Predicting Adversarial Cyber Intrusion Stages Using Autoregressive Neural Networks

Aunshul Rege[1], Zoran Obradovic[2], Nima Asadi[2], Edward Parker[1],
Rohan Pandit[2], Nicholas Masceri[1], and Brian Singer[1]

[1]Department of Criminal Justice, Temple University
[2]Computer and Information Sciences Department, Temple University

***Abstract*—Current approaches to cybersecurity are response-driven and ineffective as they do not account for dynamic adversarial movement. Using empirical evidence of observations done at two Red Team-Blue Team cybersecurity training exercises held at Idaho National Laboratory (INL) and the Michigan Cyber Range, this paper predicts how adversaries move during cyberattacks. In this study, a framework for temporal analysis of the observations is provided to make predictions of how adversaries progress through cyberattacks. We utilized and compared four different predictive models to make temporal predictions of how adversaries progress through cyberattacks: the Non-linear autoregressive model (NAR), the Non-linear autoregressive exogenous model (NARX), the autoregressive neural networks for multi-steps-ahead prediction, and autoregressive integrated moving average (ARIMA). The obtained empirical results provide evidence that the trained models are able to capture different variations in adversarial movement across the two datasets with reliable accuracy.**

1. Introduction

Rapidly advancing technologies have resulted in highly interconnected information networks and integrated systems, which has made them more accessible and vulnerable to cyberattacks [2]. Advanced Persistent Threats (APTs) have increasingly been targeting critical infrastructures, such as power grids, transportation, and water, constantly circumventing traditional reactive security measures, and resulting in large and costly damages [1]. Organizations thus need to develop defenses that can respond rapidly, proactively, and dynamically to more effectively manage APTs [2]. Many scholars have researched the area of proactive or anticipatory defense, such as moving target defense techniques that use spatio-temporal randomization to distort attackers view of the network [3], and bipartite graph-based machine learning algorithms and synthetic data learning method, which serve as proactive filter-based methods for network defense [2]. These important technical contributions aid in the paradigm shift from reactive to anticipatory cybersecurity. However, they are technical in nature and minimize the significance of adversarial trajectories. This paper uses datasets of qualitative observations conducted at two cybersecurity training events: (i) a US Industrial Control Systems Computer Emergency Response Team's (ICS-CERT) Red Team-Blue Team cybersecurity training exercise held at Idaho National Laboratory (INL) in 2014, and (ii) a force on force (paintball) exercise held at the 2015 North American International Cyber Summit (NAICS). The goal of this study is to provide a framework for prediction of how adversaries move as cyberattacks progress. Therefore, we utilize and compare a number of machine learning algorithms for making temporal predictions of the adverserial cyber intrusion stages. This paper is structured as follows. Section 2 discusses the 12 stages of adversarial movement, known as the intrusion chain model. The third section outlines the methodology of creating the time series data and predictive models. Next, the prediction results are presented and discussed. Finally, this paper discusses the findings and possible implications for the intrusion chain model, as well as the importance and temporal characteristics of certain intrusion stages.

2. Intrusion Chain Model for Adversarial Attack Trajectories

The technical domain offers intrusion chain models that capture the step-by-step process of cyberattacks.

We use the 12-step cyberintrusion chain model put forward by [1] (displayed in Figure 1 below), as it allows for (i) thorough analysis of which stages adversaries progress through adversarial movement, (ii) a means to assess how much time adversaries spend on different stages, (iii) exploring the possibly iterative nature of the cyberattack process through its cyclical structure, and (iv) incorporating human behavior (stages 2 and 4). First, adversaries select their targets. Second, they find partners that complement and supplement their own skill sets to form alliances. Adversaries then design and build their attack vectors and/or gather toolkits necessary to execute attacks. Fourth, adversaries obtain target infrastructure blueprints, identify target vulnerabilities, and employ social engineering practices. Fifth, adversaries gather information on any security protocols set in place by defenders that they may encounter. Doing so allows them to create appropriate evasion and response plans. Next, adversaries will deploy their attack vectors, skills, and

Digital Object Identifier 10.1109/MIS.2018.111145153 1541-1672/$26.00 2018 IEEE

Figure 1. Intrusion Chain Model [1]

knowledge to gain a foothold into the target environment. In the seventh stage, adversaries gain preliminary access into the targeted environment that allows them to install malware. Adversaries then establish more points of access into the targeted environment and gain access to additional systems that will increase their access and control. Tenth, adversaries who want to persist in their attacks will strengthen their presence by gaining credentials, and using these to move laterally and deeper into the targeted environment. This pivoting and lateral movement allows adversaries to establish control over as many different parts of the system as possible. Finally, adversaries remove data and/or accomplish their objectives and remove evidence of their presence and actions in the targeted environment.

