Hidden Markov Model Enabled Prediction and Visualization of Cyber Agility in IoT era

Eric Muhati and Danda B. Rawat *Senior Member, IEEE*

***Abstract*—Cyber-threats are continually evolving and growing in numbers and extreme complexities with the increasing connectivity of the Internet of Things (IoT). Existing cyber-defense tools seem not to deter the number of successful cyber-attacks reported worldwide. If defense tools are not seldom, why does the *cyber-chase* trend favor bad actors? Although cyber-defense tools monitor and try to diffuse intrusion attempts, research shows the required agility speed against evolving threats is way too slow. One of the reasons is that many intrusion detection tools focus on anomaly alerts' accuracy, assuming that pre-observed attacks and subsequent security patches are adequate. Well, that is not the case. In fact, there is a need for techniques that go beyond intrusion accuracy against specific vulnerabilities to the prediction of cyber-defense performance for improved proactivity. This paper proposes a combination of cyber-attack projection and cyber-defense agility estimation to dynamically but reliably augur intrusion detection performance. Since cyber-security is buffeted with many unknown parameters and rapidly changing trends, we apply a machine learning (ML) based hidden markov model (HMM) to predict intrusion detection agility. HMM is best known for robust prediction of temporal relationships mid noise and training brevity corroborating our high prediction accuracy on three major open-source network intrusion detection systems, namely Zeek, OSSEC, and Suricata. Specifically, we present a novel approach for combined projection, prediction, and cyber-visualization to enable precise agility analysis of cyber defense. We also evaluate the performance of the developed approach using numerical results.**

***Index Terms*—attack projection, agility prediction, IoT predictive analytics, cyber visualization**

I. Introduction

THE Internet-of-Things (IoT), big data, artificial intelligence, cloud computing, mobile devices, and social media have caused an unprecedented innovation explosion. This increased uptake in hyperconnected systems has unfortunately also attracted nefarious actors responsible for many adaptive and persistent cyber-threats. Although the connection between cyber-attacks and cyber-defense evolution differs, the complexity and severity levels of cyber-attacks have higher augmentation. New technology, protocol changes over time, and discovery of vulnerabilities spur malicious actors while cyber-defenders often respond *reactively*. These asymmetric actions keep attackers one step ahead, labeled as the *cyber-chase*. The rapid and dynamic development of persistent threats poses a formidable challenge as attackers keep responding with more dangerous attack methods [1]. Consequently, future technology stands at a perilous change if the *cyber-chase* does not include inexhaustible adaptation to the ever-evolving cyber-attacks.

The authors are with the Department of Electrical and Computer Engineering, Howard University, Washington, D.C, 20059.
E-mail: {eric.muhati, danda.rawat}@howard.edu
Manuscript received Nov XX, 2020; revised Month XX, 2020.

Although it is practically impossible to develop solutions immune to all future threats, research highlights a measure of cyber-defense deftness would significantly improve cyber-defense evolution over time [2]–[5]. And while such a measure would not be panacea, potential cyber-defense progress under enhanced attacks can be accurately forecasted, rather than single-use of stale *static* metrics (i.e., *precision* and *recall*) [6]. Apart from merely trying to out-guess attackers, *static* metrics contribute to tractable analysis as shown by [7]. The resulting underestimated true probability of future threats results in a feigned sense of security. The prevalent *go-to-option* for network defense described in survey [8] is network intrusion detection systems (NIDS), whose performances are commonly validated through some attack data.

However, in real-life scenario, cyber-attacks exponentially evolve through big nefarious ecosystems, excoriating current over-reliance on *static* NIDS metrics and sporadic *reactive* security patches. Would it be possible to understand NIDS agility to future threats based on observed reactionary time to patch new vulnerabilities and impeccably predict NIDS response faced with evolved cyber-attacks? Whereas it is possible to i) project attack steps [9], ii) understand an evolving adversarial ultimate goal [10], and iii) predict when an intrusion will happen i.e., *intrusion prediction* [2], such combined research approaches are seldom.

Currently, most NIDS effort is importantly but inadequately placed on improving risky activities' detection speed. With the recent introduction and research uptake in systematic quantitative metrics to measure cyber-defense evolution proposed in [11], focus on NIDS performance deftness *tomorrow* can be explored. Therefore, in this paper, we propose a combination of attack projection from a given attack dataset, attack intention recognition, and attack prediction to create a model predicting future NIDS unknown performance under evolved attacks. Since we can never be precisely sure of the attacker's behavior in real-life scenarios, our machine learning algorithm requires excellent ability to approximate unseen unobservable states and transitions, with minimal dependency on possessing complete information. Markov models are best in such situations by allowing accurate prediction even if some attack steps cannot be completely inferred [12].

HMM variants adds to this advantage by offering our approach robust prediction of temporal relationships mid noise and training brevity, to corroborate our high prediction accuracy on three major open-source network intrusion detection systems, namely Zeek [13], OSSEC [14] and Suricata [15]. We conclude with a novel visualization model for enhanced output comparison of the predicted cyber-agility since cyber visualization offers a better depiction of security events. Fig 1

shows an overview summary of our proposed model.

To understand the major concepts of our paper, the remaining portion is organized as follows. We compare related works concerned with cyber-agility prediction through machine learning-based threat prediction and NIDS evolution in Section II, then give background details enabling our novel proposal in Section III. We present our implementation in Section IV followed by the performance evaluation of our numerical results in Section V. Section VI gives an overview of the clear model output through an additional visual display module. Finally, we conclude in Section VII.

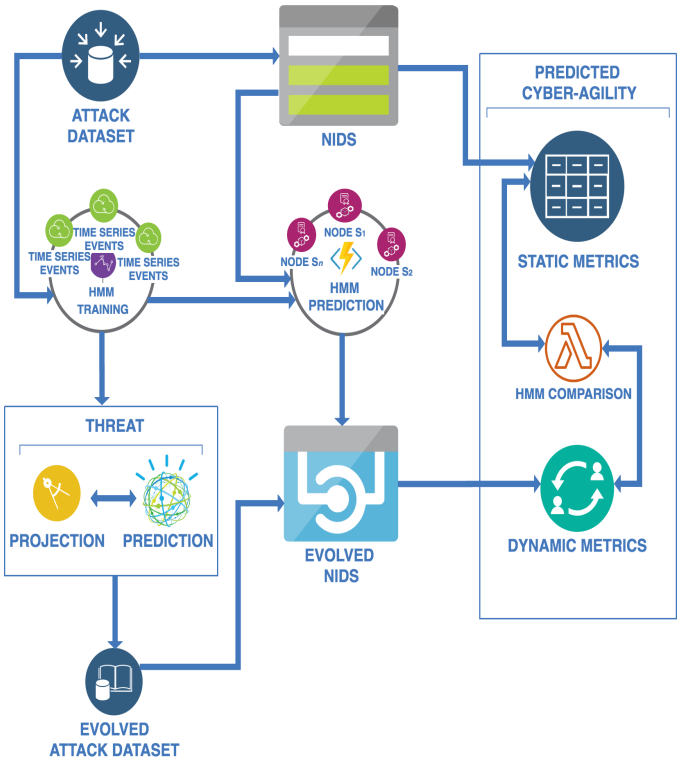


Figure 1: Overview of our proposed model.

II. Related Work

The composite framework to guide agility decisions explained in [6] depicts cyber-agility as the dynamic balance of speed and strength capabilities for guaranteed recovery of critical digital services under-exploited vulnerabilities. Many strategies that attempt to build cyber-hardened tools, as shown in survey [4], capture the dynamics of adapting vulnerabilities analysis and defense power estimation to minimize attack effects. Our paper's concern is proposed markov model methods that combine prediction of both threat severity and defense power to investigate the dynamic evolution of cyber-security.

