, K. Kent, and B. Kim, "Computer security incident handling guide," NIST, Gaithersburg, MD, USA, Special Publication 800-61, 2004, p. 11.

[17] S. M. Hirsh, K. Decker Harris, J. Nathan Kutz, and B. W. Brunton, "Centering data improves the dynamic mode decomposition," 2019, *arXiv:1906.05973*. [Online]. Available: http://arxiv.org/abs/1906.05973

[18] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, Jul. 2018.

[19] A. H. Lashkari, A. Fitriah, A. Kadir, L. Taheri, and A. Ghorbani, "Toward developing a systematic approach to generate benchmark Android malware datasets and classification," in *Proc. 52nd IEEE Int. Carnahan Conf. Secur. Technol. (ICCST)*, Montreal, QC, Canada, 2018, pp. 1–7.

[20] Q. Liu, "Intelligent environmental monitoring system based on multi-sensor data technology," *Int. J. Ambient Comput. Intell.*, vol. 11, no. 4, pp. 57–71, Oct. 2020.

[21] C. Lobry and T. Sari, "Non-standard analysis and representation of reality," *Int. J. Control*, vol. 81, no. 3, pp. 519–536, 2008.

[22] J. M. Lopez, J. Vega, S. Dormido-Canto, A. Murari, J. M. Ramirez, M. Ruiz, G. D. Arcas, and C. Jet-Efda, "Integration and validation of a disruption predictor simulator in JET," *Fusion Sci. Technol.*, vol. 63, no. 1, pp. 26–33, Jan. 2013.

[23] F. J. Montáns, F. Chinesta, R. Gómez-Bombarelli, and J. N. Kutz, "Complex algorithms for data-driven model learning in science and engineering," *Complexity*, vol. 2019, Jun. 2019, Art. no. 5040637.

[24] J. R. Moya, N. DeCastro-García, R. A. Fernández-Díaz, and J. Lorenzana Tamargo, "Expert knowledge and data analysis for detecting advanced persistent threats," *Open Math.*, vol. 15, no. 1, pp. 1108–1122, 2017.

[25] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proc. Nat. Acad. Sci. USA*, vol. 17, no. 5, pp. 315–318, 1931.

[26] B. Koopman and J. vonNeumann, "Dynamical systems of continuous spectra," *Proc. Nat. Acad. Sci. USA*, vol. 18, no. 3, pp. 255–263, 1932.

[27] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, "Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator," *Chaos: Interdiscipl. J. Nonlinear Sci.*, vol. 27, no. 10, Oct. 2017, Art. no. 103111.

[28] C. Mackey and C. Kribs, "Can scavengers save zebras from anthrax? A modeling study," *Infectious Disease Model.*, vol. 6, pp. 56–74, Jan. 2021.

[29] I. Mezić and A. Banaszuk, "Comparison of systems with complex behavior," *Phys. D: Nonlinear Phenomena*, vol. 197, nos. 1–2, pp. 101–133, Oct. 2004.

[30] I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions," *Nonlinear Dyn.*, vol. 41, pp. 309–325, Aug. 2005.

[31] E. H. Moore, "On the reciprocal of the general algebraic matrix," *Bull. Amer. Math. Soc.*, vol. 26, no. 9, pp. 394–395, 1920.

[32] R. Penrose, "A generalized inverse for matrices," *Proc. Cambridge Philos. Soc.*, vol. 51, no. 3, pp. 406–413, 1955.

[33] N. R. Pokhrel, H. Rodrigo, and C. P. Tsokos, "Cybersecurity: Time series predictive modeling of vulnerabilities of desktop operating system using linear and non-linear approach," *J. Inf. Secur.*, vol. 8, no. 4, pp. 362–382, 2017.

[34] G. A. Rattá, J. Vega, A. Murari, and G. Vagliasindi, "Inspection of disruptive behaviours at JET using generative topographic mapping," in *From Physics to Control Through an Emergent View*. Singapore, 2010, 315–320.

[35] H. R. Santos. *IA y Analítica Predictiva: Del Dato a la Inteligencia de Ciberseguridad*. [Online]. Available: https://www.incibe-cert.es/blog/

[36] M. Sainz, I. Garitano, M. Iturbe, and M. Zurutuza, "Deep packet inspection for intelligent intrusion detection in software-defined industrial networks: A proof of concept," *Logic J. IGPL*, vol. 28, no. 4, pp. 461–472, 2020.

[37] A. Salova, J. Emenheiser, A. Rupe, J. P. Crutchfield, and R. M. D'Souza, "Koopman operator and its approximations for systems with symmetries," 2019, *arXiv:1904.11472*. [Online]. Available: http://arxiv.org/abs/1904.11472

[38] P. J. Schmid and J. Sesterhenn, "Dynamic mode decomposition of numerical and experimental data," in *Proc. Amer. Phys. Soc. (61st. Annu. Meeting APS Division Fluid Dyn.)*, 2008, pp. 5–28.