3. Research Sites for Data Collection

Red team-blue team exercises (RTBTEs) are often used in the cybersecurity arena for training purposes and involves one group of security experts (red team) attacking a computer system, while the opposing group (blue team) defends it [5]. RTBTEs serve as a learning platform for participants to better understand vulnerabilities, points of attacks, how best to secure and defend systems in real-time, how to manage limited resources, and how to ensure system confidentiality, integrity, and availability are maintained during cyberattacks [5]. RTBTEs offer a rich platform to do social science field research, where researchers can observe the complex phenomena of real-time adversarial movement, adaptations, and group dynamics. This study used two RTBTE research sites to collect data.

Dataset Research Site 1: The United States Industrial Control Systems Computer Emergency Response Team (ICS-CERT) offers cybersecurity training events hosted at Idaho National Laboratory (INL) (henceforward referred to as ICS-CERT/INL). An 8-hour Red Team/Blue Team exercise (RTBTE) was conducted where Red Team attacked cyber-physical systems that had to be defended by the Blue Team. The Red Team was randomly assembled with ten members who were a mix of system administrators, control systems engineers, and information technology specialists. Researchers observed the Red Team over the 8 hour exercise.

Dataset Research Site 2: Alphaville, a robust virtual environment provided by the Merit Network and the Michigan Cyber Range, mirrors services and information found in small cities and has five locations. Each of these locations (a school, a library, a city hall, a small business, and a power company) has servers and firewalls with intentional vulnerabilities, making Alphaville an ideal platform for cybersecurity training exercises. During the 2015 North American International Cyber Summit (NAICS), a force on force (paintball) exercise was conducted, where all teams had to attack and defend. Here teams of five participants battled to (i) penetrate and control Alphaville's network, critical servers and firewalls and (ii) defend these controlled assets from rival teams. Researchers observed one of the competing teams that had four members. The exercise lasted for 5 hours, during which the team attempted to control various parts of Alphaville's infrastructure.

4. Methodology

4.1. Qualitative Data Collection and Field Research

Both venues identified above served as the field sites, where researchers conducted detailed observations and interviews with participants. In the first research site, researchers observed the Red Team over the 8 hour exercise, and in the second research site, researchers observed one of the competing teams during the 5 hour exercise. While interviews were conducted before, during, and after the exercise to supplement and complement the observations, the bulk of the data at both sites came from observations. Once RTBTEs commenced, participants often became focused on the exercise and the researchers did not wish to break their concentration or disrupt their efforts by asking interview questions.

The researchers assigned each member's actions (in both datasets) to the 12 stage model. The researchers used interviews conducted during and after the exercise to ensure that this assignment was done correctly. The two datasets were then analyzed by transforming the observations from the written up field notes into the time series representing the amount of time the teams spent on each of the 12 intrusion chain stages.

4.2. Time Series Generation

In order to obtain the time series, we used the recorded time stamp of the start and end times of the intrusion stages where each time point in the time series represents 1 minute time span. Therefore, for each one minute time point, the value of each time series is determined by accumulating the

Digital Object Identifier 10.1109/MIS.2018.111145153 1541-1672/$26.00 2018 IEEE

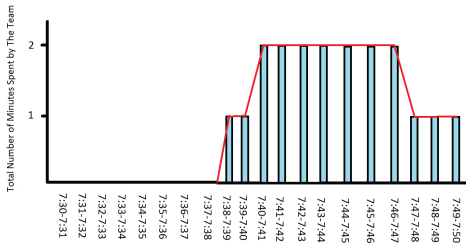

Figure 2. An example time series generated using the accumulated number of minutes spent by the entire team for the first 20 minutes of intrusion stage 6 (Deployment) of the INL dataset within each one minute time interval.

number of minutes spent by the Red Team on its corresponding intrusion stage. An example time series generated using this approach is depicted in Figure 2. Note that the value of the time series can be larger than 1 due to the fact that the time series values represent the sum of the number of minutes spent by the whole team on an intrusion stage within a one minute time span.