ML prediction and classification of attacks focusing on intrusion detection at current time t, is not new. However, to the best of our knowledge, with a minor exception of [16], we are the first to estimate cyber-defense at future time t'' based on previous time t', then calculate cyber-agility as a measure of intrusion detection performance. The predictive defense algorithm proposed by Colbaugh and Glass [16] is close to our approach in that the authors predict defense algorithms against current and future attacks by combining *game theory* and ML. The model predicts cyber-attacks and extrapolates cyber-defense evolution by modeling attempts to transform data in the future as actionable attacks. However, Colbaugh and Glass make a critical assumption of *inevitable errors* to estimate optimal defense performance from tweaked defense parameters instead of astutely incorporating real statistics from actual NIDS output. Furthermore, working on a Spam/non-Spam dataset, [16] proposed tweaking of defense parameters assumes predictable attackers' resources, which is not the case in real-world scenarios. On the contrary, our proposed model adopts real NIDS metrics for minimal supervised training to assure error elimination.

Abraham and Nair [17] propose a predictive security analytics framework based on Markov models for exploitability analysis. Through network security analysis, [17] objective is achieved using intrinsic characteristics of vulnerabilities provided by common vulnerability scoring system (CVSS). Finally, they calculated a temporal exploitability score using expected path length and probabilistic path metrics to propose improved decision-making and risk reduction. However, these complete impact path values of an attack path certainty lack factors in causal performance preconditions and post-conditions. The causal knowledge is included in [18] where strategies are deployed to correlate probabilities between two hyper-alerts of known and unknown attack scenarios then predict the next goal of the attacks. Two modes are applied: an offline mode where low-level alert normalization causes the aggregation of similar features and generates hyper-alerts. An online mode where a Bayesian network episode tree is constructed that learns critical multi-step attack episodes.

Prediction capabilities are also proposed by Xu et al. [19] through a vine copula approach for modeling dependencies between possible threats and devices successfully attacked. Early-warning mechanism effectiveness is critically examined through a mixed vine copula model placed in a four-dimensional time-series. Data from Center for Applied Internet Data Analysis (CAIDA)'s network telescope is used to examine precluded cyber-attack numbers and victims due to early warning. Using a markov decision process (MDP), [20] also proposes prediction capabilities for optimal allocation of finite cyber-defense assets during development and deployment of a mission-critical system. An MDP with running-on and running-off states is agile if it runs-on even after a successful attack and moves to running-off for recovery purposes only. Both [19] and [20] have no defense parameter factorization to improve future intrusion detection.

A proactive approach to preventing malicious activities before they happen based on NIDS detection output is proposed in [21] using HMM with a prime interest multi-step attacks. An experimental evaluation shows multi-step attacks prediction using an alert severity modulation through correlation. Based on the alert observation, the prediction component attempts to predict a possible future problem, then executes a set of network responses to prevent the growth of multi-step attacks. On the other hand, Zhang et al. in [22], [23], predict multi-step attacks using two different HMMs

techniques. The Baum-welch algorithm is used to optimize the first H MM m odel, f ollowed b y t he s econd H MM deployed without a training and optimization phase. Zhang et al. test their proposed model using a multi-step attack dataset with results showing an effective decrease in false-positive alerts from the untrained HMM model. In [23], the authors offer an interesting gray-box method that predicts cyber-attack 1 or 24 hours ahead using three separate datasets in three different periods. This doubled up approach is based on the time series theory (TST) for 1-hour prediction notice and extreme value theory (EVT) for a 24 hours prediction notice. Although high accuracy prediction accuracy is demonstrated, the authors note a significant l imitation o f t heir a pproach f rom t he honeypot dataset collected in their gray-box model.

Our paper picks from Mireles et al. [11] quantitative metrics that capture static metrics and produce dynamic system behavior for a formal understanding of evolutionary cyber strategies. The authors validate their proposed quantitative metrics through two real-world datasets on an open-source NIDS but without any ML training concept. The dataset and NIDS versions are only used to measure cyber-agility, while our paper includes ML to go beyond calculating numeric metrics and proactively predict defense performance.

III. Background

A. Threat Prediction

Cyber-threat prediction has seen recent research prevalence. Similar to how weather forecasts can help mitigate natural hazards, cyber-forecast and inclusion can significantly improve cyber-situations through accurate threat prediction. Existing literature studies increase or decrease of cyber-attacks in different classifications like the use of Bayesian method [24], use of a fractionally differenced autoregressive integrated moving average (FARIMA) model under long-range dependent time series data [25], extreme cyber-attack rates modeled using of time series magnitudes and inter-arrival times [26], and improved accuracy prediction through extreme time-series data values in FARIMA+generalized auto regressive conditional heterocedascity (GARCH) model [23]. An accurate forecast can only be assured through much needed but not readily available large cyber-datasets. Understanding we can never have enough information about *incognito* adversarial activities; predicting future threats based on past context can still result in warped results. Nonetheless, a comparison of NIDS performance deftness from predicted threats is a critical problem yet to be extensively researched.

B. Threat Projection

The prediction problem in Section III-A is problematic because it relies on threat projection. An examination of prediction mid existing obstacles is done in retrospect, i.e., while attacks have or are taking place [9]. Threat prediction is a subset of threat projection, i.e., predictions are projections, but projections are not necessarily predictions. We can think of it this way; a cyber-attack is a threat, but not all threats materialize into actual cyber-attacks. While it is essential to deduce cyber-attacks from an existing set of facts, threat projection gives a more realistic view of the nefarious context from unknown geographically dispersed resources [27]. Recognized as an addition to attack plan recognition, [28] was the first to propose attack projection under an enhanced set of requirements for network security.

Since then, traditional attack graphs have been used to correlate security events and represent attack scenarios from possible network vulnerabilities [29], [30]. Nevertheless, projecting an attacker's complete capabilities is not feasible; thus, our model assumes a previously successful exploit can continually compromise networked assets. This supports the realistic inclusion of publicly available cyber-attack datasets to our model, with the attack surface depending on networked assets and running services. More recently, best probability answers to dynamic attack behaviors have subsequently been produced without relying on pre-defined attack plans from matched patterns [24], [31]. However, due to the exponential growth and damage from evolved cyber-attacks, it is not enough to project or predict threats only for resource allocation. We find a research gap in dynamic NIDS performance analysis from a combination of projection and prediction applied to cyber-agility.

C. Cyber Agility

Developing cybersecurity frameworks with periodic *reactive* security patches to identify threats and protect networked assets does not occlude the ever-evolving cyber-attacks. Same as installing and updating a firewall is not enough to practically stop every cyber-attack. Recognition of this dire need has resulted in recent uptake on cyber agility as a part of cyber-defense *upgrade* [11], [32], [33]. Cyber agility dynamically facilitates timely and economic changes of underlying system parameters to defend against unknown threats proactively. Dynamic change of asymmetric system properties requires satisfaction of conflicting threat projection and prediction constraints. According to [11], the existing research gap of including cyber-agility on NIDS models can largely be attributed to difficulties in quantifying benefits versus the cost of individual or combined agility parameters.

D. Cyber Visualization

Visualization has emerged as a promising technique to demystify complex data, leveraging humans' unique perceptual capabilities. In a survey of cyber defense practitioners [34], feedback indicated that security visualizations could support analysis and communication of findings when inspecting large volumes of data. Too frequently, however, cyber-defense solutions lack capabilities required for security analysts' accurate, actionable insight with high speed [35]. Although attackers' tactics frequently change, the cyber-visualization of predictive analytics can efficiently interpret real-time network data for fast and informed decisions. Thus, our paper adds this most crucial method of simplifying the model output through a visual module.

E. Hidden Markov Model Preliminary

Static rather than dynamic metrics have long been used as a measure of NIDS performance partly because it is not completely possible to estimate all cyber-activities dynamically *on-the-fly*, as nefarious actors' *states* are directly visible only when attacks are ongoing or afterward. HMM fits this challenge as it is a statistical markov model with hidden states that can be defined by parameters $\lambda = \theta(\pi, u, v)$, where π is the prior probability, $\mathcal{U}$ is a transition matrix, and $\mathcal{V}$ is an observation matrix. Assuming that current NIDS current performance at time t is dependent only on the previous time t' performance, and NIDS performance at a future time t'', is dependent only on performance at t, HMM can predict λ hidden state sequences accurately and estimate future NIDS performance as demonstrated by [36]. Hence, given previous NIDS *static* metrics denoted as $\mathcal{N}$ at time t', and recent development of formal specification for cyber-agility as described in [5], we wish to predict future NIDS performance $\mathcal{N}^{t''}$ based on our selected dataset, using the quantitative metrics described in [37] as

$$P(\mathcal{N}^{t''}|\mathcal{Y}) = \int P\left(\mathcal{N}^{t'}|\lambda(\mathcal{U},\mathcal{V})P(\mathcal{N}^{t}|\theta\mathcal{Y})P(\mathcal{V}|\mathcal{Y})d\theta d\mathcal{Y}\right) \tag{1}$$

where $\mathcal{Y}$ represents observations in a dataset, e.g., NIDS alerts. Therefore, while existing literature has siloed modeling of adversarial behavior and cyber-agility application, our paper addresses a research gap by offering a combined approach. As summarized in Fig. 1, our model takes previous stand-alone aspects of cyber-agility to propose a novel interconnected approach.