[39] R. Su and C. Zhang, "Stability and hopf bifurcation periodic orbits in delay coupled lotka-volterra ring system," *Open Math.*, vol. 17, no. 1, pp. 962–978, Aug. 2019.

[40] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, and Y. Xiang, "Data-driven cybersecurity incident prediction: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1744–1772, 2nd Quart., 2019.

[41] N. Takeishi, Y. Kawahara, and T. Yairi, "Learning Koopman invariant subspaces for dynamic mode decomposition," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 1130–1140.

[42] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition: Theory and applications," *J. Comput. Dyn.*, vol. 1, no. 2, pp. 391–421, 2014.

[43] R. Vega Vega, H. Quintián, J. L. Calvo-Rolle, Á. Herrero, and E. Corchado, "Gaining deep knowledge of Android malware families through dimensionality reduction techniques," *Log. J. IGPL*, vol. 27, no. 2, pp. 160–176, Mar. 2019.

[44] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition," *J. Nonlinear Sci.*, vol. 25, no. 6, pp. 1307–1346, 2015.

[45] M. O. Williams, M. S. Hemati, S. T. M. Dawson, I. G. Kevrekidis, and C. W. Rowley, "Extending data-driven Koopman analysis to actuated systems," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 704–709, 2016.

[46] P. You, J. Pang, and E. Yeung, "Deep Koopman controller synthesis for cyber-resilient market-based frequency regulation," *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 720–725, 2018.

[47] J. Zeng and B. Li, "Research on cooperation strategy between government and green supply chain based on differential game," *Open Math.*, vol. 17, no. 1, pp. 828–855, Aug. 2019.

[48] A. H. Lashkari, A. Andi Fitriah Kadir, L. Taheri, and A. Ali Ghorbani, "Toward developing a systematic approach to generate benchmark Android malware datasets and classification," in *Proc. 52nd Int. Carnahan Conf. Secur. Technol. (ICCST)*, Montreal, QC, Canada, Oct. 2018, pp. 1–7.

[49] (2021). *CISA Incident Reporting System*. [Online]. Available: https://www.us-cert.gov/forms/report

[50] *Indian Computer Emergency Response Team*. Accessed: Apr. 24, 2021. [Online]. Available: https://www.cert-in.org.in/PDF/certinirform.pdf

[51] *Cyber Security Incident Report. Technical Rationale and Justification for Reliability*, Standard CIP-008-6, Jan. 2019. [Online]. Available: https://www.nerc.com/

[52] (2013). *European Network and Information Security Agency (ENISA), Technical Guideline on Incident Reporting*. [Online]. Available: https://resilience.enisa.europa.eu/

MIGUEL V. CARRIEGOS received the B.Sc. and Ph.D. degrees in mathematics from the Universidad de Valladolid, in 1994 and 1999, respectively.

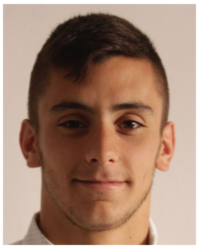
From 1996 to 1997, he was a Research Fellow with the Universidad de León, where he has been an Associate Professor, since 2003. From 2008 to 2016, he was as the Head of the Department of Mathematics and has been the Director of the Research Institute of Cybersecurity, since 2014. His research interests include linear and commutative algebra, systems theory, and cybersecurity.

ÁNGEL L. MUÑOZ CASTÑEDA received the B.Sc. degree from the Universidad de Salamanca, in 2011, and the Ph.D. degree in mathematics from Freie Universität Berlin, in 2017.

From 2017 to 2019, he was a Research Associate with the Universidad de León, where he has been an Assistant Professor, since 2019. His research interests include commutative algebra, algebraic geometry, mathematical methods in machine learning, and applications in cybersecurity.

M. T. TROBAJO received the B.S. and Ph.D. degrees in mathematics from the Universidad de Salamanca, in 1994 and 2002, respectively.

She was an Assistant Professor with the Universidad de Salamanca, from 1994 to 1996, and with Universidad de León, from 1996 to 2003. Since 2003, she has been an Associate Professor with the Universidad de León. She serves as a District Delegate for Real Sociedad Matemática Española as well as for Mathematics Olympiad. She is also the Head of the NARVIC Research Group. Her research interests include mathematical systems theory, applied statistics, and data analysis.



DIEGO ASTERIO DE ZABALLA received the B.Sc. degree in mathematics and the B.Eng. degree in computer science from the Universidad de Granada, in 2020.

He has been a Research Assistant with the Instituto de Ciencias Aplicadas a Ciberseguridad, Universidad de León, since 2020.