After creating the time series data, we were able to perform a prediction of the time series values using recurrent neural networks (RNN). RNNs are popular predictive models in identification and control of dynamic processes. Generally, this family of predictive algorithms consists of a multi-layer perception (MLP) that takes as input a window of past independent inputs as well as past outputs and calculates the current output. In this study, the values of the time series representation of the intrusion chain stages and their prior prediction outputs are used as input to the MLP. Therefore, if the prior activities of the attacking team members are recorded through a defense log mechanism, a prediction of their future activities is possible using this approach. This includes predicting the amount of time the adversarial team will invest on each intrusion chain stage n steps later in their adversarial process. This prediction can help the efficiency and precision of dynamic defense measures by providing the possibility of targeting the intrusion stages that are more likely to be focused on by the adversarial team. In this study, we utilized and compared four different predictive models: the Non-linear autoregressive model (NAR), the Non-linear autoregressive exogenous model (NARX), the autoregressive neural networks for multi-steps-ahead prediction, and autoregressive integrated moving average (ARIMA). Given that we focused on application and comparison of the autoregressive models family, other neural network models such as the long short-term memory (LSTM) were not included. In the next sections we explain each of these algorithms briefly.

4.3. Non-linear Autoregressive Neural Network Model

In various applications, time series are characterized by high variations and sporadic behavior. This makes it difficult

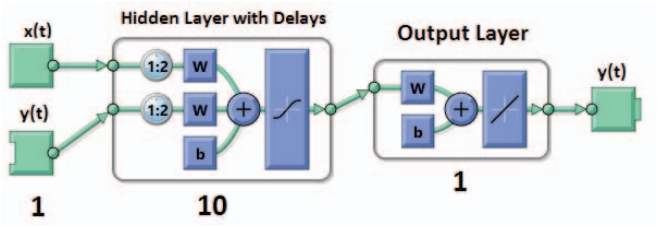

Figure 3. NARX model architecture

to model time series using liner models, therefore, a non-linear approach is more appropriate. The amount of time that the adversarial team spends on each intrusion stage can include such fleeting transient periods due to various reasons such as disruptions from the defense team, adversarial team's own failures, changes in team dynamics, etc. Therefore, in this study we used non-linear models to predict the values of the time series data generated for the intrusion stages.

A non-linear autoregressive neural network is a discrete, non-linear model that can be written as follows [6]:

$$y(t) = h(y(t1), y(t2), ..., y(t-p)) + e(t) \quad (1)$$

where the NAR network is used to predict the value of a data series y at time t, y(t), using the p past values of the series. The function h is unknown in advance, and the training of the neural network aims to approximate the function through optimizing the network weights and neuron bias. The network training function in this study updates the weight and bias values according to LevenbergMarquardt optimization [6]. Also, e(t) is the error of approximation of the series y at time t.

4.4. Non-linear autoregressive neural network with exogenous Input

In many real applications, there is an important correlation between the modelled time series and additional external data. Thus, the integration of knowledge or data about weather could benefit the time series modelling process process to provide an accurate forecast. The model Non-linear autoregressive with exogenous (External) inputs (NARX) is proposed in [7]. NARX predict series given past values of series and another external series. The equation that models the NARX network behavior for time series prediction is shown below [7], [8]:

$$y(t) = h(x(t-1), ..., x(t-k), y(t-1), ..., y(t-p)) + e(t) \quad (2)$$

As presented in Figure 3, the difference between this model and the NAR model is in the input where in the NARX model, in addition to the past values of the target time series, the past values of another time series is fed into the network. In this study, we suggest two approaches for selecting the exogenous time series. In the first approach, for prediction of the value of each intrusion stage, select its previous stage according to the intrusion chain model as the

Digital Object Identifier 10.1109/MIS.2018.111145153 1541-1672/$26.00 2018 IEEE

exogenous stage. For instance, for prediction of the intrusion stage 6 "deployment", we use the previous value of stage 5 "test for detection". For the second approach, we select the most correlated time series to the target time series as the exogenous input. The reason behind this approach is that the most correlated time series provides the most information about the target time series, which can help increase the accuracy of the model. Pearson correlation was used to find the correlations between the time series.