IV. Implementation

A. Attack Projection

Cyber-attacks exhibit both deterministic and considerably more complicated stochastic patterns. The latter is prevalent in real-world scenarios. Deterministic models provide a useful approximation to handle stochastic patterns' extreme values, bursts, and temporal elements [38]. This has contributed to many time-series based forecasting of cyber-attack count data like the proposal in [39]. We also adopt a time-series approach for our attack prediction, fitted to an open-source evolved attack dataset with increased stochastic intensity over time. This provides a complex challenge in the attack projection training phase where attack intensity denoted as $\mathcal{A}$, at a selected time t, and projected attack intensity $\mathcal{A}_{(t+n)}$, must be consistent with observations in the attack dataset, for a random period n, where $0 < t < t + n$.

To solve this problem, we note cyber-attacks tend to exploit a disproportionately small set of vulnerabilities as demonstrated by [40], thereby subjecting our attack projection to include impact analysis. First, we assume all assets criticality values are known by related security analysts and thus assign an influence score $\mathcal{S}$ to all network assets depending on a host's subnet. A high level *parent* subnet e.g., a main router, that could affect many *child* subnets, is tagged with a higher $\mathcal{S}$ value. Then we calculate an attack impact score with respect to any observed attacks, on a network node s, running services z, recorded in a set $\mathcal{I} = \{\mathcal{I}_{s_1}, \mathcal{I}_{s_2}, ..., \mathcal{I}_{s_n}\}$ as

$$\mathcal{I}_{s_n} = \sum_{z \in s_n} \frac{\mathcal{S}(z,s).\alpha z}{\mathcal{S}(z,s)} \tag{2}$$

where $\mathcal{S}$ is a predefined influence score as described in section III-B, and n is the n_{th} network node.

Having calculated the impact score of attacks in our dataset, we consider a set of cyber-attacks $\mathcal{Y} = \{\mathcal{Y}_{t_1}^{(q,r,s)}, \mathcal{Y}_{t_2}^{(q,r,s)}, ..., \mathcal{Y}_{t_n}^{(q,r,s)}\}$ over a time horizon, where q is attack type and r is the nefarious agent, and s is the targeted network node. The goal is to obtain maximum attack intensity likelihood model parameters on (1) through

$$\mathcal{A}_{t_n}(\theta, \mathcal{I}_{s_n}) = P(\mathcal{Y}_{t_n} \mid \theta, \mathcal{I}_{s_n}) \tag{3}$$

to give the projected attack $\mathcal{C}_t$ in a set of attack time intervals $\mathcal{C} = \{\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2 ..., \mathcal{C}_t\}$ at time t and some random noise ω as

$$\mathcal{C}_t = f(\mathcal{A}_{t_n}) + \omega_t \tag{4}$$

B. Model for Cyber-Agility

We refer to two recent works on the use statistical methods in cyber-agility [6], [11] to formulate our problem on cyber-attack evolution $\mathcal{C}_t \in [0, \mathcal{C}_\mathcal{T}]$ where $\mathcal{T}$ is max time in a finite experiment setup. The NIDS evolution is $\mathcal{N}_t \in [0, \mathcal{N}_\mathcal{T}]$ where $\mathcal{N}_t$ is set of defense time intervals $\mathcal{N} = \{\mathcal{N}_0, \mathcal{N}_1, \mathcal{N}_2 ..., \mathcal{N}_t\}$. NIDS performance at time $\mathcal{N}_t = \mathcal{N}_{t+1}$ implies no security patch or new version was applied between time $\{\mathcal{N}_t : \mathcal{N}_{t+1}\}$. Subsequently, $\mathcal{P}_t^0\{\mathcal{N}_t : \mathcal{C}_t\}$ represents the NIDS performance success rate i.e., true positive (TPR) or $\mathcal{P}_t^1\{\mathcal{N}_t : \mathcal{C}_t\}$ failure

Table I: Summary of Key Notations

Notation	Description
$\mathcal{Y}$	Attack dataset [Ø]
$\mathcal{Y}^{(q,r,s)}$	Attack (type, agent, target node) respectively
$\mathcal{C}$	Attack time intervals [Ø]
$P(\mathcal{C})$	predicted attack intensity [Ø]
$\mathcal{N}$	Defense time intervals [Ø]
$\mathcal{P}$	Normalized NIDS performance $[0, 1]$
$\mathcal{P}_t^0$	NIDS success rate
$\mathcal{P}_t^1$	NIDS failure rate
$\mathcal{F}$	NIDS $\overline{R}$ performance over time
$\mathcal{T}$	Max time interval
t'	Previous time interval
t	Current time
t''	Future time interval
$\mathcal{L}$	NIDS evolution lagtime
$\mathcal{I}$	impact score [Ø]
$\mathcal{S}$	predefined influence score
$\subseteq$	NIDS evolution threshold
$\mathcal{X}_{N_t}$	NIDS $\overline{R}$ agility at a given time
$\mathcal{M}$	NIDS $\overline{R}$ agility [Ø]
$P(\mathcal{M}_t)$	Predicted NIDS $\overline{R}$ agility

i.e., false negative (FNR), against attack $\mathcal{C}_t$. Typically, lower values from the ranges in metrics $p \in \mathcal{P}$ normalized to $[0, 1]$ is desired i.e., high NIDS performance.

To check cyber agility of a particular NIDS, we consider high performance under evolved cyber-attacks e.g., lower values $\forall \mathcal{P}$ in $\sum_{t=1}^{\mathcal{T}} \mathcal{P}_t^0\{\mathcal{N}_t : \mathcal{C}_t\}$ or $(1 - \mathcal{P}_t^0)$ for false positives(FP). We define time taken by a cyber-attack to evolve as initial time t' and future time t'', measured at time t such that $t' < t < t''$. The NIDS performance over a time horizon is in a set $\mathcal{F} = \{\mathcal{F}_0, \mathcal{F}_1, ...\mathcal{F}_t\}$ derived from

$$\mathcal{F}_t = \begin{cases} \mathcal{P}_t^0\{\mathcal{N}_t : \mathcal{C}_t\}, & \text{if } p \in \mathcal{P} = 0 \\ \mathcal{P}_t^1\{\mathcal{N}_t : \mathcal{C}_t\}, & \text{otherwise} \end{cases} \tag{5}$$

where $\{\mathcal{N}_{t'} : \mathcal{C}_{t'}\} < \{\mathcal{N}_{t''} : \mathcal{C}_{t''}\}$. With evolved attacks, security professionals will always react to patch any new vulnerabilities discovered but the rule of thumb shows responses lag in time. We denote this lag time as $\mathcal{L}_t$, which measures time before the NIDS is patched or evolved to address decreased performance. Hence, metric $p \in \mathcal{P}$ of interest, we define a threshold θ where $0 \leq \theta \leq 1$ represent an acceptable defense evolution considering minimum λ as

$$\mathcal{L}_{\mathcal{N}_t} = min\left\{\lambda : \mathcal{F}_t\left(\mathcal{C}_{t-\lambda}, p\right) \geq \theta\right\}, \quad \lambda = \{0, 1, 2, ...T\}, t \leq \lambda \tag{6}$$

for $\lambda = 0, 1, 2, ..., T$ and $t \geq \lambda$ if such λ exists. $\mathcal{L}_{\mathcal{N}_t} = 0$ means there is no lag time in NIDS evolution response i.e., even *zero day* threats are handled which might not be practical. Hence, rather than demand immediate acceptable NIDS evolution, we can accept the average NIDS agility as

$$\mathcal{X}_{\mathcal{N}_t} = \frac{1}{(\mathcal{T} - \lambda) + 1} \sum_{t=\lambda}^{\mathcal{T}} \left(\mathcal{L}_{\mathcal{N}_t}\left(\mathcal{C}_{t-\lambda}, p\right) \geq \theta\right) \tag{7}$$

C. Cyber-Agility Prediction

The unpredictable nature of cyber-attacks evolution would makes it difficult to conduct supervised learning with unavailable reference to the next attack. Based on this assumption, we focus on getting the right balance between exploring future threats and using existing knowledge for ML-based process. HMM is a powerful modeling technique when an autonomous system states are partially observable. The traditional markov model (MM) is limited to full or single observation symbols, while HMM utilizes self-contained structures to solve a subset of problems efficiently. We store observed NIDS agility in a set $\mathcal{M} = \{\mathcal{X}_{\mathcal{N}_1^t}, \mathcal{X}_{\mathcal{N}_2^t}, ..., \mathcal{X}_{\mathcal{N}_\mathcal{T}^t}\}$.

With this in mind, our cyber-agility prediction is categorized as a *memory-less* stochastic process in which future NIDS performance does not depend on steps that lead up to the present state i.e., *the markov property*. Rather, only knowledge of observed NIDS performance is used to determine a future state's probability distribution. This particular assumption covered the reality that NIDS performance cannot be predicted

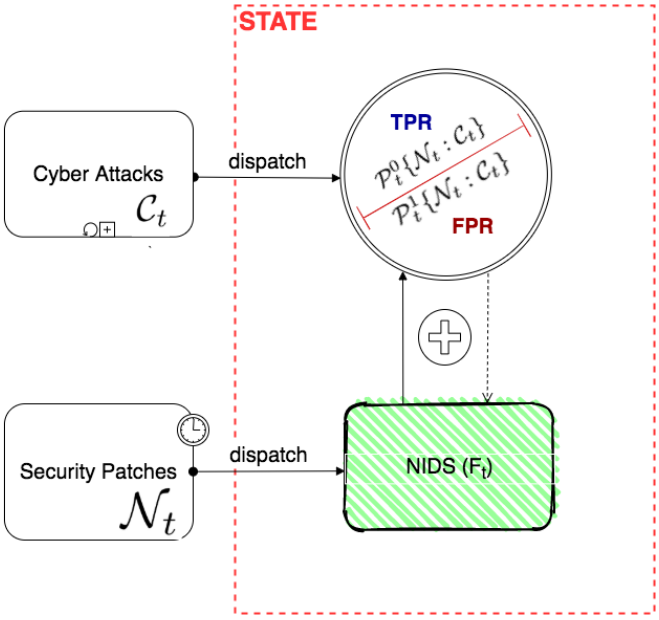


Figure 2: Summarized Agility Model.

with complete certainty based on observed metrics. If we let $\mathcal{M} = (M_t)_{t \geq 0}$ then

$$P\Big(\mathcal{M}_t = i_t \mid \mathcal{M}_{t_1} = i_{t_1}, \mathcal{M}_{t_2} = i_{t_2}..., \mathcal{M}_{t_{(n-1)}} = i_{(t-1)}\Big) = P\Big(\mathcal{M}_t = i_t \mid \mathcal{M}_{(t-1)} = i_{(t-1)}\Big) \tag{8}$$

where $\forall i_1, ..., i_n \in \mathcal{T}$ and any sequence $0 \leq t_1 < t_2 < ... < t_n$. As summarized in Fig 2, our cyber-agility model is achieved through an input combination of cyber-attacks $\mathcal{C}_t$ produced continually through the projection explained in Section 4, together with timely released NIDS security patches summarized in Table II. These inputs are examined mid hidden states through HMM on NIDS performance $\mathcal{F}_t$ and statics NIDS performance metrics depending on if the cyber-attacks are i) successfully detected, i.e., true negatives (TN), ii) false negative (FN) from missed detection of intrusive behavior, iii) incorrectly classifying benign behavior as an attack, i.e., false positives (FP), and lastly iv) correctly predicting benign activities, i.e., true positive (TP), is placed in a *confusion matrix* with transition probabilities $\mathcal{M}_{xy}$ between each state as

$$\mathcal{M}_{xy} = \begin{bmatrix} \mathcal{M}_{t_{1,1}}^{TN} & \mathcal{M}_{t_{1,x}}^{FN} \\ \mathcal{M}_{t_{y,1}}^{TP} & \mathcal{M}_{t_{x,y}}^{FP} \end{bmatrix} \tag{9}$$

where $\sum_{x=1}^{\mathcal{T}}, \mathcal{M}_{xy} = 1$, for all $x, = 1$, and t_i is the n^{th} time interval item in set $\mathcal{M}$. The conditional probability does not depend on current observation time t' of the observations, so that

$$P\Big(\mathcal{M}_{(t+k)} = y \mid \mathcal{M}_k = x\Big) = P\Big(\mathcal{M}_t = y \mid \mathcal{M}_0 = x\Big), \quad \text{where} \quad k \geq 0 \tag{10}$$

The fact that time t is continuous means the NIDS performance can move between states at any time, not just at integer times. Hence the *transition probability* is

$$P(\mathcal{M}_t^{xy}) = P(\mathcal{M}_{(t+k)} = y \mid \mathcal{M}_s = x), \quad \text{where} \quad t, k \geq 0 \tag{11}$$

We show Chapman-Kolmogorov equations [41] holds for $P(\mathcal{M}_t^{xy})$ as

$$P(\mathcal{M}_{t+k}^{xy}) = (P_t)(P_k) \iff P(\mathcal{M}_{t+k}^{xy}) = \sum_{j \in k} (P_t^{xj})(P_k^{jy}) \tag{12}$$

Proof:.

$$\begin{aligned} P(\mathcal{M}_{t+k}^{xy}) &= P(\mathcal{M}_{(t+k)} = y \mid \mathcal{M}_0 = x) \\ &= \sum_j P(\mathcal{M}_{(t+k)} = y \mid \mathcal{M}_k = j, \mathcal{M}_0 = x) \\ &\quad P(\mathcal{M}_k = j \mid \mathcal{M}_0 = x) \\ &= P(\mathcal{M}_{(t+k)} = y) P(\mathcal{M}_k = j \mid \mathcal{M}_0 = x) \\ &= \sum_j (P_t^{xj})(P_k^{jy}) \end{aligned}$$

□

D. HMM Training

When states are hidden and we are not sure of the true values of (3) from predicted attack $\mathcal{C}_t$ or (9) of $\mathcal{M}_{t+k}^{xy}$, the best alternative is to estimate our "best guess" for what $\mathcal{A}_{t_n}(\theta, \mathcal{I}_{s_n})$ and $\mathcal{M}_{xy}$, then repeat update on parameters until convergence. In order to compute the most probable sequence of hidden states $\mathcal{A}_{t_n}(\theta, \mathcal{I}_{s_n})$ and $\mathcal{M}_{xy}$ based on dataset $\mathcal{Y}$, we will use the Viterbi algorithm as summarized in pseudo-code Algorithm 1, that provides a satisfactory bit error rate performance, high speed operation and ease of implementation [42]. We propagate the Viterbi algorithm backward pass recursively to find the most likely sequence between t' and t as

$$\begin{aligned} \delta_t(\mathcal{U}) \triangleq \max_{\{\mathcal{A}_{n_1}, \mathcal{M}_{n_1}^{xy}\}, .., \{\mathcal{A}_{n-1}, \mathcal{M}_{n-1}^{xy}\}} & P\{\mathcal{A}_{1:t-1}, \mathcal{M}_{1:t-1}^{xy}\} \\ & , \{\mathcal{A}_t = \mathcal{U} \mid \mathcal{C}_{1:t}, \mathcal{M}_t^{xy} = \mathcal{U} \mid \mathcal{N}_{1:t}\} \end{aligned} \tag{13}$$