4.5. Non-linear autoregressive neural network for Multi-Step Prediction

In many applications, predicting the more value of time series for more than one step ahead has a significant value. Achieving such a prediction for intrusion stages has the benefit of creating more robust proactive measures. In order to make a multi-step-ahead prediction, we can train a neural network up the present with all the known values of a time-series in open-loop mode, then convert the architecture to closed-loop mode for multiple steps ahead predictions of the future values [9]. Therefore, In order to predict the next p time steps, we can use the network to predict the y outputs using each of its predictions feedback to help the network perform the next prediction.

4.6. Autoregressive Integrated Moving Average

An autoregressive integrated moving average (ARIMA) model makes prediction of time series values based upon prior values (AR terms) as well as the errors made by previous predictions (MA terms). This allows the model to adjust itself to sudden changes in the time series.

Therefore, the ARIMA forecasting equation for a stationary time series is a linear regression equation in which the predictors are the lags of the dependent variable and/or lags of the prediction errors This method can be modeled as follows:

$$x_t = \delta + \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t \tag{3}$$

Where x_t is a linear function of the values of x at the previous two times. Assume we have observed p data values of the time series and wish to use the observed data and estimated model to forecast the value of x_{p+1} , the values of the series at the next time point. The equations for making this prediction is:

$$x_{p+1} = \delta + \phi_1 x_p + \phi_2 x_{p-1} + w_{p+1} \tag{4}$$

As stated in this equation, the observed values of x_n and x_{n-1} are used and w_{p+1} is replaced by the assumed mean of the errors. This model is explained in more detail in [10].

5. Results

Time series representations were created for each intrusion stage through the process explained in the methodology section. Each time point in the generated time series represents a one minute time span. For each time point, the value of each time series is the accumulated number of minutes spent by the entire team on its corresponding intrusion stage. Ten Time series were created for the first dataset, and eight time series were created for the second dataset. The length of the generated time series for the first and second datasets were 480 and 300 minutes respectively.

5.1. Experimental Setup and Hyper Parameters

The datasets were randomly divided into three segments for training, and testing purpose: 85% of each time series (336 minutes for dataset 1, and 210 minutes for dataset 2) was used for training, and 15% (72 minutes for dataset 1 and 45 minutes for dataset 2) for testing. Also, 5-fold cross-validation was performed during training. This segmentation was applied to two datasets for separate experiments. The neural network structures consisted of 10 hidden neurons, and the number of delays d was set to 2. To ensure that the neural networks have reliable accuracies, each model was trained 20 times, and the MSE values were averaged to obtain a final MSE value.

5.2. Prediction Results

Through the aforementioned experiments, we try to predict the time that the Red Team will spend on each intrusion stage in the next step(s). This means that a prediction takes place for each generated intrusion chain time series separately, and the predicted value represents the amount of time, in minutes, that the whole team will spend on the corresponding intrusion stage during the next step. Comparing the predictions of the time series can reveal what intrusion stage will be more focused on by the Red Team during the next step. The performance of the models was evaluated using the statistical error measurements; mean square error (MSE) which is defined in the following equations:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(\hat{Y}_i - Y_i)^2 \tag{5}$$

The MSE measures the average of the errors absolute value and then the errors are turned positive by raising it to the square, with the advantage of being easy to handle and with a wider usage in the optimization technique.

The prediction results of the NARX approach for time series corresponding to two intrusion chain stage time series (deployment, and initial intrusion) are presented in Figure 4. The bottom figures show the error rate for each time point. The training (blue), validation (green), and testing (red) time points are randomly sampled. Also, the error rate during each of the three mentioned phases is presented on the right side of the same figure. We can observe the convergence of the error rate as the neural network epoch progresses. The circles in that figure show the first epoch where the best (converged) performance is achieved. For instance, We can observe that the best testing performance