The most probable path leading from t to t' is given as

$$\delta_t(\mathcal{U}) \triangleq \max_{1 \leq t' \leq \mathcal{T}} \delta_{t-1}(t') \{\mathcal{M}^{xy} \mathcal{V}_{\mathcal{N}_{t'}}(\mathcal{U}), \mathcal{A}_t \mathcal{C}_v(u)\} \tag{14}$$

One advantage over the Viterbi algorithm is we can avoid underflow since $\log \max = \max \log$ through

$$\begin{aligned} \log \delta_t(u) \triangleq \max_{t''} \log \delta_{t-1}(t'') + \{\log \mathcal{M}^{xy} \\ + \log \mathcal{V}_{\mathcal{N}_{t'}}, \log \mathcal{A}_t + \log \mathcal{V}_{\mathcal{C}_{t'}} \end{aligned} \tag{15}$$

Algorithm 1 HMM training for attack prediction

Require: $\mathcal{Y}$, $\mathcal{T}$,$\mathcal{C}$,$\mathcal{U}$, $\mathcal{V}$, π
1: **Initialize:** $\delta_1 = \pi \odot \mathcal{C}(\mathcal{Y}_{\mathcal{U}_1}), \alpha_1$
2: **for** $t \quad 1, \mathcal{T}$ **do**
3: **for** $i \leftarrow 1, \mathcal{Y}_n$ **do**
4: $[\alpha_t^i, \delta_t^i] = \max(\log \delta_{(t-1)}^{(:)} + \log \mathcal{C}_i) + \log \mathcal{V}_{\mathcal{Y}_t}^i$
5: $\mathcal{J}_{\mathcal{T}}^* = \arg\max(\delta_{\mathcal{T}})$
6: **for** $t \quad \mathcal{T}_{n-1}, 1$ **do**
7: $\mathcal{J}_{\mathcal{T}}^* = \alpha_{t+1} \mathcal{J}_{t+1}^*$;
return $(\mathcal{J}_{1:\mathcal{T}}^*)$

Algorithm 2 HMM training for agility prediction

Require: $\mathcal{F}$, $\mathcal{T}$,$\mathcal{P}$,$\mathcal{M}^{xy}$,$\mathcal{N}$,π
1: **Initialize:** $\delta_1 = \pi \odot \mathcal{P}(\mathcal{F}_{\mathcal{M}^{xy}}), \alpha_1$
2: **for** $t \quad 1, \mathcal{T}$ **do**
3: **for** $i \leftarrow 1, \mathcal{Y}_n$ **do**
4: $[\alpha_t^i, \delta_t^i] = \max(\log \delta_{(t-1)}^{(:)} + \log \mathcal{P}_i) + \log \mathcal{N}_{\mathcal{F}_t}^i$
5: $\mathcal{K}_{\mathcal{T}}^* = \arg\max(\delta_{\mathcal{T}})$
6: **for** $t \quad \mathcal{T}_{n-1}, 1$ **do**
7: $\mathcal{K}_{\mathcal{T}}^* = \alpha_{t+1} \mathcal{K}_{t+1}^*$
return $(\mathcal{K}_{1:\mathcal{T}}^*)$

E. Prediction Accuracy

In this section, the probability and frequency of both future threat occurrence and future NIDS average performance are estimated, using the transition matrix created and the initial probability vector described in section IV-B. The attack dataset and NIDS performance metrics give the deterministic model parameters needed for our agility HMM training in Algorithm 2, making it possible to predict agility as shown in Algorithm 3 corresponding to the observations in dataset $\mathcal{Y}$ and $\mathcal{F}$. That means, the NIDS performance observed at a time t, should be same as predicted NIDS performance during training. From the results, we will be able to calculate accuracy through:

F. Selected NIDS

In this section, we describe the defense tools applied to ascertain our proposed model. The selection criteria depend on NIDS evolution in terms of patches or different release versions. As we reviewed existing research NIDS models and keen to ensure replicability of our model, we selected open-source NIDS tools analyzed and compared in survey [43]. The NIDS tools publish long-term support (LTS) release version $\approx$ every year, e.g., 3.0.0, and feature release, e.g., 3.1.0 $\approx$ within a particular year. For the sake of relevancy and based on what is available online, we select the first and last release in a year for four years, as summarized in Table II.

1) Zeek [13]: formerly known as Bro, is a UNIX based NIDS that passively monitors network traffic by parsing and executing event-oriented analysis for known malicious patterns.

2) OSSEC [14]: compares specified rules or signatures against analyzed event logs from a group of agents, e.g., log data points, and responds through multiple mechanisms like third party support portals or *self-healing* firewall policies.

Algorithm 3 Cyber-attacks and cyber-agility prediction

Require: $\mathcal{C}$,$\mathcal{F}$,$\mathcal{N}$, $\mathcal{M}^{xy}$,$\mathcal{T}$,$\mathcal{Y}$,θ
Initialize: Algorithm {1 & 2}, Penalty $\lambda = .001$
$\mathcal{C}, P(\mathcal{M}^{xy})$ $\emptyset$
for $t \in \mathcal{T}$ **do**
  for each datapoint $\in \mathcal{Y}_t$ **do**
    Compute $\mathcal{I}_{s_t}$(Solve(2))by forward/
      backward propagation
    Compute $\mathcal{A}_t$(Solve(3))
    Update θ using λ
  $\mathcal{C}_t$ Predicted value
  for each datapoint $\in \mathcal{F}_t$ **do**
    Compute $\mathcal{L}_{\mathcal{N}_t}$(Solve(5))by forward/
      backward propagation
    Compute $\mathcal{X}_{\mathcal{N}_t}$(Solve(7))
    Update θ using λ
  $P(\mathcal{M}_t^{xy}) \leftarrow$ Predicted value
return $(\mathcal{C}_{1:\mathcal{T}} : P(\mathcal{M}_{1:\mathcal{T}}^{xy}))$

3) Suricata [15]: developed by the Open Information Security Foundation in December 2009, Suricata is designed to use rules from different resources and functionality such as Snort VRT, Snort logging, rule language options, and IPv6.

G. Attack Dataset

Existing research is having a significant challenge in finding comprehensive and valid datasets to test and evaluate proposed techniques. In order to continue placing high relevance to our model, we required an attack dataset that is not somehow obsolete and but relevant to present-day attacks with a reasonable time collection period. We selected the most recent and publicly available cyber-attack datasets Kent2015 [44], presented in survey [45]. Compared to other datasets collection period of ≈ 14 days or less, Kent was collected at Los Alamos National Laboratory network for 58 days and has over 130 million flows of unidirectional flow-based network traffic with several host-based log files. Kent's $\approx$ 12GB five data elements compressed dataset, contains $1,648,275,307$ total events for $12,425$ users, $17,684$ computers, and $62,974$ processes. The collection duration and number of attacks represent a reasonable attack evolution for our paper.

V. Performance Evaluation

This section describes i) the selected NIDS used for research cyber-agility, ii) the selected dataset to train our cyber-threats projection and prediction model, iii) experiment setup, and iii) discussions of evaluation metrics for empirical studies. We introduce parameter settings of our model and show the resulting baseline method for our accuracy evaluation.

A. Experiment

To evaluate our proposed model, we implement our experiment's main components on four MS Windows server 2016 Intel(R) Xeon CPU E5 – 2676 v3 @2.40GHz with 12GB RAM. We set up a virtual network topology of 360 different virtual computing devices sub-divided into 14 subnets. At least 50% of the devices are virtual computer nodes running a balanced mix of MS Windows v10 and Ubuntu v20.04. During the training period, we run as-is a section of cyber-attacks in the selected attack dataset from $t = 1$ to an arbitrary time $t+n$. We match the attacks to different release versions of selected NIDS and collect *statics* metrics through intrusion alerts. After recording the training results, we run a full simulation of the attack dataset from a period $t+n$ to max experiment time $\mathcal{T}$, where $1 \leq t+n \leq \mathcal{T}$, and obtain prediction.

As shown in Fig 4, our threat prediction score was on average consistent with observations in the attack dataset apart from the start of the experiment where threshold θ has not been adjusted based on λ. However, our primary goal is to achieve a similar or higher prediction score on NIDS agility. We perform the same process on the simulated nodes by running the attack dataset threats up to a randomly selected time against different release versions of the NIDS discussed in section IV-F. We further compare predicted metrics versus expected output under the consideration that an attack day represents 25 NIDS evolution days. This is because the release versions of our selected NIDS span a total of 1461 days, i.e., 4 years, and our attack dataset spans 58 days.

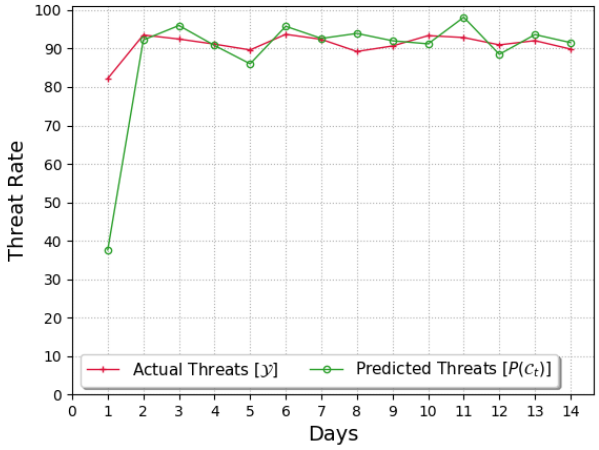


Figure 4: Actual vs Predicted Threats

To analyze our model performance, we consider both accuracy and F1 score based on TP, FP, TN, and FN since accuracy would simply consider the number of correct predictions over total observations in the dataset. Accuracy, therefore, would not consider that different cyber-systems have differing FP and FN costs. F1 gives us a harmonic mean of our model's precision and recall. We obtain the F1 score from

$$F_\beta = \frac{TP.\left(1+\beta^2\right)}{TP.\left(1+\beta^2\right) + FN.\beta^2 + FP} \tag{16}$$

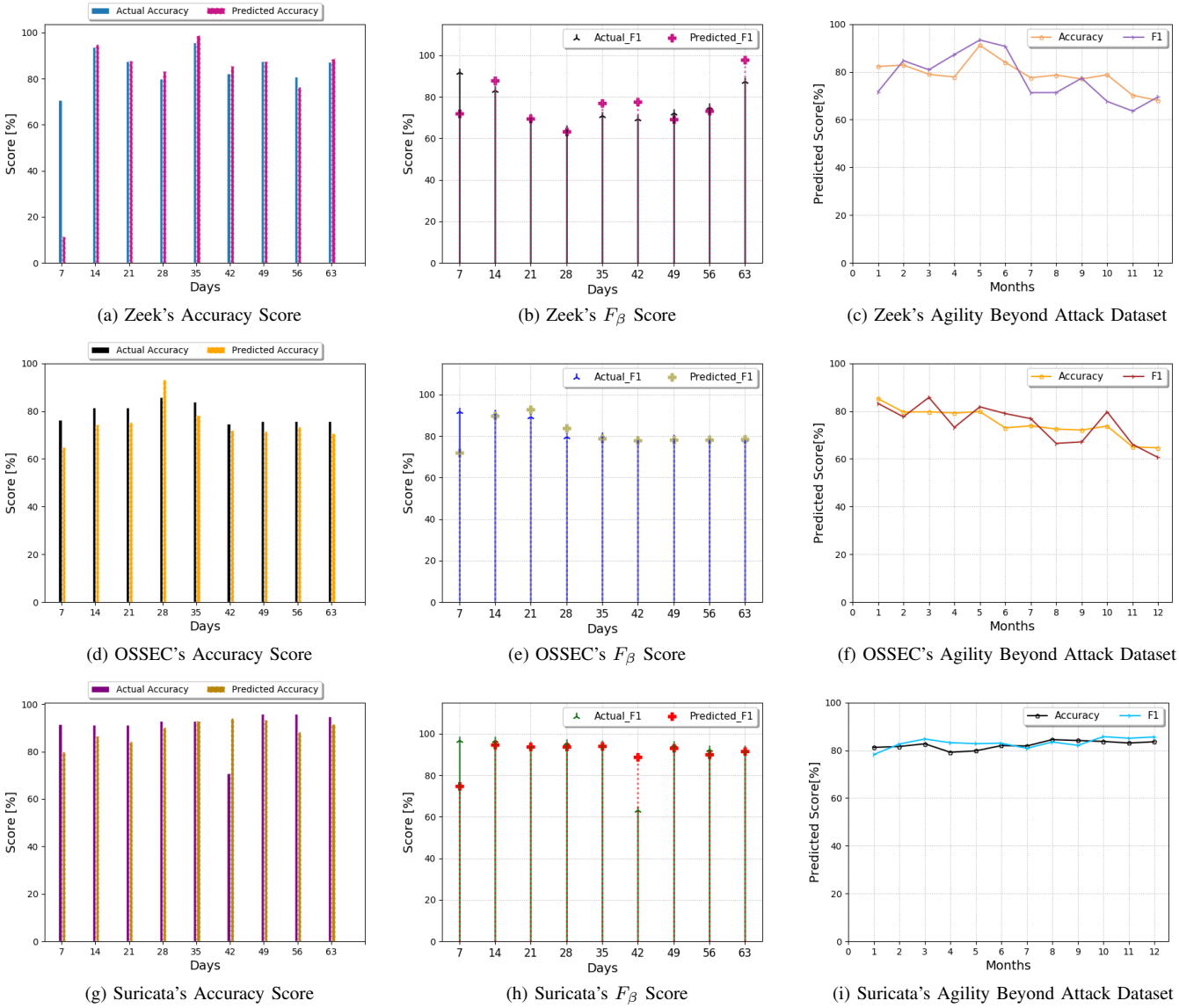
where $\beta = 1$ and obtain accuracy score from

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{17}$$

Fig 3 shows our model achieves a high prediction accuracy on all selected NIDS with each NIDS performance graph

Table II: Selected NIDS Feature Version and Release Date

NIDS	Year	First Version	Release Date(t')	Last Version	Release Date(t'')
Zeek [13]	2017	v2.5.1-beta	June,07	v2.5.2	October,16
	2018	v2.5.3	February,14	v2.6-beta3	November,15
	2019	v2.6.0	July,08	v3.0.1	December,10
	2020	v3.1.0-rcI	February,10	v3.1.4	June,19
OSSEC [14]	2017	v2.9.0	February,09	v2.9.3	December,23
	2018	v2.9.4	June,20	v3.1.0	October,12
	2019	v3.2.0	February,05	v3.5.0	November,18
	2020	v3.6.0	February,14	v3.6.0-update	July,17
Suricata [15]	2017	v3.2.2	June,07	v4.0.3	December,08
	2018	v4.0.4	February,14	v4.1.2	December,21
	2019	v4.0.7	March,07	v4.1.4	October,21
	2020	v4.1.6	February,13	v5.0.3	April,28



(a) Zeek's Accuracy Score (b) Zeek's F_β Score (c) Zeek's Agility Beyond Attack Dataset

(d) OSSEC's Accuracy Score (e) OSSEC's F_β Score (f) OSSEC's Agility Beyond Attack Dataset

(g) Suricata's Accuracy Score (h) Suricata's F_β Score (i) Suricata's Agility Beyond Attack Dataset

Figure 3: NIDS performance on a 9 week attack dataset and further 12 months agility prediction against release versions from 2017 to 2020.

represented separately for a clear depiction of the predicted values. The best accuracy prediction is achieved on Suricata, partly attributed to the almost constant software upgrade and patch time, compared to the other selected NIDS. The worst performance recorded is on OSSEC and can partially be attributed to minimal major release versions. The experiment gets to an end with only a minor update applied for OSSEC. The highest performing NIDS is Suricata, as shown in Fig 3i, whereby, assuming all experiment constraints and assumptions hold for 1 year, i.e., continue patching of new vulnerabilities at the same rate, Suricata's agility remains on high levels.