Digital Object Identifier 10.1109/MIS.2018.111145153 1541-1672/$26.00 2018 IEEE

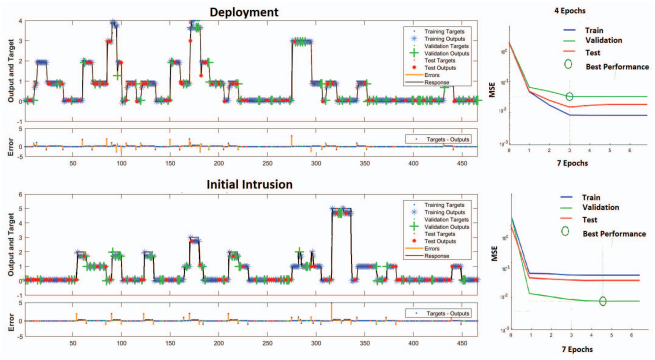

Figure 4. The NARX prediction results for intrusion stages 6 (Deployment) and 7 (Initial Intrusion) for dataset 1 (INL)

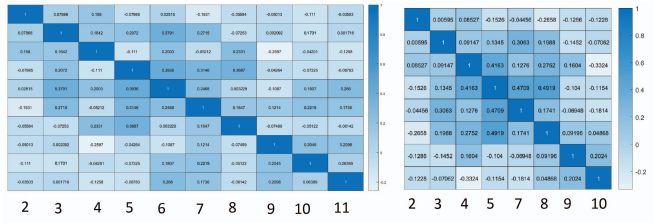

Figure 5. The correlation map for the two datasets: On the right side: Alphaville dataset; On the left side: INL dataset. The numbers on the columns and rows correspond to the intrusion chain according to Figure.1. For example, number 2 corresponds to Find and Organize Accomplices intrusion stage.

(red line) for 'Deployment' intrusion stage time series is at epoch 4, where the MSE is 0.046. The complete testing MSE results are presented in Table1.

The NARX method in this study included two variations of the exogenous input. The first approach is to use the 12-stage intrusion chain model in which the intrusion steps occur in sequential order. Using that model, in order to make a prediction for each intrusion stage, we used its previous stage as the exogenous input. For instance, in order to predict the value of a time point in the Initial Intrusion stage, in addition to its own time series, the Deployment stage was also used for neural network learning. In the second approach, we used the most correlated time series to each of the time series as the exogenous input. This approach tries to find which time series can reinforce the neural network's learning power without considering the order of the time series based on the intrusion chain model. The correlation table for all intrusion stages for both datasets is presented in Figure 5. For instance, in the first data set (Figure 5., the left table) in order to make a prediction for intrusion stage 2 (Find and organize accomplices), aside from using the data from previous steps of its corresponding time series, we also incorporate the data from the previous steps of the most correlated time series to it, which belongs to intrusion stage 4 (Research target infrastructure/employees).

The complete prediction results are provided in Table 1 and Table 2 for four neural network architectures: NAR, sequential NARX (two variations: sequential Narx(S) , and correlation-based NARX(C)), ARIMA, and multi-Step-ahead NAR(M). In that table, the results with lower MSE are displayed in bold. Note that sine we did not observe the Red Team performing intrusion stage 11 during the Alphaville experiment, it is not included in table 2. We can observe that the correlation-based NARX model provides the highest accuracy in 6 cases in the first experiment which is presented in table 1. These results give an empirical proof that that utilizing the most correlated time series as the exogenous input improves the accuracy of the model more than picking the time series according to the sequential 12-stage intrusion model. One can argue that this conclusion is understandable due to the fact that in the correlation-based model, we heuristically look for the most relevant intrusion chain time series to the target time series and employ it to enhance the learning process of our model. Similar conclusion can be drawn from the second experiment which is presented in table 2. In that experiment, the predictions made by incorporating the most correlated intrusion stage time series show a superior accuracy for more than half of the intrusion stages compared to other models.

TABLE 1. A COMPARISON OF THE MSE VALUES OF THE FOUR METHODS FOR EACH TIME SERIES ON DATASET 1 (INL). NARX(S) IS THE SEQUENTIAL INPUT NARX MODEL, NARX(C) IS THE CORRELATION-BASED INPUT NARX, AND NAR(M) IS THE TEN MINUTES STEP AHEAD NAR MODEL

Intrusion Stage	NAR	NARX(S)	NARX(C)	ARIMA	NAR(M)
2	0.0908	0.0720	**0.0531**	0.1048	0.0586
3	0.1140	0.0466	0.0203	**0.0070**	0.1220
4	**0.0154**	0.0833	0.0601	0.0216	0.0442
5	0.0424	**0.0085**	0.0622	0.0432	0.0662
6	0.2100	0.0467	**0.0188**	0.0344	0.0312
7	0.0463	0.0804	**0.0422**	0.0743	0.0845
8	0.0786	0.0601	0.0611	**0.0306**	0.0588
9	0.0497	0.0957	**0.0232**	0.0861	0.0918
10	0.0132	0.0223	**0.0080**	0.0324	0.0324
11	0.0416	0.0632	**0.0481**	0.0557	0.0870
Mean	0.0702	0.0578	**0.0397**	0.0800	0.0676