VI. Visualization

Big network data has made security analysts face the "Needle in a Haystack" problem when analyzing cyber-security events. Thus, cyber-visualization has become an essential aid in mediating interactions of different points of view for the same data [46]. This paper includes an exploratory proof-of-concept prototype developed as a web service that queries required data from virtual nodes setup and pre-defined text files to send defined output to a visual display module. The front-end is developed through a combination of JavaScript and C# published as a WebGL project from new Unity software Entity Component System (ECS) [47] with a sample view shown in Fig 5.

We first develop a dashboard to summarize the experiment setup depending on the users' environment. The number of network devices under analysis is given in the first dashboard bar; the NIDS versions selected, a summary of attack dataset period, and network subnets. For our case, these summaries are 360 devices, NIDS version 2017-2020, Kent2015 dataset for 58days, and 14 subnets, respectively. Additionally, A search from the standard search bar marked by (s), can be done to show the exact details of the NIDS versions selected. The other respective marked columns are summarized below.

1) *Select NIDS*: allows upload of NIDS version events similar to the format in Table II and specifications of what a particular patch release was meant to handle.
2) *Attack dataset*: once the NIDS versions upload is done, next is to upload an attack dataset with the same format and correlation as our attack dataset described in Section IV-G.
3) *Network simulation*: randomly selects the main operating systems and possible running services in nodes depending on the total number of nodes selected. Such parameters can be customized to fit the exact required network model, including device types and network subnets. After the network simulation is complete, the dashboard marked as (db) will appear to summarize information about the network model. A node's state depending on total running services under simulated cyber-attack is denoted with color red to mean severe services affected, yellow to show services project to fall victim next to the cyber-attack and green to show services currently unaffected.
4) *HMM training*: performs our proposed threat projection and cyber-agility prediction while the center display marked (cd), shows top affected nodes and subnets, since it would not be practical to display all nodes including healthy nodes especially as expected, total network nodes could run in thousands.
5) *Reports*: produces Fig 3 based on the results obtained.
6) *Scroll bar*: assists in scrolling between prediction months with a maximum of 12 months.

VII. Conclusion

Networked systems have frequently included vast attack surfaces from inherent vulnerabilities, leading to a colossal incursion of sophisticated cyber-attacks from a well entangled nefarious ecosystem. In an effort to diffuse this exponential growth in number and complexity of cyber-threats, predictive analytics has been applied to threat forecasting in place of traditional reactive cyber-defense mechanisms. On the one hand, predictive analysis has proven successful in predicting the time and quantity of attacks while network intrusion detection systems bolster secure network perimeters. If it is possible to forecast cyber-threats based on time-series events correctly and effectively assign cyber-defense resources as needed, why does the *cyber-chase* continues to favor cyber-attackers? A gap exists in the holistic estimation of network defense tools' performance under an uncertain or hidden state of future threats.

In this paper, an HMM-based machine learning algorithm, best known for robust prediction of temporal relationships mid noise and training brevity, is used to address the identified research gap through a novel combination of cyber-threat projection and cyber-agility prediction. In order to make our research modern and relevant, our experiment applies a recent publicly available cyber-attack dataset to 3 open-source NIDS, namely Zeek, OSSEC, and Suricata, and demonstrates high prediction accuracy for a future period. Research on cyber-defense can apply our framework to different datasets and proposed models to draw more insights on the much-needed cyber-agility analysis. Our experiment can also be replicated for planned pre-defined security patch release dates instead of patches released that only*react* to exploited vulnerabilities.

Our future works is based on the understanding that any prediction mechanism is as good as the training dataset. To check our models' validity, HMM is applied to a selected dataset with a basic assessment of numerically quantified details but scanty awareness of exposed attack surface. Thus, our assumption is limited to vulnerabilities remaining constant and only change based on previous successful attacks or NIDS version. Our future works shall include greater collaboration on attack datasets collated for more comprehensive dynamic cyber-threats attack surfaces under extended periods and extensive cyber-attack surface definition.

Acknowledgment

This work was supported in part by the US NSF under grants CNS/SaTC 2039583, CMMI 2036359 and HRD 1828811, and by the DoD Center of Excellence in AI and Machine Learning (CoE-AIML) at Howard University under Contract Number W911NF-20-2-0277.

Figure 5: Visualization Overview.

References

[1] M. Branlat, A. Morison, and D. Woods, "Challenges in managing uncertainty during cyber events: Lessons from the staged-world study of a large-scale adversarial cyber security exercise," in *Human Systems Integration Symposium*, 2011, pp. 10–25.

[2] M. Abdlhamed, K. Kifayat, Q. Shi, and W. Hurst, "Intrusion prediction systems," in *Information Fusion for Cyber-Security Analytics*. Springer, 2017, pp. 155–174.

[3] J.-H. Cho, S. Xu, P. M. Hurley, M. Mackay, T. Benjamin, and M. Beaumont, "Stram: Measuring the trustworthiness of computer-based systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–47, 2019.

[4] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Computing Surveys (CSUR)*, vol. 49, no. 4, pp. 1–35, 2016.

[5] J.-H. Cho, P. M. Hurley, and S. Xu, "Metrics and measurement of trustworthy systems," in *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE, 2016, pp. 1237–1242.

[6] L. M. Marvel, S. Brown, I. Neamtiu, R. Harang, D. Harman, and B. Henz, "A framework to evaluate cyber agility," in *MILCOM 2015-2015 IEEE Military Communications Conference*. IEEE, 2015, pp. 31–36.

[7] S. Pfleeger and R. Cunningham, "Why measuring security is hard," *IEEE Security & Privacy*, vol. 8, no. 4, pp. 46–54, 2010.

[8] M. Bijone, "A survey on secure network: intrusion detection & prevention approaches," *American Journal of Information Systems*, vol. 4, no. 3, pp. 69–88, 2016.

[9] S. J. Yang, H. Du, J. Holsopple, and M. Sudit, "Attack projection," in *Cyber Defense and Situational Awareness*. Springer, 2014, pp. 239–261.

[10] A. A. Ahmed and N. A. K. Zaman, "Attack intention recognition: A review." *IJ Network Security*, vol. 19, no. 2, pp. 244–250, 2017.

[11] J. D. Mireles, E. Ficke, J.-H. Cho, P. Hurley, and S. Xu, "Metrics towards measuring cyber agility," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3217–3232, 2019.

[12] M. Husák, J. Komárková, E. Bou-Harb, and P. Čeleda, "Survey of attack projection, prediction, and forecasting in cyber security," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 640–660, 2018.

[13] J. Siwek, "The zeek network security monitor." [Online]. Available: https://zeek.org/

[14] S. Shinn, "World's most widely used host intrusion detection system - hids." [Online]. Available: https://www.ossec.net/

[15] O. I. S. Foundation. [Online]. Available: https://suricata-ids.org/

[16] R. Colbaugh and K. Glass, "Predictive defense against evolving adversaries," in *2012 IEEE International Conference on Intelligence and Security Informatics*. IEEE, 2012, pp. 18–23.

[17] S. Abraham and S. Nair, "Exploitability analysis using predictive cybersecurity framework," in *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*. IEEE, 2015, pp. 317–323.

[18] A. Ahmadian Ramaki and A. Rasoolzadegan, "Causal knowledge analysis for detecting and modeling multi-step attacks," *Security and Communication Networks*, vol. 9, no. 18, pp. 6042–6065, 2016.

[19] M. Xu, L. Hua, and S. Xu, "A vine copula model for predicting the effectiveness of cyber defense early-warning," *Technometrics*, vol. 59, no. 4, pp. 508–520, 2017.

[20] L. L. Njilla, C. A. Kamhoua, K. A. Kwiat, P. Hurley, and N. Pissinou, "Cyber security resource allocation: a markov decision process approach," in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, 2017, pp. 49–52.

[21] A. S. Sendi, M. Dagenais, M. Jabbarifar, and M. Couture, "Real time intrusion prediction based on optimized alerts with hidden markov model," *Journal of networks*, vol. 7, no. 2, p. 311, 2012.

[22] Y. Zhang, D. Zhao, and J. Liu, "The application of baum-welch algorithm in multistep attack," *The Scientific World Journal*, vol. 2014, 2014.

[23] Z. Zhan, M. Xu, and S. Xu, "Predicting cyber attack rates with extreme values," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1666–1677, 2015.