TABLE 2. A COMPARISON OF THE MSE VALUES OF THE FOUR METHODS FOR EACH TIME SERIES ON DATASET 2 (ALPHAVILLE)

Intrusion Stage	NAR	NARX(S)	NARX(C)	ARIMA	NAR(M)
2	0.0373	0.0713	**0.0266**	0.0637	0.0412
3	0.0311	0.0412	**0.0233**	0.2452	0.3122
4	**0.2004**	0.2820	0.2218	0.0216	0.0266
5	**0.0022**	0.0016	0.0056	0.0106	0.0092
7	0.0542	0.0822	**0.0450**	0.0743	0.9044
8	0.1206	0.1488	**0.1152**	0.1821	0.1571
9	0.0045	0.0072	**0.0021**	0.0133	0.0155
10	0.1602	0.2086	**0.1288**	0.2084	0.2213
Mean	0.0763	0.0906	**0.0729**	0.0964	0.2109

6. Discussion

This paper offers a framework for dynamic prediction of adversarial movement across the cyber intrusion chain. However, there are some limitations to this analysis. First,

Digital Object Identifier 10.1109/MIS.2018.111145153 1541-1672/$26.00 2018 IEEE

the analysis put forth in this paper is based on two case studies, which has implications for generalizability of the results and accuracies of the compared models. Also, it can be useful to test and compare the accuracies of the various RNN architectures in the presence of more chaotic intrusion chain time series. However, due to the nature of these datasets, we can argue that it is not remarkably expected that the time series generated for each one minute time span can show highly chaotic behavior. This is due to the fact that a highly chaotic behavior of the intrusion chain time series would require frequent changes in the decision making process of the adversarial team within each minute time span. Second, it is worth considering that there are many permutation and combinations of attack scenarios, adversarial types and motivations, objectives,and organizational dynamics, which cannot be accounted for by two case studies. Therefore, more observations would be needed to better be able to make a predictive and dynamic mechanism for the intrusion chain analysis. While these are indeed valid limitations, the analysis method offered in this paper provides a verifiable framework which can be utilized and performed in various experimental setups and dynamics as RNNs have shown robust performance in a wide range of applications involving time series prediction and analysis. Here we present our conclusions from our experimental analysis:

1. NARX(C) model best predicts adversarial movement (for both datasets) for 60% of stages in dataset 1 and 75% of stages in dataset 2. Despite the multitude of differences in the datasets with regards to exercise duration (8 hours vs. 5 hours), structure (red/blue vs. paintball), setting (cyber-physical facility vs. virtual city), team size (10 members vs. 4 members), and the team members's familiarity (randomly assembled vs. some prior relationship), the model was still able to predict each teammovement through the 12 stages and the time spent on these stages with reliable accuracies in the next time step/unit.

2. The NARX(C) model was the best at predicting behavior, indicating that using the most correlated stages for prediction had better accuracy than predictions based on sequential stages. This might indicate that intrusion chains are complex and do not effectively capture adversarial back and forth movements, that adversaries might be working on multiple intrusion chains (and stages) concurrently.

3. NARX(C) model best predicted three stages of the intrusion chain: finding and organizing accomplices (2), expanding access (9), strengthening foothold (10). The stages used for predicting these stages in Dataset 1 were researching target infrastructure (4), exfiltrating data (11), and initial intrusion (7), while for Dataset 2, the researching target infrastructure (4), strengthening foothold (10), and expanding access (9) stages were used for prediction. This suggests that stages (i) more time is spent on these stages (ii) at similar times, and (iii) for similar durations. Thus, for Dataset 2, the expanding access (9) stage was used to predict the strengthening foothold (10) stage. This meant that the team spent more time on these stages (relative to the other stages), for similar durations, at similar temporal points during the exercise.