[24] C. Ishida, Y. Arakawa, I. Sasase, and K. Takemori, "Forecast techniques for predicting increase or decrease of attacks using bayesian inference," in *PACRIM. 2005 IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2005.* IEEE, 2005, pp. 450–453.

[25] Z. Zhan, M. Xu, and S. Xu, "Characterizing honeypot-captured cyber attacks: Statistical framework and case study," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1775–1789, 2013.

[26] C. Peng, M. Xu, S. Xu, and T. Hu, "Modeling and predicting extreme cyber attack rates via marked point processes," *Journal of Applied Statistics*, vol. 44, no. 14, pp. 2534–2563, 2017.

[27] E. Muhati, D. B. Rawat, M. Garuba, and L. Njilla, "Cyvi: Visualization of cyber-attack and defense effects in geographically referenced networks," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2020, pp. 1–4.

[28] C. W. Geib and R. P. Goldman, "Plan recognition in intrusion detection systems," in *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01*, vol. 1. IEEE, 2001, pp. 46–55.

[29] X. Qin and W. Lee, "Attack plan recognition and prediction using causal networks," in *20th Annual Computer Security Applications Conference*. IEEE, 2004, pp. 370–379.

[30] L. Wang, A. Liu, and S. Jajodia, "Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts," *Computer communications*, vol. 29, no. 15, pp. 2917–2933, 2006.

[31] D. S. Fava, S. R. Byers, and S. J. Yang, "Projecting cyberattacks through variable-length markov models," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 359–369, 2008.

[32] P. McDaniel, T. Jaeger, T. F. La Porta, N. Papernot, R. J. Walls, A. Kott, L. Marvel, A. Swami, P. Mohapatra, S. V. Krishnamurthy *et al.*, "Security and science of agility," in *Proceedings of the First ACM Workshop on Moving Target Defense*, 2014, pp. 13–19.

[33] J. H. H. Jafarian, E. Al-Shaer, and Q. Duan, "Spatio-temporal address mutation for proactive cyber agility against sophisticated attackers," in *Proceedings of the First ACM Workshop on Moving Target Defense*, 2014, pp. 69–78.

[34] A. D'Amico, L. Buchanan, D. Kirkpatrick, and P. Walczak, "Cyber operator perspectives on security visualization," in *Advances in Human Factors in Cybersecurity*. Springer, 2016, pp. 69–81.

[35] D. Staheli, T. Yu, R. J. Crouser, S. Damodaran, K. Nam, D. O'Gwynn, S. McKenna, and L. Harrison, "Visualization evaluation for cyber security: Trends and future directions," in *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, 2014, pp. 49–56.

[36] M. Johansson and T. Olofsson, "Bayesian model selection for markov, hidden markov, and multinomial models," *IEEE signal processing letters*, vol. 14, no. 2, pp. 129–132, 2007.

[37] Z. Ghahramani, "An introduction to hidden markov models and bayesian networks," in *Hidden Markov models: applications in computer vision*. World Scientific, 2001, pp. 9–41.

[38] K. R. Rohloff and T. Bacşar, "Deterministic and stochastic models for the detection of random constant scanning worms," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 18, no. 2, pp. 1–24, 2008.

[39] J. Viinikka, H. Debar, L. Mé, A. Lehikoinen, and M. Tarvainen, "Processing intrusion detection alert aggregates with time series modeling," *Information Fusion*, vol. 10, no. 4, pp. 312–324, 2009.

[40] R. Harang and A. Kott, "Burstiness of intrusion detection process: Empirical evidence and a modeling approach," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2348–2359, 2017.

[41] G. Haag, "Derivation of the chapman–kolmogorov equation and the master equation," in *Modelling with the Master Equation*. Springer, 2017, pp. 39–61.

[42] K. Gupta, P. Ghosh, R. Piplia, and A. Dey, "A comparative study of viterbi and fano decoding algorithm for convolution codes," in *AIP Conference Proceedings*, vol. 1324, no. 1. American Institute of Physics, 2010, pp. 34–38.

[43] S. Kumar, "Survey of current network intrusion detection techniques," *Washington Univ. in St. Louis*, pp. 1–18, 2007.

[44] A. D. Kent, "Comprehensive, multi-source cyber-security events data set," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2015.

[45] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.

[46] C. Zhong, J. Yen, P. Liu, R. F. Erbacher, C. Garneau, and B. Chen, "Studying analysts? data triage operations in cyber defense situational analysis," in *Theory and models for cyber situation awareness*. Springer, 2017, pp. 128–169.

[47] L. Meijer, "On dots: Entity component system - unity software," https://blogs.unity3d.com/2019/03/08/on-dots-entity-component-system, 2019.

Eric Muhati received the B.E. degree in CS from Strathmore University, Kenya and Master's degree from Georgetown University, USA. He is currently working towards a Ph.D. in CS degree with the Department of Electrical Engineering and Computer Science, Howard University, USA, where he is a Graduate Research Assistant at the Howard University Data Science & Cybersecurity Center (DSC^2). His Ph.D. research addresses dynamic network security through augmented machine learning scaled with cyber visualization. His interests cover the broader area of operational cyber-security, including threat detection, cyber-attacks characterization, network data mining, applied machine learning, cyber-security for critical infrastructure/cyber-physical-systems(CPS)/Internet-of-Things(IoT, and cyber-data visualization.

Danda B. Rawat (*IEEE Senior Member, 2013*) is a Full Professor in the Department of Electrical Engineering & Computer Science (EECS), Director of the Howard University Data Science and Cybersecurity Center, Director of DoD Center of Excellence in AI/ML (CoE-AIML), Director of Cyber-security and Wireless Networking Innovations (CWiNs) Research Lab, Graduate Program Director of Graduate CS Programs and Director of Graduate Cybersecurity Certificate Program at Howard University, Washington, DC, USA. Dr. Rawat is engaged in research and teaching in the areas of cybersecurity, machine learning, big data analytics and wireless networking for emerging networked systems including cyber-physical systems, Internet-of-Things, multi domain battle, smart cities, software defined systems and vehicular networks. His professional career comprises more than 18 years in academia, government, and industry. He has secured over $16 million in research funding from the US National Science Foundation (NSF), US Department of Homeland Security (DHS), US National Security Agency (NSA), US Department of Energy, National Nuclear Security Administration (NNSA), DoD and DoD Research Labs, Industry (Microsoft, Intel, etc.) and private Foundations. Dr. Rawat is the recipient of NSF CAREER Award in 2016, Department of Homeland Security (DHS) Scientific Leadership Award in 2017, Researcher Exemplar Award 2019 and Graduate Faculty Exemplar Award 2019 from Howard University, the US Air Force Research Laboratory (AFRL) Summer Faculty Visiting Fellowship in 2017, Outstanding Research Faculty Award (Award for Excellence in Scholarly Activity) at GSU in 2015, the Best Paper Awards (IEEE CCNC, IEEE ICII, BWCA) and Outstanding PhD Researcher Award in 2009. He has delivered over 20 Keynotes and invited speeches at international conferences and workshops. Dr. Rawat has published over 200 scientific/technical articles and 10 books. He has been serving as an Editor/Guest Editor for over 50 international journals including the Associate Editor of IEEE Transactions of Service Computing, Editor of IEEE Internet of Things Journal, Associate Editor of IEEE Transactions of Network Science and Engineering and Technical Editors of IEEE Network. He has been in Organizing Committees for several IEEE flagship conferences such as IEEE INFOCOM, IEEE CNS, IEEE ICC, IEEE GLOBECOM and so on. He served as a technical program committee (TPC) member for several international conferences including IEEE INFOCOM, IEEE GLOBECOM, IEEE CCNC, IEEE GreenCom, IEEE ICC, IEEE WCNC and IEEE VTC conferences. He served as a Vice Chair of the Executive Committee of the IEEE Savannah Section from 2013 to 2017. Dr. Rawat received the Ph.D. degree from Old Dominion University, Norfolk, Virginia. Dr. Rawat is a Senior Member of IEEE and ACM, a member of ASEE and AAAS, and a Fellow of the Institution of Engineering and Technology (IET). He is an ACM Distinguished Speaker.