4. Prior research [4] suggests that after the Red Team experienced disruptions (caused by opponent players), the team spent more time on certain stages for at least 5-10 minutes immediately after the disruptions. Therefore, we used a closed loop architecture of the autoregressive neural network model to create a prediction of 10 next time steps of the intrusion stages using feedback from previous predictions. This offers insight into which stages the Red Team focuses more on, how much time they spend on these stages, and whether they also focus on other stages simultaneously. As it can be observed from Tables 1 and 2, the analysis results suggests that a more accurate prediction for the 1-minute prediction was achieved compared to the 10-minute-ahead prediction using the closed loop network. However, both 1-minute and 10-minute-ahead models offer a means to predict adversarial movements within 1 and 10 minute time stamps and show strong performance regarding adapting to the sudden variations in the process which can help understand variations in attack progression.

7. Conclusion

Cybersecurity experts have identified five APT trends, namely that there will be more attacks, more obfuscation, continued false attribution, greater shifts from opportunity-based attacks to more targeted attacks, and more damage that ranges from data manipulation to data encryption or deletion. Governments and critical infrastructure must continuously adapt to an ever-changing and evolving threat landscape by embracing proactive cybersecurity approaches that provides insight into the most likely who, what, where, when, and how of attacks and the best way to begin looking for them. This paper proposed and used an innovative mixed methods-based approach to predict adversarial movement thereby contributing to the dialog on proactive security. The experimental results show that the machine learning models of this study can provide a reliable prediction of the movements of the adversaries during cyberattacks.

Acknowledgments

This material is based upon work supported by the National Science Foundation CAREER Award, Grant No. 1453040 and partially by National Science Foundation CPS Award, Grant No. 1446574.

The authors thank the United States Industrial Control Systems Computer Emergency Response Team (ICS-CERT) and Idaho National Laboratory (INL) for allowing data collection at their September/October 2014 Red Team/Blue Team Cybersecurity Training Exercise. The authors also thank the Merit Network and the Michigan Cyber Range for allowing data collection at their 2015 NAICS event.

References

[1] M. Cloppert "Security Intelligence: Attacking the Cyber Kill Chain" Online at http://digital-forensics.sans.org/blog/2009/10/14/security-intelligence-attacking-the-kill-chain, Retrieved Feb 2, 2014.

[2] R. Colbaugh and K. Glass, "Proactive Defense for Evolving Cyber Threats". Sandia National Laboratories [SAND2012-10177]. Online at https://fas.org/irp/eprint/proactive.pdf, Retrieved Feb 15, 2017.

[3] J. 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

[4] A. Rege, Z. Obradovic, N. Asadi, E. Parker, B. Singer and N. Masceri, (forthcoming). "A Temporal Assessment of Cyber Intrusion Chains Using Multidisciplinary Frameworks and Methodologies". *Proceedings of the International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA 2017)*. Institute of Electrical and Electronics Engineers (IEEE) Xplore Digital Library.

[5] R. Meija, "Red team versus blue team: How to run an effective simulation". Online at http://www.csoonline.com/article/2122440/disaster-recovery/emergency-preparedness-red-team-versus-blue-team-how-to-run-an-effective-simulation.html, Retrieved Feb 15, 2016

[6] T. W. S. Chow and C. T. Leung, "Non-linear autoregressive integrated neural network model for short-term load forecasting." *IEEE Proceedings-Generation, Transmission and Distribution*, 143(5), 500-506.

[7] T. Lin, B. G. Horne, P. Tino and C. L. Giles. "Learning long-term dependencies in NARX recurrent neural networks." *IEEE Transactions on Neural Networks*, 1996, 7(6), 1329-1338.

[8] H. T. Siegelmann, B. G. Horne and C. L. Giles, "Computational capabilities of recurrent NARX neural networks." *IEEE Transactions on Systems, Man, and Cybernetics*, Part B (Cybernetics), 27(2), 208-215.

[9] A. Andalib andF. Atry, " Multi-step ahead forecasts for electricity prices using NARX: a new approach, a critical analysis of one-step ahead forecasts.", *Energy Conversion and Management,* 50(3), 739-747.

[10] J. Rob and G. Athanasopoulos, "Notation for ARIMA Models Time Series Forecasting System". SAS Institute. Retrieved 19 May 2015.

Digital Object Identifier 10.1109/MIS.2018.111145153 1541-1672/$26.00 2018 IEEE